Tell us about your PDF experience.

Virtual machines in Azure

Documentation for creating and managing virtual machines in Azure.

Latest features

WHAT'S NEW

Dasv7 size series - general purpose

Dsv6 size series - general purpose

Fasv7 size series - compute optimized

Azure Boost

Hibernation

Linux quickstarts

QUICKSTART

Azure CLI

Azure portal

Azure PowerShell

Terraform

Windows quickstarts

QUICKSTART

Azure portal

Azure PowerShell

Terraform

Azure CLI

Self-paced training

TRAINING

Linux in Azure learning path

Create a Windows virtual machine in Azure

Troubleshooting

HOW-TO GUIDE

Troubleshooting Guides

Troubleshoot deployment issues

Troubleshoot allocation failures

Troubleshoot unexpected reboots

Microsoft Q&A

Architecture

ARCHITECTURE

Azure Well-Architected Framework Virtual Machine considerations

Azure Well-Architected Framework Disk Storage considerations

Cloud adoption framework

Azure architecture center

Resources

OVERVIEW

Azure Quickstart templates

Linux VM pricing

Regional availability

Virtual machines selector tool

Virtual machines in Azure

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

Azure virtual machines (VMs) are one of several types of on-demand, scalable computing resources that Azure offers. Typically, you choose a virtual machine when you need more control over the computing environment than the other choices offer. This article gives you information about what you should consider before you create a virtual machine, how you create it, and how you manage it.

An Azure virtual machine gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it. However, you still need to maintain the virtual machine by performing tasks, such as configuring, patching, and installing the software that runs on it.

Azure virtual machines can be used in various ways. Some examples are:

- **Development and test** – Azure virtual machines offer a quick and easy way to create a computer with specific configurations required to code and test an application.
- **Applications in the cloud** – Because demand for your application can fluctuate, it might make economic sense to run it on a virtual machine in Azure. You pay for extra virtual machines when you need them and shut them down when you don't.
- **Extended datacenter** – virtual machines in an Azure virtual network can easily be connected to your organization's network.

What do I need to think about before creating a virtual machine?

There's always a multitude of design considerations when you build out an application infrastructure in Azure. These aspects of a virtual machine are important to think about before you start:

- The names of your resources
- The location where the resources are stored
- The size of the virtual machine
- The maximum number of virtual machines that can be created
- The operating system that the virtual machine runs

- The configuration of the virtual machine after it starts
- The related resources that the virtual machine needs

> **Note**
>
> Trusted Launch as default (TLaD) is available in preview for new Generation 2 Virtual machines (VMs). With TLaD, any new Generation 2 VMs created through any client tools defaults to Trusted Launch with secure boot and vTPM enabled. Register for the TLaD preview to validate the default changes in your respective environment and prepare for the upcoming change.

Parts of a VM and how they're billed

When you create a virtual machine, you're also creating resources that support the virtual machine. These resources come with their own costs that should be considered.

The default resources supporting a virtual machine and how they're billed are detailed in the following table:

Expand table

Resource	Description	Cost
Virtual network	For giving your virtual machine the ability to communicate with other resources	Virtual Network pricing
A virtual Network Interface Card (NIC)	For connecting to the virtual network	There's no separate cost for NICs. However, there's a limit to how many NICs you can use based on your VM's size. Size your VM accordingly and reference Virtual Machine pricing.
A private IP address and sometimes a public IP address.	For communication and data exchange on your network and with external networks	IP Addresses pricing
Network security group (NSG)	For managing the network traffic to and from your VM. For example, you might need to open port 22 for SSH access, but you might want to block traffic to port 80. Blocking and allowing port access is done through the NSG.	There are no additional charges for network security groups in Azure.

Resource	Description	Cost
OS Disk and possibly separate disks for data.	It's a best practice to keep your data on a separate disk from your operating system, in case you ever have a VM fail, you can detach the data disk, and attach it to a new VM.	All new virtual machines have an operating system disk and a local disk. Azure doesn't charge for local disk storage. The operating system disk, which is usually 127GiB but is smaller for some images, is charged at the regular rate for disks. You can see the cost for attach Premium (SSD based) and Standard (HDD) based disks to your virtual machines on the managed disks pricing page.
In some cases, a license for the OS	For providing your virtual machine runs to run the OS	The cost varies based on the number of cores on your VM, so size your VM accordingly. The cost can be reduced through the Azure Hybrid Benefit.

You can also choose to have Azure can create and store public and private SSH keys - Azure uses the public key in your VM and you use the private key when you access the VM over SSH. Otherwise, you need a username and password.

By default, these resources are created in the same resource group as the VM.

Locations

There are multiple geographical regions around the world where you can create Azure resources. Usually, the region is called **location** when you create a virtual machine. For a virtual machine, the location specifies where the virtual hard disks are stored.

This table shows some of the ways you can get a list of available locations.

Expand table

Method	Description
Azure portal	Select a location from the list when you create a virtual machine.
Azure PowerShell	Use the Get-AzLocation command.
REST API	Use the List locations operation.

Method	Description
Azure CLI	Use the az account list-locations operation.

Availability

There are multiple options to manage the availability of your virtual machines in Azure.

- **Availability Zones** are physically separated zones within an Azure region. Availability zones guarantee virtual machine connectivity to at least one instance at least 99.99% of the time when you have two or more instances deployed across two or more Availability Zones in the same Azure region.
- **Virtual Machine Scale Sets** let you create and manage a group of load balanced virtual machines. The number of virtual machine instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update many virtual machines. Virtual machines in a scale set can also be deployed into multiple availability zones, a single availability zone, or regionally.

Fore more information see Availability options for Azure virtual machines and SLA for Azure virtual machines.

Sizes and pricing

The size of the virtual machine that you use is determined by the workload that you want to run. The size that you choose then determines factors such as processing power, memory, storage capacity, and network bandwidth. Azure offers a wide variety of sizes to support many types of uses.

Azure charges an hourly price based on the virtual machine's size and operating system. For partial hours, Azure charges only for the minutes used. Storage is priced and charged separately.

Scaling to multiple VMs

The number of virtual machines that your application uses can scale up and out to whatever is required to meet your needs. Azure offers several automatic scaling systems and instance types based on the size of your workload. For more information on scaling, compare VM-based compute products.

Virtual machine total core limits

Your subscription has default quota limits in place that could impact the deployment of many virtual machines for your project. The current limit on a per subscription basis is 20 virtual machine total cores per region. Limits can be raised by filing a support ticket requesting an increase Quota and capacity are checked separately during deployment. Even with approved quota, deployment can still fail if there isn't enough capacity in the target region or availability zone.

Managed disks

Managed disks handles Azure Storage account creation and management in the background for you, and ensures that you don't have to worry about the scalability limits of the storage account. You specify the disk size and the performance tier (Standard or Premium), and Azure creates and manages the disk. As you add disks or scale the virtual machine up and down, you don't have to worry about the storage being used. If you're creating new virtual machines, use the Azure CLI or the Azure portal to create virtual machines with Managed OS and data disks. If you have virtual machines with unmanaged disks, you can convert your virtual machines to be backed with managed disks.

You can also manage your custom images in one storage account per Azure region, and use them to create hundreds of virtual machines in the same subscription. For more information about managed disks, see the managed disks Overview.

Distributions

Microsoft Azure supports various Linux and Windows distributions. You can find available distributions in the marketplace, Azure portal or by querying results using CLI, PowerShell, and REST APIs.

This table shows some ways that you can find the information for an image.

Expand table

Method	Description
Azure portal	The values are automatically specified for you when you select an image to use.
Azure PowerShell	Get-AzVMImagePublisher -Location *location* Get-AzVMImageOffer -Location *location* -Publisher *publisherName*

Method	Description
	Get-AzVMImageSku -Location *location* -Publisher *publisherName* -Offer *offerName*
REST APIs	List image publishers List image offers List image skus
Azure CLI	az vm image list-publishers --location *location* az vm image list-offers --location *location* --publisher *publisherName* az vm image list-skus --location *location* --publisher *publisherName* --offer *offerName*

Microsoft works closely with partners to ensure the images available are updated and optimized for an Azure runtime. For more information on Azure partner offers, see the Azure Marketplace

Cloud-init

Azure supports for cloud-init across most Linux distributions that support it. We're actively working with our Linux partners to make cloud-init enabled images available in the Azure Marketplace. These images make your cloud-init deployments and configurations work seamlessly with virtual machines and virtual machine scale sets.

For more information, see Using cloud-init on Azure Linux virtual machines.

Storage

- Introduction to Microsoft Azure Storage
- Add a disk to a Linux virtual machine using the azure-cli
- How to attach a data disk to a Linux virtual machine in the Azure portal

Local temporary storage

Some Azure VM sizes include local temporary ephemeral storage, with some of the newer sizes using Temporary local NVMe disks. Local temporary storage uses additional disks provisioned directly as local storage to an Azure virtual machine host, rather than on remote Azure Storage. This type of storage is best suited for data that does not need to be retained permanently, such as caches, buffers, and temporary files. Since data stored on these disks isn't backed up, it's lost when the VM is deallocated or deleted. The ephemeral storage is recreated on startup. Local ephemeral disks are different from Ephemeral OS disks.

Local temporary disks present just like other remote storage such as Premium SSD v2 disks, but have added performance benefits and don't count against the IOPS and throughput limits of the VM.

The following lists key characteristics of local ephemeral storage:

- **Ultra-low latency**: Read/write operations occur on hardware physically co-located with the VM's compute resources, minimizing network hops and enabling sub-millisecond response times.
- **Superior IOPS and throughput**: Performance scales with the underlying local storage, often delivering 8-10 times higher IOPS than Standard HDDs when using Premium SSD as the base, with limits tied to the VM size (up to the NVMe disk capacity, capped at 2,040 GiB).

Azure VM sizes featuring a 'd' in their naming convention - such as the Da***d***sv6, Ea***d***sv6, and FXm***d***sv2 series - include dedicated local NVMe storage. These sizes enable placing temporary files, swap space, or high-I/O components like SQL Server's `tempdb` database files on the ephemeral disk for enhanced performance.

Networking

- Virtual Network Overview
- IP addresses in Azure
- Opening ports to a Linux virtual machine in Azure
- Create a Fully Qualified Domain Name in the Azure portal

Service disruptions

At Microsoft, we work hard to make sure that our services are always available to you when you need them. Forces beyond our control sometimes impact us in ways that cause unplanned service disruptions.

Microsoft provides a Service Level Agreement (SLA) for its services as a commitment for uptime and connectivity. The SLA for individual Azure services can be found at Azure Service Level Agreements.

Azure already has many built-in platform features that support highly available applications. For more about these services, read Disaster recovery and high availability for Azure applications.

This article covers a true disaster recovery scenario, when a whole region experiences an outage due to major natural disaster or widespread service interruption. These are rare occurrences, but you must prepare for the possibility that there's an outage of an entire region. If an entire region experiences a service disruption, the locally redundant copies of your data would temporarily be unavailable. If you enabled geo-replication, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region isn't recoverable, Azure remaps all of the DNS entries to the geo-replicated region.

In the case of a service disruption of the entire region where your Azure virtual machine application is deployed, we provide the following guidance for Azure virtual machines.

Option 1: Initiate a failover by using Azure Site Recovery

You can configure Azure Site Recovery for your VMs so that you can recover your application with a single click in matter of minutes. You can replicate to Azure region of your choice and not restricted to paired regions. You can get started by replicating your virtual machines. You can create a recovery plan so that you can automate the entire failover process for your application. You can test your failovers beforehand without impacting production application or the ongoing replication. In the event of a primary region disruption, you just initiate a failover and bring your application in target region.

Option 2: Wait for recovery

In this case, no action on your part is required. Know that we're working diligently to restore service availability. You can see the current service status on our Azure Service Health Dashboard.

This option is the best if you don't set up Azure Site Recovery, read-access geo-redundant storage, or geo-redundant storage prior to the disruption. If you set up geo-redundant storage or read-access geo-redundant storage for the storage account where your VM virtual hard drives (VHDs) are stored, you can look to recover the base image VHD and try to provision a new VM from it. This option isn't preferred because there are no guarantees of synchronization of data, which means this option isn't guaranteed to work.

> **Note**
>
> Be aware that you don't have any control over this process, and it will only occur for region-wide service disruptions. Because of this, you must also rely on other application-

specific backup strategies to achieve the highest level of availability. For more information, see the section on Data strategies for disaster recovery.

Resources for service disruptions

- Start protecting your applications running on Azure virtual machines using Azure Site Recovery
- To learn more about how to implement a disaster recovery and high availability strategy, see Disaster recovery and high availability for Azure applications.
- To develop a detailed technical understanding of a cloud platform's capabilities, see Azure resiliency technical guidance.
- If the instructions aren't clear, or if you would like Microsoft to do the operations on your behalf, contact Customer Support.

Data residency

In Azure, the feature to enable storing customer data in a single region is currently only available in the Southeast Asia Region (Singapore) of the Asia Pacific Geo and Brazil South (Sao Paulo State) Region of Brazil Geo. For all other regions, customer data is stored in Geo. For more information, see Trust Center.

Next steps

Create your first virtual machine!

- Portal
- Azure CLI
- PowerShell

Last updated on 12/01/2025

Compare virtual machine-based compute products

Article • 05/07/2025

Microsoft Azure offers various virtual machine based compute products designed to meet different workload requirements. Standard Virtual Machines (VMs), Virtual Machine Scale Sets (VMSS), and Compute Fleet are some of the compute products that provide flexibility and scalability for various applications.

Virtual Machines (VMs) are the fundamental building blocks of Azure compute. They offer full control over the operating system and applications, making them suitable for a wide range of workloads. VMs provide the flexibility to choose the operating system, size, and configuration that best fits your needs.

Virtual Machine Scale Sets (VMSS) extend the capabilities of VMs by enabling the deployment and management of a group of VMs. This service is ideal for applications that require high availability or scalability. VMSS can ensure that your applications can handle varying loads efficiently by automatically increasing or decreasing the number of VMs in response to changes in demand.

Compute Fleet is designed for large-scale, distributed computing environments. It allows you to manage and orchestrate a fleet of VMs across multiple regions and availability zones. This service is particularly beneficial for high-performance computing (HPC) applications, big data processing, and other workloads that require significant computational power and distributed resources.

The product you select depends on the workload you intend to run. Check out the product comparison table in this article or in the Azure portal to choose between these solutions.

Product comparison table

Expand table

	 **Virtual Machines**	 **Virtual Machine Scale Sets**	 **Compute Fleet**
Product Description	Create and manually configure individual VMs.	Load-balanced, automatically scaling VM groups with workload optimization across multiple availability zones.	Provision thousands of mixed-size VMs for performance and high availability.

	Virtual Machines	Virtual Machine Scale Sets	Compute Fleet
Instances	Single instances deployed individually	2 to 1,000 instances	Up to 10,000 instances
Product Differences See top product capabilities and limitations side by side	- Deploy and manage each VM individually - Fine-tune custom applications - Assigned to one availability zone at a time	- Up to 1,000 mixed-size VMs in a group - Scale automatically - Fault domains for high availability - Integrate Azure Spot VMs to cut costs - Optimized for stateless or stateful workloads	- Up to 10,000 mixed-size VMs in a group - Fault domains for high availability - Hyper-scale with demand - Maintain capacity with Spot VMs to cut costs - Optimize allocation by price, capacity, or both
Use Case Find the right product for your project	- Small, simple jobs - Specialized single-instance apps - Test configurations and experiment with Azure	- Scaling beyond a single VM - Platform engineering or infrastructure as code - Moving on-premises apps to the cloud - Parallelized high-performance computing workloads - Databases - Batch processing - Maintain performance while controlling costs - Mix VM sizes for more flexibility	- Large scale, highly parallelized workloads or batch jobs - Flexibility with VM sizes - Large scale cost optimization with Azure Spot

Product specifications

Cost

The cost of any Azure VM environment depends upon the number of VM instances, their configuration, and selected add-on services. There's no added charge for using Virtual Machine Scale Sets or Compute Fleet.

Expand table

Virtual Machines	Virtual Machine Scale Sets	Compute Fleet
- Billed as an individual VM instance - Cost-effective for consistent load and traffic	 **Optimize costs with VMSS** - VMSS can automatically use less capacity when load/traffic is low, and optimizes cost while accommodating the dips and spikes of cyclical, intermittent, and growing workloads - Use 100% Spot VMs, or mix with standard ones, to cut cost	 **Optimize costs with Compute Fleet** - Combine discounts (like reserved instances and savings plans) with Spot VM savings, plus rank the top VM sizes you want Fleet to prioritize - Your fleet will maintain capacity with discounted Spot VMs to optimize cost

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Quickstart: Create a Linux virtual machine with the Azure CLI on Azure

Applies to: ✔ Linux VMs

Deploy to Azure

This quickstart shows you how to use the Azure CLI to deploy a Linux virtual machine (VM) in Azure. Use the Azure CLI to create and manage Azure resources from the command line or in scripts.

If you don't have an Azure subscription, create a free account before you begin.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, select **Try it** from the upper right corner of a code block. You can also open Cloud Shell in a separate browser tab by going to https://shell.azure.com/bash. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and select **Enter** to run it.

If you prefer to install and use the CLI locally, this quickstart requires Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Log in to Azure using the CLI

To run commands in Azure by using the CLI, you need to sign in first. Sign in by using the `az login` command.

Create a resource group

A resource group is a container for related resources. You must place all resources in a resource group. Use the az group create command to create a resource group with the previously defined `$MY_RESOURCE_GROUP_NAME` and `$REGION` parameters.

Bash

```
export RANDOM_ID="$(openssl rand -hex 3)"
export MY_RESOURCE_GROUP_NAME="myVMResourceGroup$RANDOM_ID"
```

```
export REGION=EastUS
az group create --name $MY_RESOURCE_GROUP_NAME --location $REGION
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMResourceGroup",
  "location": "eastus",
  "managedBy": null,
  "name": "myVMResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```


Create the virtual machine

To create a VM in this resource group, use the `vm create` command.

The following example creates a VM and adds a user account. The `--generate-ssh-keys` parameter causes the CLI to look for an available SSH key in `~/.ssh`. If it finds one, it uses that key. If not, it generates and stores a key in `~/.ssh`. The `--public-ip-sku Standard` parameter ensures that the machine is accessible through a public IP address. Finally, it deploys the latest `Ubuntu 22.04` image.

Configure all other values by using environment variables.

Bash

```
export MY_VM_NAME="myVM$RANDOM_ID"
export MY_USERNAME=azureuser
export MY_VM_IMAGE="Canonical:0001-com-ubuntu-minimal-jammy:minimal-22_04-lts-
gen2:latest"
az vm create \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --name $MY_VM_NAME \
    --image $MY_VM_IMAGE \
    --admin-username $MY_USERNAME \
    --assign-identity \
    --generate-ssh-keys \
    --public-ip-sku Standard
```

It takes a few minutes to create the VM and supporting resources. The following example output shows the VM create operation was successful.

Results:

JSON


```
{
  "fqdns": "",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMResourceGroup/providers/Microsoft.Compute/virtualMa
chines/myVM",
  "location": "eastus",
  "macAddress": "00-0D-3A-10-4F-70",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.147.208.85",
  "resourceGroup": "myVMResourceGroup",
  "zones": ""
}
```


Enable Azure AD Authentication for a Linux virtual machine in Azure

The following code example deploys a Linux VM and then installs the extension to enable Azure AD Authentication for a Linux VM. VM extensions are small applications that provide post-deployment configuration and automation tasks on Azure virtual machines.

Bash

```
az vm extension set \
    --publisher Microsoft.Azure.ActiveDirectory \
    --name AADSSHLoginForLinux \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --vm-name $MY_VM_NAME
```

Store IP address of VM to use with SSH

Run the following command to store the IP address of the VM as an environment variable:

Bash

```
export IP_ADDRESS=$(az vm show --show-details --resource-group
$MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME --query publicIps --output tsv)
```

SSH into the VM

You can now SSH into the VM by running the output of the following command in your ssh client of choice:

Bash

```
ssh -o StrictHostKeyChecking=no $MY_USERNAME@$IP_ADDRESS
```

Related content

- Learn about virtual machines
- Use Cloud-Init to initialize a Linux VM on first boot
- Create custom VM images
- Load Balance VMs

Last updated on 03/27/2026

Quickstart: Create a Linux virtual machine in the Azure portal

Summarize this article for me

Applies to: ✔ Linux VMs

Azure virtual machines (VMs) can be created through the Azure portal. The Azure portal is a browser-based user interface to create Azure resources. This quickstart shows you how to use the Azure portal to deploy a Linux virtual machine (VM) running Ubuntu Server 22.04 LTS. To see your VM in action, you also SSH to the VM and install the NGINX web server.

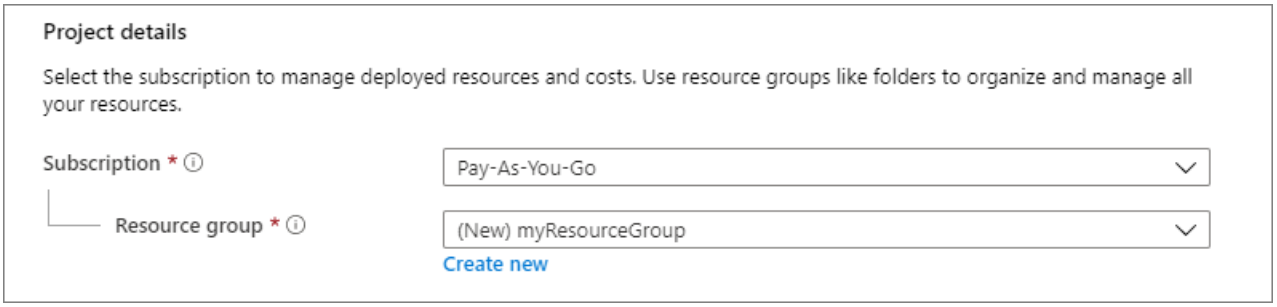
If you don't have an Azure subscription, create a free account before you begin.

Sign in to Azure

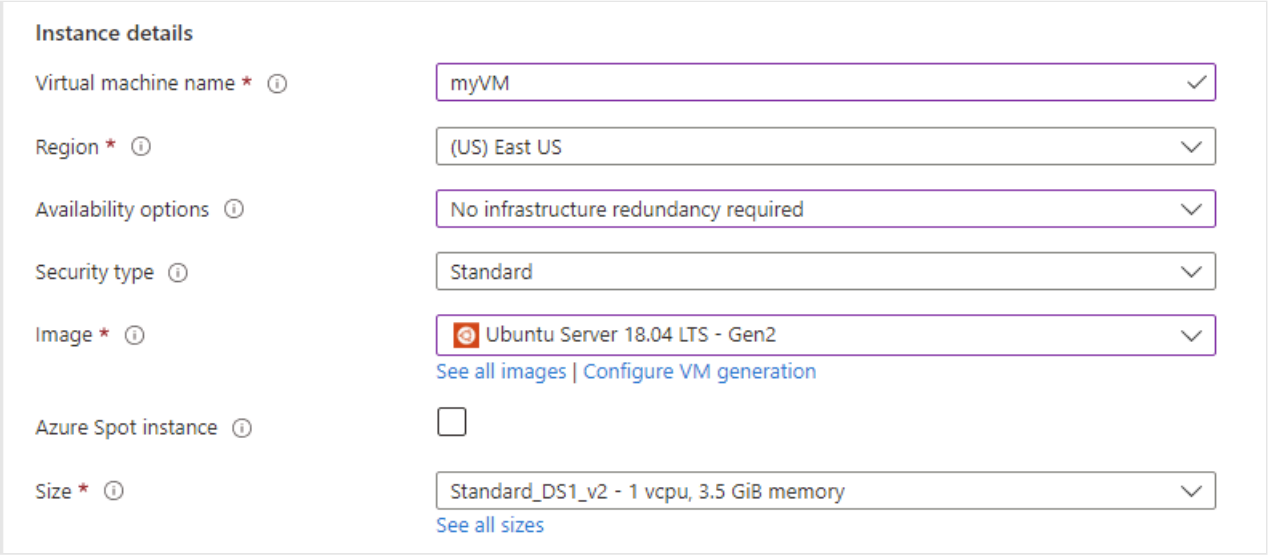
Sign in to the Azure portal.

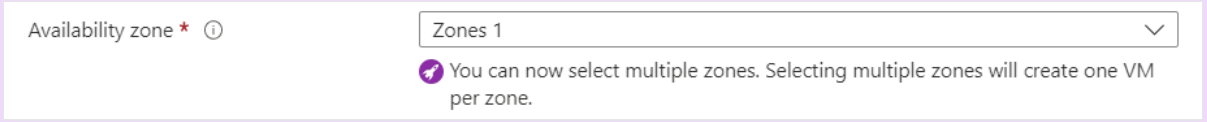
Create virtual machine

1. Enter *virtual machines* in the search.
2. Under **Services**, select **Virtual machines**.
3. In the **Virtual machines** page, select **Create** and then **Virtual machine**. The **Create a virtual machine** page opens.
4. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose to **Create new** resource group. Enter *myResourceGroup* for the name.*.



5. Under **Instance details**, enter *myVM* for the **Virtual machine name**. Under **Availability options**, select *No infrastructure redundancy required*. Under **Security type**, select *Standard*. Choose *Ubuntu Server 22.04 LTS - Gen2* for your **Image**. Leave the other defaults. The default size and pricing is only shown as an example. Size availability and pricing are dependent on your region and subscription.

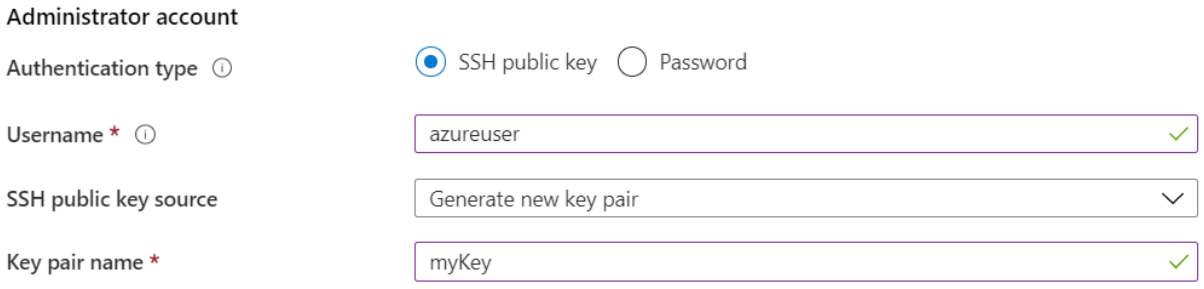


> **Note**
>
> Some users will now see the option to create VMs in multiple zones. To learn more about this new capability, see Create virtual machines in an availability zone.
>
> 
>

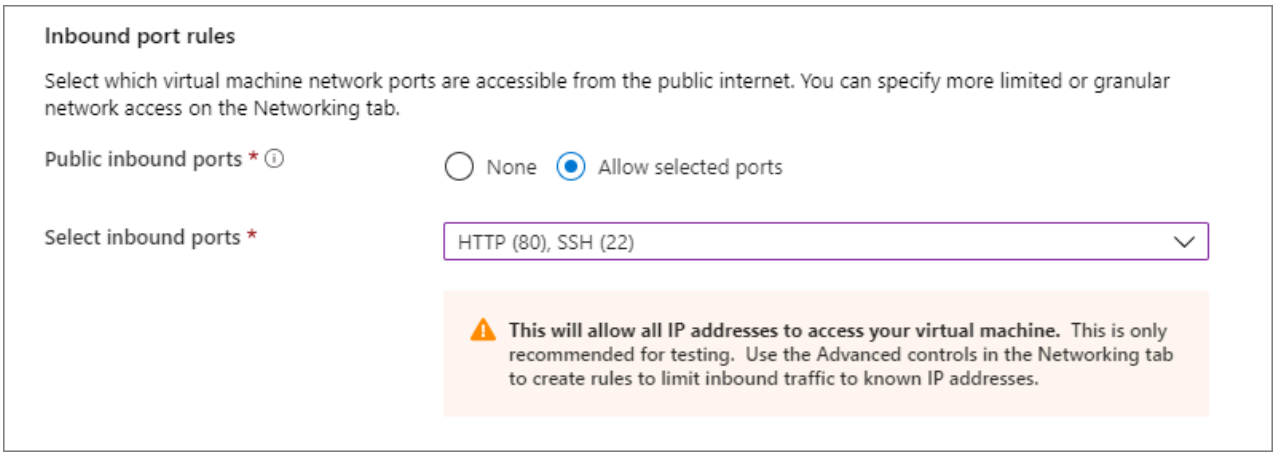
6. Under **Administrator account**, for **Authentication type**, select **SSH public key**.

7. In **Username** enter *azureuser*.

8. For **SSH public key source**, leave the default of **Generate new key pair**, and then enter *myKey* for the **Key pair name**.



9. Under **Inbound port rules** > **Public inbound ports**, choose **Allow selected ports** and then select **SSH (22)** and **HTTP (80)** from the drop-down.



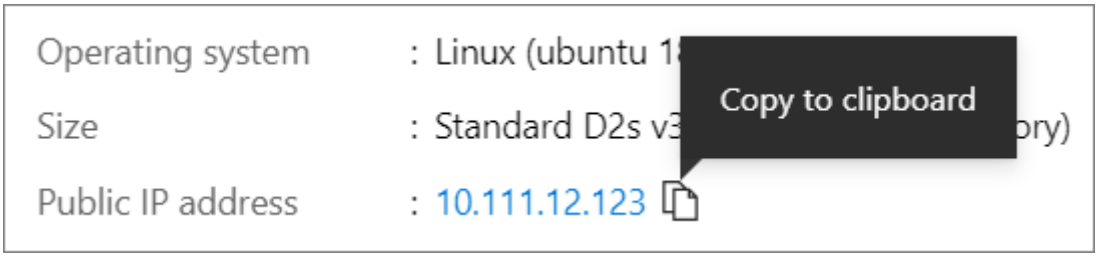
10. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page. A final validation runs.

11. On the **Create a virtual machine** page, review the details about the VM you're about to create. When you're ready, select **Create**.

12. When the **Generate new key pair** window opens, select **Download private key and create resource**. Your key file will be download as **myKey.pem**. Make sure you know where the `.pem` file was downloaded; you'll need the path to it in the next step.

13. When the deployment is finished, select **Go to resource**.

14. On the page for your new VM, select the public IP address and copy it to your clipboard.



Connect to virtual machine

Create an SSH connection with the VM.

1. If you're on a Windows machine, open a PowerShell prompt. If you're on a Mac or Linux machine, open a Bash prompt and set read-only permission on the .pem file using `chmod 400 ~/Downloads/myKey.pem`.

2. At your prompt, open an SSH connection to your virtual machine. Replace the IP address with the one from your VM, and replace the path to the `.pem` with the path to where the key file was downloaded.

Console

```
ssh -i ~/Downloads/myKey.pem azureuser@10.111.12.123
```

> **Tip**
>
> The SSH key you created can be used the next time your create a VM in Azure. Just select the **Use a key stored in Azure** for **SSH public key source** the next time you create a VM. You already have the private key on your computer, so you won't need to download anything.

Install web server

To see your VM in action, install the NGINX web server. From your SSH session, update your package sources and then install the latest NGINX package.

Ubuntu

Bash

```
sudo apt-get -y update
sudo apt-get -y install nginx
```

When done, type `exit` to leave the SSH session.

View the web server in action

Use a web browser of your choice to view the default NGINX welcome page. Type the public IP address of the VM as the web address. The public IP address can be found on the VM overview page or as part of the SSH connection string you used earlier.



Clean up resources

Delete resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources.

1. On the Overview page for the VM, under **Essentials**, select the link to the resource group you created for this Quickstart.
2. At the top of the page for the resource group, select **Delete resource group**.
3. A page will open warning you that you are about to delete resources. Type the name of the resource group, select **Delete**, and confirm deletion to finish deleting the resources and the resource group.

Auto-shutdown

If the VM is still needed, Azure provides an Auto-shutdown feature for virtual machines to help manage costs and ensure you aren't billed for unused resources.

1. On the **Operations** section for the VM, select the **Auto-shutdown** option.
2. A page will open where you can configure the auto-shutdown time. Select the **On** option to enable and then set a time that works for you.
3. Once you have set the time, select **Save** at the top to enable your Auto-shutdown configuration.

> ⓘ **Note**

Remember to configure the time zone correctly to match your requirements, as (UTC) Coordinated Universal Time is the default setting in the Time zone dropdown.

For more information see Auto-shutdown.

Next step

In this quickstart, you deployed a Linux virtual machine and installed a basic web server.

To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Azure Linux virtual machine tutorial

Last updated on 02/06/2026

Quickstart: Create a Linux virtual machine in Azure with PowerShell

Applies to: ✔ Linux VMs

The Azure PowerShell module is used to create and manage Azure resources from the PowerShell command line or in scripts. This quickstart shows you how to use the Azure PowerShell module to deploy a Linux virtual machine (VM) in Azure. This quickstart uses the latest Debian marketplace image. To see your VM in action, you'll also SSH to the VM and install the NGINX web server.

If you don't have an Azure subscription, create a free account before you begin.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create a resource group

Create an Azure resource group with New-AzResourceGroup. A resource group is a logical container into which Azure resources are deployed and managed:

Azure PowerShell

```
New-AzResourceGroup -Name 'myResourceGroup' -Location 'EastUS'
```

Create a virtual machine

We will be automatically generating an SSH key pair to use for connecting to the VM. The public key that is created using `-GenerateSshKey` will be stored in Azure as a resource, using the name you provide as `SshKeyName`. The SSH key resource can be reused for creating additional VMs. Both the public and private keys will also be downloaded for you. When you create your SSH key pair using the Cloud Shell, the keys are stored in a storage account that is automatically created by Cloud Shell. Don't delete the storage account, or the file share in it, until after you have retrieved your keys or you will lose access to the VM.

You will be prompted for a user name that will be used when you connect to the VM. You will also be asked for a password, which you can leave blank. Password log in for the VM is disabled when using an SSH key.

In this example, you create a VM named *myVM*, in *East US*, using the *Standard_B2s* VM size.

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName 'myResourceGroup' `
    -Name 'myVM' `
    -Location 'East US' `
    -image Debian11 `
    -size Standard_B2s `
    -PublicIpAddressName myPubIP `
    -OpenPorts 80 `
    -GenerateSshKey `
    -SshKeyName mySSHKey
```

The output will give you the location of the local copy of the SSH key. For example:

Output

```
Private key is saved to /home/user/.ssh/1234567891
Public key is saved to /home/user/.ssh/1234567891.pub
```

Cost information isn't presented during the virtual machine creation process for PowerShell like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

It will take a few minutes for your VM to be deployed. When the deployment is finished, move on to the next section.

Install NGINX

To see your VM in action, install the NGINX web server.

Azure PowerShell

```
Invoke-AzVMRunCommand `
   -ResourceGroupName 'myResourceGroup' `
   -Name 'myVM' `
   -CommandId 'RunShellScript' `
   -ScriptString 'sudo apt-get update && sudo apt-get install -y nginx'
```

The `-ScriptString` parameter requires version `4.27.0` or later of the `Az.Compute` module.

View the web server in action

Get the public IP address of your VM:

Azure PowerShell

```
Get-AzPublicIpAddress -Name myPubIP -ResourceGroupName myResourceGroup | select
"IpAddress"
```

Use a web browser of your choice to view the default NGINX welcome page. Enter the public IP address of the VM as the web address.



Clean up resources

When no longer needed, you can use the Remove-AzResourceGroup cmdlet to remove the resource group, VM, and all related resources:

Azure PowerShell

```
Remove-AzResourceGroup -Name 'myResourceGroup'
```

Next steps

In this quickstart, you deployed a simple virtual machine, created a Network Security Group and rule, and installed a basic web server. To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Azure Linux virtual machine tutorials

Last updated on 02/06/2026

Quickstart: Use Terraform to create a Linux VM

Article • 08/22/2024

Applies to: ✔ Linux VMs

Article tested with the following Terraform and Terraform provider versions:

This article shows you how to create a complete Linux environment and supporting resources with Terraform. Those resources include a virtual network, subnet, public IP address, and more.

Terraform enables the definition, preview, and deployment of cloud infrastructure. Using Terraform, you create configuration files using HCL syntax. The HCL syntax allows you to specify the cloud provider - such as Azure - and the elements that make up your cloud infrastructure. After you create your configuration files, you create an *execution plan* that allows you to preview your infrastructure changes before they're deployed. Once you verify the changes, you apply the execution plan to deploy the infrastructure.

In this article, you learn how to:

- ✔ Create a random value for the Azure resource group name using random_pet.
- ✔ Create an Azure resource group using azurerm_resource_group.
- ✔ Create a virtual network (VNET) using azurerm_virtual_network.
- ✔ Create a subnet using azurerm_subnet.
- ✔ Create a public IP using azurerm_public_ip.
- ✔ Create a network security group using azurerm_network_security_group.
- ✔ Create a network interface using azurerm_network_interface.
- ✔ Create an association between the network security group and the network interface using azurerm_network_interface_security_group_association.
- ✔ Generate a random value for a unique storage account name using random_id.
- ✔ Create a storage account for boot diagnostics using azurerm_storage_account.
- ✔ Create a Linux VM using azurerm_linux_virtual_machine
- ✔ Create an AzAPI resource azapi_resource.
- ✔ Create an AzAPI resource to generate an SSH key pair using azapi_resource_action.

Prerequisites

- Install and configure Terraform

Implement the Terraform code

> **Note**
>
> The sample code for this article is located in the Azure Terraform GitHub repo. You can view the log file containing the test results from current and previous versions of Terraform.
>
> See more articles and sample code showing how to use Terraform to manage Azure resources

1. Create a directory in which to test the sample Terraform code and make it the current directory.

2. Create a file named `providers.tf` and insert the following code:

```Terraform
terraform {
  required_version = ">=0.12"

  required_providers {
    azapi = {
      source  = "azure/azapi"
      version = "~>1.5"
    }
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~>2.0"
    }
    random = {
      source  = "hashicorp/random"
      version = "~>3.0"
    }
  }
}

provider "azurerm" {
  features {}
}
```

3. Create a file named `ssh.tf` and insert the following code:

```Terraform
```

```
resource "random_pet" "ssh_key_name" {
  prefix    = "ssh"
  separator = ""
}

resource "azapi_resource_action" "ssh_public_key_gen" {
  type        = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  resource_id = azapi_resource.ssh_public_key.id
  action      = "generateKeyPair"
  method      = "POST"

  response_export_values = ["publicKey", "privateKey"]
}

resource "azapi_resource" "ssh_public_key" {
  type      = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  name      = random_pet.ssh_key_name.id
  location  = azurerm_resource_group.rg.location
  parent_id = azurerm_resource_group.rg.id
}

output "key_data" {
  value = azapi_resource_action.ssh_public_key_gen.output.publicKey
}
```

4. Create a file named `main.tf` and insert the following code:

Terraform

```
resource "random_pet" "rg_name" {
  prefix = var.resource_group_name_prefix
}

resource "azurerm_resource_group" "rg" {
  location = var.resource_group_location
  name     = random_pet.rg_name.id
}

# Create virtual network
resource "azurerm_virtual_network" "my_terraform_network" {
  name                = "myVnet"
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
}

# Create subnet
resource "azurerm_subnet" "my_terraform_subnet" {
  name                 = "mySubnet"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name =
azurerm_virtual_network.my_terraform_network.name
```

```
  address_prefixes     = ["10.0.1.0/24"]
}

# Create public IPs
resource "azurerm_public_ip" "my_terraform_public_ip" {
  name                = "myPublicIP"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  allocation_method   = "Dynamic"
}

# Create Network Security Group and rule
resource "azurerm_network_security_group" "my_terraform_nsg" {
  name                = "myNetworkSecurityGroup"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  security_rule {
    name                       = "SSH"
    priority                   = 1001
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "22"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }
}

# Create network interface
resource "azurerm_network_interface" "my_terraform_nic" {
  name                = "myNIC"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                          = "my_nic_configuration"
    subnet_id                     =
azurerm_subnet.my_terraform_subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          =
azurerm_public_ip.my_terraform_public_ip.id
  }
}

# Connect the security group to the network interface
resource "azurerm_network_interface_security_group_association"
"example" {
  network_interface_id      =
azurerm_network_interface.my_terraform_nic.id
  network_security_group_id =
azurerm_network_security_group.my_terraform_nsg.id
}
```

```
# Generate random text for a unique storage account name
resource "random_id" "random_id" {
  keepers = {
    # Generate a new ID only when a new resource group is defined
    resource_group = azurerm_resource_group.rg.name
  }

  byte_length = 8
}

# Create storage account for boot diagnostics
resource "azurerm_storage_account" "my_storage_account" {
  name                     = "diag${random_id.random_id.hex}"
  location                 = azurerm_resource_group.rg.location
  resource_group_name      = azurerm_resource_group.rg.name
  account_tier             = "Standard"
  account_replication_type = "LRS"
}

# Create virtual machine
resource "azurerm_linux_virtual_machine" "my_terraform_vm" {
  name                  = "myVM"
  location              = azurerm_resource_group.rg.location
  resource_group_name   = azurerm_resource_group.rg.name
  network_interface_ids =
[azurerm_network_interface.my_terraform_nic.id]
  size                  = "Standard_DS1_v2"

  os_disk {
    name                 = "myOsDisk"
    caching              = "ReadWrite"
    storage_account_type = "Premium_LRS"
  }

  source_image_reference {
    publisher = "Canonical"
    offer     = "0001-com-ubuntu-server-jammy"
    sku       = "22_04-lts-gen2"
    version   = "latest"
  }

  computer_name  = "hostname"
  admin_username = var.username

  admin_ssh_key {
    username   = var.username
    public_key =
azapi_resource_action.ssh_public_key_gen.output.publicKey
  }

  boot_diagnostics {
    storage_account_uri =
azurerm_storage_account.my_storage_account.primary_blob_endpoint
  }
}
```

5. Create a file named `variables.tf` and insert the following code:

Terraform

```
variable "resource_group_location" {
  type        = string
  default     = "eastus"
  description = "Location of the resource group."
}

variable "resource_group_name_prefix" {
  type        = string
  default     = "rg"
  description = "Prefix of the resource group name that's combined with
a random ID so name is unique in your Azure subscription."
}

variable "username" {
  type        = string
  description = "The username for the local account that will be
created on the new VM."
  default     = "azureadmin"
}
```

6. Create a file named `outputs.tf` and insert the following code:

Terraform

```
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}

output "public_ip_address" {
  value =
azurerm_linux_virtual_machine.my_terraform_vm.public_ip_address
}
```

Initialize Terraform

Run terraform init to initialize the Terraform deployment. This command downloads the Azure provider required to manage your Azure resources.

Console

```
terraform init -upgrade
```

Key points:

- The `-upgrade` parameter upgrades the necessary provider plugins to the newest version that complies with the configuration's version constraints.

Create a Terraform execution plan

Run terraform plan to create an execution plan.

Console

```
terraform plan -out main.tfplan
```

Key points:

- The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

Apply a Terraform execution plan

Run terraform apply to apply the execution plan to your cloud infrastructure.

Console

```
terraform apply main.tfplan
```

Key points:

- The example `terraform apply` command assumes you previously ran `terraform plan -out main.tfplan`.
- If you specified a different filename for the `-out` parameter, use that same filename in the call to `terraform apply`.
- If you didn't use the `-out` parameter, call `terraform apply` without any parameters.

Cost information isn't presented during the virtual machine creation process for Terraform like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Verify the results

Azure CLI

1. Get the Azure resource group name.

   Console

   ```
   resource_group_name=$(terraform output -raw resource_group_name)
   ```

2. Run az vm list with a JMESPath query to display the names of the virtual machines created in the resource group.

   Azure CLI

   ```
   az vm list \
     --resource-group $resource_group_name \
     --query "[].{\"VM Name\":name}" -o table
   ```

Clean up resources

When you no longer need the resources created via Terraform, do the following steps:

1. Run terraform plan and specify the `destroy` flag.

   Console

   ```
   terraform plan -destroy -out main.destroy.tfplan
   ```

   Key points:

   - The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.

- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

2. Run terraform apply to apply the execution plan.

```
Console

terraform apply main.destroy.tfplan
```

Troubleshoot Terraform on Azure

Troubleshoot common problems when using Terraform on Azure

Next steps

In this quickstart, you deployed a simple virtual machine using Terraform. To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Azure Linux virtual machine tutorials

ⓘ **Note:** The author created this article with assistance from AI. Learn more

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a Linux VM cluster in Azure using Terraform

Article • 08/22/2024

Applies to: ✔ Linux VMs

This article shows you how to create a Linux VM cluster (containing two Linux VM instances) in Azure using Terraform.

In this article, you learn how to:

- ✔ Create a random value for the Azure resource group name using random_pet.
- ✔ Create an Azure resource group using azurerm_resource_group.
- ✔ Create a virtual network using azurerm_virtual_network
- ✔ Create a subnet using azurerm_subnet
- ✔ Create a public IP using azurerm_public_ip
- ✔ Create a load balancer using azurerm_lb
- ✔ Create a load balancer address pool using azurerm_lb_backend_address_pool
- ✔ Create a network interface using azurerm_network_interface
- ✔ Create a managed disk using azurerm_managed_disk
- ✔ Create a availability set using azurerm_availability_set
- ✔ Create a Linux virtual machine using azurerm_linux_virtual_machine
- ✔ Create an AzAPI resource azapi_resource.
- ✔ Create an AzAPI resource to generate an SSH key pair using azapi_resource_action.

Prerequisites

- Install and configure Terraform

Implement the Terraform code

> ⓘ **Note**
>
> The sample code for this article is located in the Azure Terraform GitHub repo. You can view the log file containing the test results from current and previous versions of Terraform.

See more [articles and sample code showing how to use Terraform to manage Azure resources]

1. Create a directory in which to test the sample Terraform code and make it the current directory.

2. Create a file named `providers.tf` and insert the following code:

```Terraform
terraform {
  required_version = ">=1.0"
  required_providers {
    azapi = {
      source  = "azure/azapi"
      version = "~>1.5"
    }
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~>3.0"
    }
    random = {
      source  = "hashicorp/random"
      version = "~>3.0"
    }
  }
}
provider "azurerm" {
  features {}
}
```

3. Create a file named `ssh.tf` and insert the following code:

```Terraform
resource "random_pet" "ssh_key_name" {
  prefix    = "ssh"
  separator = ""
}

resource "azapi_resource_action" "ssh_public_key_gen" {
  type        = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  resource_id = azapi_resource.ssh_public_key.id
  action      = "generateKeyPair"
  method      = "POST"

  response_export_values = ["publicKey", "privateKey"]
}

resource "azapi_resource" "ssh_public_key" {
```

```
  type      = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  name      = random_pet.ssh_key_name.id
  location  = azurerm_resource_group.rg.location
  parent_id = azurerm_resource_group.rg.id
}

output "key_data" {
  value = azapi_resource_action.ssh_public_key_gen.output.publicKey
}
```

4. Create a file named `main.tf` and insert the following code:

Terraform

```
resource "random_pet" "rg_name" {
  prefix = var.resource_group_name_prefix
}

resource "azurerm_resource_group" "rg" {
  name     = random_pet.rg_name.id
  location = var.resource_group_location
}

resource "random_pet" "azurerm_virtual_network_name" {
  prefix = "vnet"
}

resource "azurerm_virtual_network" "test" {
  name                = random_pet.azurerm_virtual_network_name.id
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
}

resource "random_pet" "azurerm_subnet_name" {
  prefix = "sub"
}

resource "azurerm_subnet" "test" {
  name                 = random_pet.azurerm_subnet_name.id
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.test.name
  address_prefixes     = ["10.0.2.0/24"]
}

resource "azurerm_public_ip" "test" {
  name                = "publicIPForLB"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  allocation_method   = "Static"
}

resource "azurerm_lb" "test" {
```

```
  name                = "loadBalancer"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  frontend_ip_configuration {
    name                 = "publicIPAddress"
    public_ip_address_id = azurerm_public_ip.test.id
  }
}

resource "azurerm_lb_backend_address_pool" "test" {
  loadbalancer_id = azurerm_lb.test.id
  name            = "BackEndAddressPool"
}

resource "azurerm_network_interface" "test" {
  count               = 2
  name                = "acctni${count.index}"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                          = "testConfiguration"
    subnet_id                     = azurerm_subnet.test.id
    private_ip_address_allocation = "Dynamic"
  }
}

resource "azurerm_availability_set" "avset" {
  name                         = "avset"
  location                     = azurerm_resource_group.rg.location
  resource_group_name          = azurerm_resource_group.rg.name
  platform_fault_domain_count  = 2
  platform_update_domain_count = 2
  managed                      = true
}

resource "random_pet" "azurerm_linux_virtual_machine_name" {
  prefix = "vm"
}

resource "azurerm_linux_virtual_machine" "test" {
  count                 = 2
  name                  =
"${random_pet.azurerm_linux_virtual_machine_name.id}${count.index}"
  location              = azurerm_resource_group.rg.location
  availability_set_id   = azurerm_availability_set.avset.id
  resource_group_name   = azurerm_resource_group.rg.name
  network_interface_ids =
[azurerm_network_interface.test[count.index].id]
  size                  = "Standard_DS1_v2"

  # Uncomment this line to delete the OS disk automatically when
deleting the VM
  # delete_os_disk_on_termination = true
```

```
  # Uncomment this line to delete the data disks automatically when
deleting the VM
  # delete_data_disks_on_termination = true

  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }

  admin_ssh_key {
    username   = var.username
    public_key =
azapi_resource_action.ssh_public_key_gen.output.publicKey
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
    name                 = "myosdisk${count.index}"
  }

  computer_name  = "hostname"
  admin_username = var.username
}

resource "azurerm_managed_disk" "test" {
  count                = 2
  name                 = "datadisk_existing_${count.index}"
  location             = azurerm_resource_group.rg.location
  resource_group_name  = azurerm_resource_group.rg.name
  storage_account_type = "Standard_LRS"
  create_option        = "Empty"
  disk_size_gb         = "1024"
}

resource "azurerm_virtual_machine_data_disk_attachment" "test" {
  count              = 2
  managed_disk_id    = azurerm_managed_disk.test[count.index].id
  virtual_machine_id =
azurerm_linux_virtual_machine.test[count.index].id
  lun                = "10"
  caching            = "ReadWrite"
}
```

5. Create a file named `variables.tf` and insert the following code:

Terraform

```
variable "resource_group_location" {
  type        = string
```

```
  description = "Location for all resources."
  default     = "eastus"
}

variable "resource_group_name_prefix" {
  type        = string
  description = "Prefix of the resource group name that's combined with
a random ID so name is unique in your Azure subscription."
  default     = "rg"
}

variable "username" {
  type        = string
  description = "The username for the local account that will be
created on the new VM."
  default     = "azureadmin"
}
```

6. Create a file named `outputs.tf` and insert the following code:

Terraform

```
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}

output "virtual_network_name" {
  value = azurerm_virtual_network.test.name
}

output "subnet_name" {
  value = azurerm_subnet.test.name
}

output "linux_virtual_machine_names" {
  value = [for s in azurerm_linux_virtual_machine.test : s.name[*]]
}
```

Initialize Terraform

Run terraform init to initialize the Terraform deployment. This command downloads the Azure provider required to manage your Azure resources.

Console

```
terraform init -upgrade
```

Key points:

- The `-upgrade` parameter upgrades the necessary provider plugins to the newest version that complies with the configuration's version constraints.

Create a Terraform execution plan

Run terraform plan to create an execution plan.

Console

```
terraform plan -out main.tfplan
```

Key points:

- The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

Apply a Terraform execution plan

Run terraform apply to apply the execution plan to your cloud infrastructure.

Console

```
terraform apply main.tfplan
```

Key points:

- The example `terraform apply` command assumes you previously ran `terraform plan -out main.tfplan`.
- If you specified a different filename for the `-out` parameter, use that same filename in the call to `terraform apply`.
- If you didn't use the `-out` parameter, call `terraform apply` without any parameters.

Cost information isn't presented during the virtual machine creation process for Terraform like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Verify the results

Azure CLI

1. Get the Azure resource group name.

   Console

   ```
   resource_group_name=$(terraform output -raw resource_group_name)
   ```

2. Run az vm list with a JMESPath query to display the names of the virtual machines created in the resource group.

   Azure CLI

   ```
   az vm list \
     --resource-group $resource_group_name \
     --query "[].{\"VM Name\":name}" -o table
   ```

Clean up resources

When you no longer need the resources created via Terraform, do the following steps:

1. Run terraform plan and specify the `destroy` flag.

   Console

   ```
   terraform plan -destroy -out main.destroy.tfplan
   ```

   Key points:

   - The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
   - The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

2. Run terraform apply to apply the execution plan.

```
Console

terraform apply main.destroy.tfplan
```

Troubleshoot Terraform on Azure

Troubleshoot common problems when using Terraform on Azure

Next steps

Azure Linux virtual machine tutorials

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create an Ubuntu Linux virtual machine using a Bicep file

Article • 08/22/2024

Applies to: ✔ Linux VMs

This quickstart shows you how to use a Bicep file to deploy an Ubuntu Linux virtual machine (VM) in Azure.

Bicep is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. It provides concise syntax, reliable type safety, and support for code reuse. Bicep offers the best authoring experience for your infrastructure-as-code solutions in Azure.

Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

Review the Bicep file

The Bicep file used in this quickstart is from Azure Quickstart Templates.

Bicep

```
@description('The name of your Virtual Machine.')
param vmName string = 'simpleLinuxVM'

@description('Username for the Virtual Machine.')
param adminUsername string

@description('Type of authentication to use on the Virtual Machine. SSH key
is recommended.')
@allowed([
  'sshPublicKey'
  'password'
])
param authenticationType string = 'password'

@description('SSH Key or password for the Virtual Machine. SSH key is
recommended.')
@secure()
param adminPasswordOrKey string

@description('Unique DNS Name for the Public IP used to access the Virtual
Machine.')
```

```
param dnsLabelPrefix string =
toLower('${vmName}-${uniqueString(resourceGroup().id)}')

@description('The Ubuntu version for the VM. This will pick a fully patched
image of this given Ubuntu version.')
@allowed([
  'Ubuntu-2004'
  'Ubuntu-2204'
])
param ubuntuOSVersion string = 'Ubuntu-2004'

@description('Location for all resources.')
param location string = resourceGroup().location

@description('The size of the VM')
param vmSize string = 'Standard_D2s_v3'

@description('Name of the VNET')
param virtualNetworkName string = 'vNet'

@description('Name of the subnet in the virtual network')
param subnetName string = 'Subnet'

@description('Name of the Network Security Group')
param networkSecurityGroupName string = 'SecGroupNet'

@description('Security Type of the Virtual Machine.')
@allowed([
  'Standard'
  'TrustedLaunch'
])
param securityType string = 'TrustedLaunch'

var imageReference = {
  'Ubuntu-2004': {
    publisher: 'Canonical'
    offer: '0001-com-ubuntu-server-focal'
    sku: '20_04-lts-gen2'
    version: 'latest'
  }
  'Ubuntu-2204': {
    publisher: 'Canonical'
    offer: '0001-com-ubuntu-server-jammy'
    sku: '22_04-lts-gen2'
    version: 'latest'
  }
}
var publicIPAddressName = '${vmName}PublicIP'
var networkInterfaceName = '${vmName}NetInt'
var osDiskType = 'Standard_LRS'
var subnetAddressPrefix = '10.1.0.0/24'
var addressPrefix = '10.1.0.0/16'
var linuxConfiguration = {
  disablePasswordAuthentication: true
  ssh: {
```

```
    publicKeys: [
      {
        path: '/home/${adminUsername}/.ssh/authorized_keys'
        keyData: adminPasswordOrKey
      }
    ]
  }
}
var securityProfileJson = {
  uefiSettings: {
    secureBootEnabled: true
    vTpmEnabled: true
  }
  securityType: securityType
}
var extensionName = 'GuestAttestation'
var extensionPublisher = 'Microsoft.Azure.Security.LinuxAttestation'
var extensionVersion = '1.0'
var maaTenantName = 'GuestAttestation'
var maaEndpoint = substring('emptystring', 0, 0)

resource networkInterface 'Microsoft.Network/networkInterfaces@2023-09-01' =
{
  name: networkInterfaceName
  location: location
  properties: {
    ipConfigurations: [
      {
        name: 'ipconfig1'
        properties: {
          subnet: {
            id: virtualNetwork.properties.subnets[0].id
          }
          privateIPAllocationMethod: 'Dynamic'
          publicIPAddress: {
            id: publicIPAddress.id
          }
        }
      }
    ]
    networkSecurityGroup: {
      id: networkSecurityGroup.id
    }
  }
}

resource networkSecurityGroup 'Microsoft.Network/networkSecurityGroups@2023-
09-01' = {
  name: networkSecurityGroupName
  location: location
  properties: {
    securityRules: [
      {
        name: 'SSH'
        properties: {
```

```
          priority: 1000
          protocol: 'Tcp'
          access: 'Allow'
          direction: 'Inbound'
          sourceAddressPrefix: '*'
          sourcePortRange: '*'
          destinationAddressPrefix: '*'
          destinationPortRange: '22'
        }
      }
    ]
  }
}

resource virtualNetwork 'Microsoft.Network/virtualNetworks@2023-09-01' = {
  name: virtualNetworkName
  location: location
  properties: {
    addressSpace: {
      addressPrefixes: [
        addressPrefix
      ]
    }
    subnets: [
      {
        name: subnetName
        properties: {
          addressPrefix: subnetAddressPrefix
          privateEndpointNetworkPolicies: 'Enabled'
          privateLinkServiceNetworkPolicies: 'Enabled'
        }
      }
    ]
  }
}

resource publicIPAddress 'Microsoft.Network/publicIPAddresses@2023-09-01' =
{
  name: publicIPAddressName
  location: location
  sku: {
    name: 'Basic'
  }
  properties: {
    publicIPAllocationMethod: 'Dynamic'
    publicIPAddressVersion: 'IPv4'
    dnsSettings: {
      domainNameLabel: dnsLabelPrefix
    }
    idleTimeoutInMinutes: 4
  }
}

resource vm 'Microsoft.Compute/virtualMachines@2023-09-01' = {
  name: vmName
```

```
  location: location
  properties: {
    hardwareProfile: {
      vmSize: vmSize
    }
    storageProfile: {
      osDisk: {
        createOption: 'FromImage'
        managedDisk: {
          storageAccountType: osDiskType
        }
      }
      imageReference: imageReference[ubuntuOSVersion]
    }
    networkProfile: {
      networkInterfaces: [
        {
          id: networkInterface.id
        }
      ]
    }
    osProfile: {
      computerName: vmName
      adminUsername: adminUsername
      adminPassword: adminPasswordOrKey
      linuxConfiguration: ((authenticationType == 'password') ? null :
linuxConfiguration)
    }
    securityProfile: (securityType == 'TrustedLaunch') ? securityProfileJson
: null
  }
}

resource vmExtension 'Microsoft.Compute/virtualMachines/extensions@2023-09-
01' = if (securityType == 'TrustedLaunch' &&
securityProfileJson.uefiSettings.secureBootEnabled &&
securityProfileJson.uefiSettings.vTpmEnabled) {
  parent: vm
  name: extensionName
  location: location
  properties: {
    publisher: extensionPublisher
    type: extensionName
    typeHandlerVersion: extensionVersion
    autoUpgradeMinorVersion: true
    enableAutomaticUpgrade: true
    settings: {
      AttestationConfig: {
        MaaSettings: {
          maaEndpoint: maaEndpoint
          maaTenantName: maaTenantName
        }
      }
    }
  }
```

```
}

output adminUsername string = adminUsername
output hostname string = publicIPAddress.properties.dnsSettings.fqdn
output sshCommand string = 'ssh
${adminUsername}@${publicIPAddress.properties.dnsSettings.fqdn}'
```

Several resources are defined in the Bicep file:

- Microsoft.Network/virtualNetworks/subnets: create a subnet.
- Microsoft.Storage/storageAccounts: create a storage account.
- Microsoft.Network/networkInterfaces: create a NIC.
- Microsoft.Network/networkSecurityGroups: create a network security group.
- Microsoft.Network/virtualNetworks: create a virtual network.
- Microsoft.Network/publicIPAddresses: create a public IP address.
- Microsoft.Compute/virtualMachines: create a virtual machine.

Deploy the Bicep file

1. Save the Bicep file as **main.bicep** to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell.

   CLI

   Azure CLI

   ```
   az group create --name exampleRG --location eastus

   az deployment group create --resource-group exampleRG --template-
   file main.bicep --parameters adminUsername=<admin-username>
   ```

   > **Note**
   >
   > Replace **<admin-username>** with a unique username. You'll also be prompted to enter adminPasswordOrKey.

   When the deployment finishes, you should see a message indicating the deployment succeeded.

Cost information isn't presented during the virtual machine creation process for Bicep like it is for the Azure portal. If you want to learn more about how cost works for virtual

machines, see the Cost optimization Overview page.

Review deployed resources

Use the Azure portal, Azure CLI, or Azure PowerShell to list the deployed resources in the resource group.

CLI

Azure CLI

```azurecli
az resource list --resource-group exampleRG
```

Clean up resources

When no longer needed, use the Azure portal, Azure CLI, or Azure PowerShell to delete the VM and all of the resources in the resource group.

CLI

Azure CLI

```azurecli
az group delete --name exampleRG
```

Next steps

In this quickstart, you deployed a simple virtual machine using a Bicep file. To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Azure Linux virtual machine tutorials

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create an Ubuntu Linux virtual machine by using an ARM template

Article • 08/22/2024

Applies to: ✔ Linux VMs

This quickstart shows you how to use an Azure Resource Manager template (ARM template) to deploy an Ubuntu Linux virtual machine (VM) in Azure.

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with ARM templates, select the **Deploy to Azure** button. The template opens in the Azure portal.



Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

Review the template

For more information on this template, see Deploy a simple Ubuntu Linux VM 18.04-LTS.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "metadata": {
    "_generator": {
      "name": "bicep",
      "version": "0.25.53.49325",
      "templateHash": "2386182166504519146"
    }
  },
```

```
  "parameters": {
    "vmName": {
      "type": "string",
      "defaultValue": "simpleLinuxVM",
      "metadata": {
        "description": "The name of your Virtual Machine."
      }
    },
    "adminUsername": {
      "type": "string",
      "metadata": {
        "description": "Username for the Virtual Machine."
      }
    },
    "authenticationType": {
      "type": "string",
      "defaultValue": "password",
      "allowedValues": [
        "sshPublicKey",
        "password"
      ],
      "metadata": {
        "description": "Type of authentication to use on the Virtual
Machine. SSH key is recommended."
      }
    },
    "adminPasswordOrKey": {
      "type": "securestring",
      "metadata": {
        "description": "SSH Key or password for the Virtual Machine. SSH key
is recommended."
      }
    },
    "dnsLabelPrefix": {
      "type": "string",
      "defaultValue": "[toLower(format('{0}-{1}', parameters('vmName'),
uniqueString(resourceGroup().id)))]",
      "metadata": {
        "description": "Unique DNS Name for the Public IP used to access the
Virtual Machine."
      }
    },
    "ubuntuOSVersion": {
      "type": "string",
      "defaultValue": "Ubuntu-2004",
      "allowedValues": [
        "Ubuntu-2004",
        "Ubuntu-2204"
      ],
      "metadata": {
        "description": "The Ubuntu version for the VM. This will pick a
fully patched image of this given Ubuntu version."
      }
    },
    "location": {
```

```
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Location for all resources."
      }
    },
    "vmSize": {
      "type": "string",
      "defaultValue": "Standard_D2s_v3",
      "metadata": {
        "description": "The size of the VM"
      }
    },
    "virtualNetworkName": {
      "type": "string",
      "defaultValue": "vNet",
      "metadata": {
        "description": "Name of the VNET"
      }
    },
    "subnetName": {
      "type": "string",
      "defaultValue": "Subnet",
      "metadata": {
        "description": "Name of the subnet in the virtual network"
      }
    },
    "networkSecurityGroupName": {
      "type": "string",
      "defaultValue": "SecGroupNet",
      "metadata": {
        "description": "Name of the Network Security Group"
      }
    },
    "securityType": {
      "type": "string",
      "defaultValue": "TrustedLaunch",
      "allowedValues": [
        "Standard",
        "TrustedLaunch"
      ],
      "metadata": {
        "description": "Security Type of the Virtual Machine."
      }
    }
  },
  "variables": {
    "imageReference": {
      "Ubuntu-2004": {
        "publisher": "Canonical",
        "offer": "0001-com-ubuntu-server-focal",
        "sku": "20_04-lts-gen2",
        "version": "latest"
      },
      "Ubuntu-2204": {
```

```
        "publisher": "Canonical",
        "offer": "0001-com-ubuntu-server-jammy",
        "sku": "22_04-lts-gen2",
        "version": "latest"
      }
    },
    "publicIPAddressName": "[format('{0}PublicIP', parameters('vmName'))]",
    "networkInterfaceName": "[format('{0}NetInt', parameters('vmName'))]",
    "osDiskType": "Standard_LRS",
    "subnetAddressPrefix": "10.1.0.0/24",
    "addressPrefix": "10.1.0.0/16",
    "linuxConfiguration": {
      "disablePasswordAuthentication": true,
      "ssh": {
        "publicKeys": [
          {
            "path": "[format('/home/{0}/.ssh/authorized_keys',
parameters('adminUsername'))]",
            "keyData": "[parameters('adminPasswordOrKey')]"
          }
        ]
      }
    },
    "securityProfileJson": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "[parameters('securityType')]"
    },
    "extensionName": "GuestAttestation",
    "extensionPublisher": "Microsoft.Azure.Security.LinuxAttestation",
    "extensionVersion": "1.0",
    "maaTenantName": "GuestAttestation",
    "maaEndpoint": "[substring('emptystring', 0, 0)]"
  },
  "resources": [
    {
      "type": "Microsoft.Network/networkInterfaces",
      "apiVersion": "2023-09-01",
      "name": "[variables('networkInterfaceName')]",
      "location": "[parameters('location')]",
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "subnet": {
                "id": "
[reference(resourceId('Microsoft.Network/virtualNetworks',
parameters('virtualNetworkName')), '2023-09-01').subnets[0].id]"
              },
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": {
                "id": "[resourceId('Microsoft.Network/publicIPAddresses',
```

```
variables('publicIPAddressName'))]"
              }
            }
          }
        ],
        "networkSecurityGroup": {
          "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
parameters('networkSecurityGroupName'))]"
        }
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups',
parameters('networkSecurityGroupName'))]",
        "[resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName'))]",
        "[resourceId('Microsoft.Network/virtualNetworks',
parameters('virtualNetworkName'))]"
      ]
    },
    {
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2023-09-01",
      "name": "[parameters('networkSecurityGroupName')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "SSH",
            "properties": {
              "priority": 1000,
              "protocol": "Tcp",
              "access": "Allow",
              "direction": "Inbound",
              "sourceAddressPrefix": "*",
              "sourcePortRange": "*",
              "destinationAddressPrefix": "*",
              "destinationPortRange": "22"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2023-09-01",
      "name": "[parameters('virtualNetworkName')]",
      "location": "[parameters('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets": [
          {
```

```
            "name": "[parameters('subnetName')]",
            "properties": {
              "addressPrefix": "[variables('subnetAddressPrefix')]",
              "privateEndpointNetworkPolicies": "Enabled",
              "privateLinkServiceNetworkPolicies": "Enabled"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/publicIPAddresses",
      "apiVersion": "2023-09-01",
      "name": "[variables('publicIPAddressName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "Basic"
      },
      "properties": {
        "publicIPAllocationMethod": "Dynamic",
        "publicIPAddressVersion": "IPv4",
        "dnsSettings": {
          "domainNameLabel": "[parameters('dnsLabelPrefix')]"
        },
        "idleTimeoutInMinutes": 4
      }
    },
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2023-09-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "hardwareProfile": {
          "vmSize": "[parameters('vmSize')]"
        },
        "storageProfile": {
          "osDisk": {
            "createOption": "FromImage",
            "managedDisk": {
              "storageAccountType": "[variables('osDiskType')]"
            }
          },
          "imageReference": "[variables('imageReference')
[parameters('ubuntuOSVersion')]]"
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
            }
          ]
        },
        "osProfile": {
```

```
          "computerName": "[parameters('vmName')]",
          "adminUsername": "[parameters('adminUsername')]",
          "adminPassword": "[parameters('adminPasswordOrKey')]",
          "linuxConfiguration": "
[if(equals(parameters('authenticationType'), 'password'), null(),
variables('linuxConfiguration'))]"
        },
        "securityProfile": "[if(equals(parameters('securityType'),
'TrustedLaunch'), variables('securityProfileJson'), null())]"
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
      ]
    },
    {
      "condition": "[and(and(equals(parameters('securityType'),
'TrustedLaunch'),
variables('securityProfileJson').uefiSettings.secureBootEnabled),
variables('securityProfileJson').uefiSettings.vTpmEnabled)]",
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "apiVersion": "2023-09-01",
      "name": "[format('{0}/{1}', parameters('vmName'),
variables('extensionName'))]",
      "location": "[parameters('location')]",
      "properties": {
        "publisher": "[variables('extensionPublisher')]",
        "type": "[variables('extensionName')]",
        "typeHandlerVersion": "[variables('extensionVersion')]",
        "autoUpgradeMinorVersion": true,
        "enableAutomaticUpgrade": true,
        "settings": {
          "AttestationConfig": {
            "MaaSettings": {
              "maaEndpoint": "[variables('maaEndpoint')]",
              "maaTenantName": "[variables('maaTenantName')]"
            }
          }
        }
      },
      "dependsOn": [
        "[resourceId('Microsoft.Compute/virtualMachines',
parameters('vmName'))]"
      ]
    }
  ],
  "outputs": {
    "adminUsername": {
      "type": "string",
      "value": "[parameters('adminUsername')]"
    },
    "hostname": {
      "type": "string",
      "value": "[reference(resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName')), '2023-09-01').dnsSettings.fqdn]"
```

```
    },
    "sshCommand": {
      "type": "string",
      "value": "[format('ssh {0}@{1}', parameters('adminUsername'),
reference(resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName')), '2023-09-01').dnsSettings.fqdn)]"
    }
  }
}
```

Several resources are defined in the template:

- Microsoft.Network/virtualNetworks/subnets: create a subnet.
- Microsoft.Storage/storageAccounts: create a storage account.
- Microsoft.Network/networkInterfaces: create a NIC.
- Microsoft.Network/networkSecurityGroups: create a network security group.
- Microsoft.Network/virtualNetworks: create a virtual network.
- Microsoft.Network/publicIPAddresses: create a public IP address.
- Microsoft.Compute/virtualMachines: create a virtual machine.

Deploy the template

1. Select the following image to sign in to Azure and open a template. The template creates a key vault and a secret.

   Deploy to Azure

2. Select or enter the following values. Use the default values, when available.

   - **Subscription**: select an Azure subscription.
   - **Resource group**: select an existing resource group from the drop-down, or select **Create new**, enter a unique name for the resource group, and select **OK**.
   - **Region**: select a region. For example, **Central US**.
   - **Admin username**: provide a username, such as *azureuser*.
   - **Authentication type**: You can choose between an SSH key or a password.
   - **Admin Password Or Key** depending on what you choose for authentication type:
     - If you choose **password**, the password must be at least 12 characters long and meet the defined complexity requirements.
     - If you choose **sshPublicKey**, paste in the contents of your public key.
   - **DNS label prefix**: enter a unique identifier to use as part of the DNS label.

- **Ubuntu OS version**: select which version of Ubuntu you want to run on the VM.
- **Location**: the default is the same location as the resource group, if it already exists.
- **VM size**: select the size to use for the VM.
- **Virtual Network Name**: name to be used for the vNet.
- **Subnet Name**: name for the subnet the VM should use.
- **Network Security Group Name**: name for the NSG.

3. Select **Review + create**. After validation completes, select **Create** to create and deploy the VM.

Cost information isn't presented during the virtual machine creation process for ARM templates like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

The Azure portal is used to deploy the template. In addition to the Azure portal, you can also use the Azure CLI, Azure PowerShell, and REST API. To learn other deployment methods, see Deploy templates.

Review deployed resources

You can use the Azure portal to check on the VM and other resource that were created. After the deployment is finished, select **Resource groups** to see the VM and other resources.

Clean up resources

When no longer needed, delete the resource group, which deletes the VM and all of the resources in the resource group.

1. Select the **Resource group**.
2. On the page for the resource group, select **Delete resource group**.
3. When prompted, type the name of the resource group and then select **Delete**.

Next steps

In this quickstart, you deployed a virtual machine by using an ARM template. To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Create and Manage Linux VMs with the Azure CLI

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a Windows virtual machine with the Azure CLI

Applies to: ✔ Windows VMs

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This quickstart shows you how to use the Azure CLI to deploy a virtual machine (VM) in Azure that runs Windows Server 2022. To see your VM in action, you then RDP to the VM and install the IIS web server.

If you don't have an Azure subscription, create a free account before you begin.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/bash. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Create a resource group

Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *Central US* location. Replace the value of the variables as needed.

Azure CLI

```
subscription="SubscriptionIDName"
resourcegroup="myResourceGroupCLI"
location="centralus"
az account set --subscription $subscription
az group create --name $resourcegroup --location $location
```

Create virtual machine

Create a VM with az vm create. The following example creates a VM named *myVM*. This example uses *azureuser* for an administrative user name. Replace the values of the variables as needed.

You'll be prompted to supply a password that meets the password requirements for Azure VMs.

Using the example below, you'll be prompted to enter a password at the command line. You could also add the `--admin-password` parameter with a value for your password. The user name and password will be used when you connect to the VM.

Azure CLI

```
vmname="myVM"
username="azureuser"
az vm create \
    --resource-group $resourcegroup \
    --name $vmname \
    --image Win2022AzureEditionCore \
    --public-ip-sku Standard \
    --admin-username $username
```

It takes a few minutes to create the VM and supporting resources. The following example output shows the VM create operation was successful.

Output

```
{
  "fqdns": "",
  "id":
"/subscriptions/<guid>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/v
irtualMachines/myVM",
  "location": "centralus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.174.34.95",
  "resourceGroup": "myResourceGroupCLI"
  "zones": ""
}
```

Take a note your own `publicIpAddress` in the output when you create your VM. This IP address is used to access the VM later in this article.

Cost information isn't presented during the virtual machine creation process for CLI like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Install web server

To see your VM in action, install the IIS web server.

Azure CLI

```
az vm run-command invoke -g $resourcegroup \
   -n $vmname \
   --command-id RunPowerShellScript \
   --scripts "Install-WindowsFeature -name Web-Server -IncludeManagementTools"
```

Open port 80 for web traffic

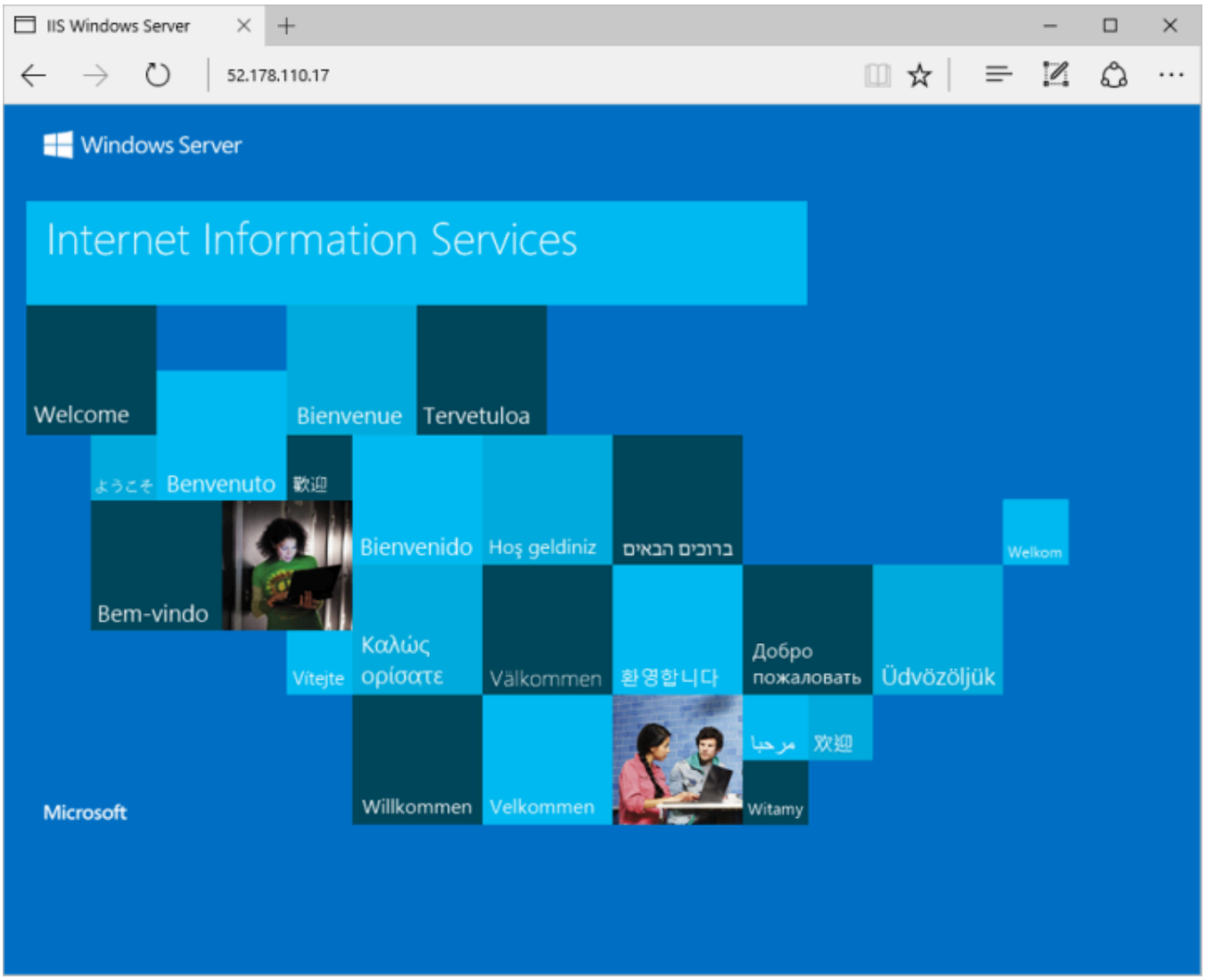
By default, only RDP connections are opened when you create a Windows VM in Azure. Use az vm open-port to open TCP port 80 for use with the IIS web server:

Azure CLI

```
az vm open-port --port 80 --resource-group $resourcegroup --name $vmname
```

View the web server in action

With IIS installed and port 80 now open on your VM from the Internet, use a web browser of your choice to view the default IIS welcome page. Use the public IP address of your VM obtained in a previous step. The following example shows the default IIS web site:



Clean up resources

When no longer needed, you can use the az group delete command to remove the resource group, VM, and all related resources:

Azure CLI

```
az group delete --name $resourcegroup
```

Next steps

In this quickstart, you deployed a simple virtual machine, open a network port for web traffic, and installed a basic web server. To learn more about Azure virtual machines, continue to the tutorial for Windows VMs.

Azure Windows virtual machine tutorials

Last updated on 02/06/2026

Quickstart: Create a Windows virtual machine in the Azure portal

Applies to: ✔ Windows VMs

Azure virtual machines (VMs) can be created through the Azure portal. This method provides a browser-based user interface to create VMs and their associated resources. This quickstart shows you how to use the Azure portal to deploy a virtual machine (VM) in Azure that runs Windows Server 2022 Datacenter. To see your VM in action, you then RDP to the VM and install the IIS web server.

If you don't have an Azure subscription, create a free account before you begin.

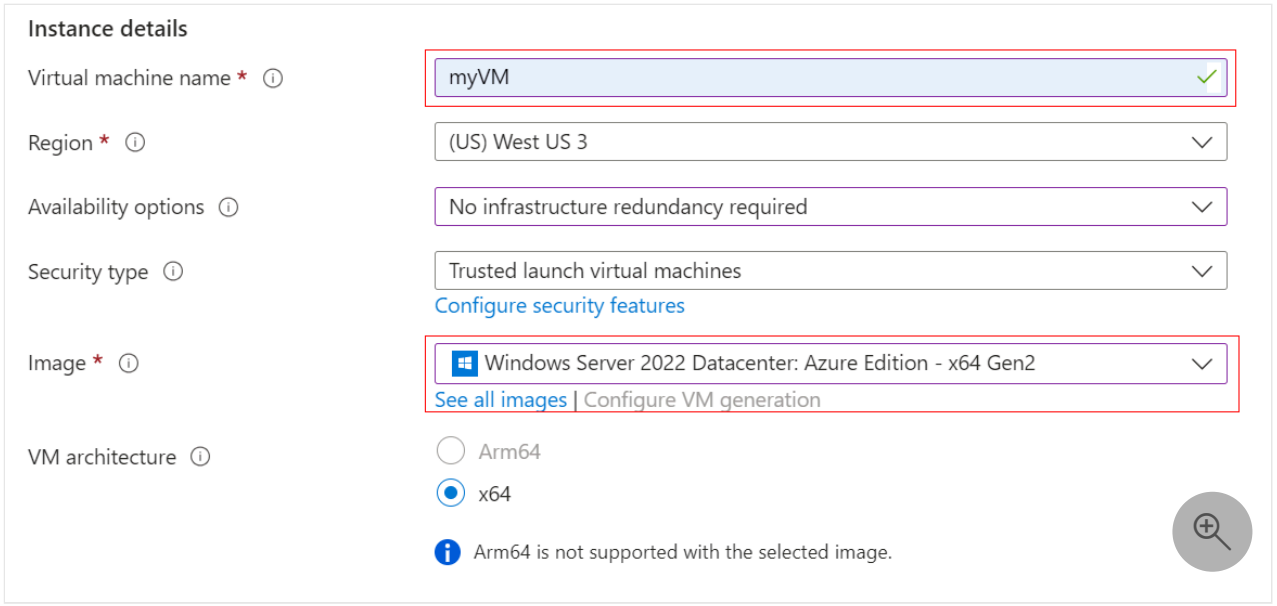
> **Important**
>
> The steps outlined in this quickstart are solely for education purposes and aren't intended for deployments to a production environment.

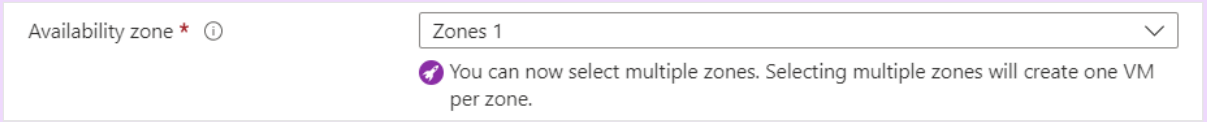
Sign in to Azure

Sign in to the Azure portal.

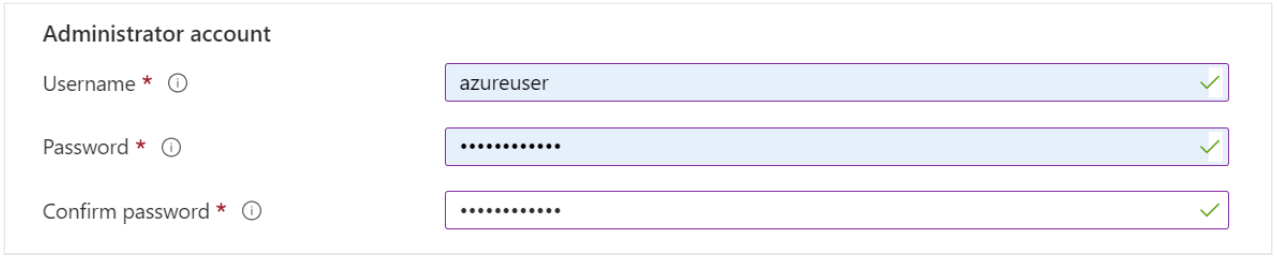
Create virtual machine

1. Enter *virtual machines* in the search.
2. Under **Services**, select **Virtual machines**.
3. In the **Virtual machines** page, select **Create** from the menu at the top of the page and then select **Virtual machine** from the drop-down. The **Create a virtual machine** page opens.
4. Under **Instance details**, enter *myVM* for the **Virtual machine name** and choose *Windows Server 2022 Datacenter: Azure Edition - x64 Gen 2* for the **Image**. The page will fill in the other fields with defaults and use the VM name to create similar names for the other resources that are needed.

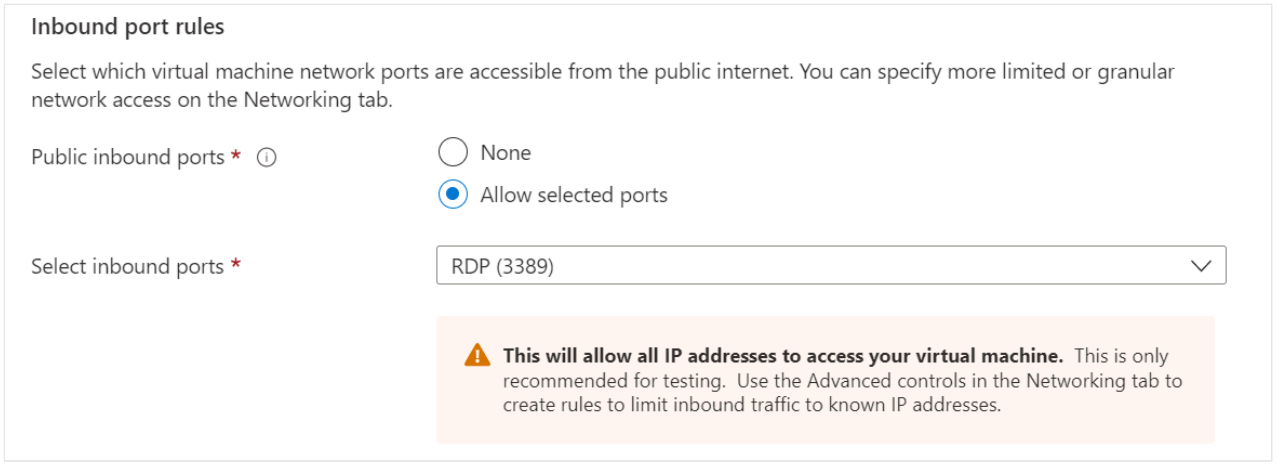


> **Note**
>
> Some users will now see the option to create VMs in multiple zones. To learn more about this new capability, see Create virtual machines in an availability zone.
>
> 
>

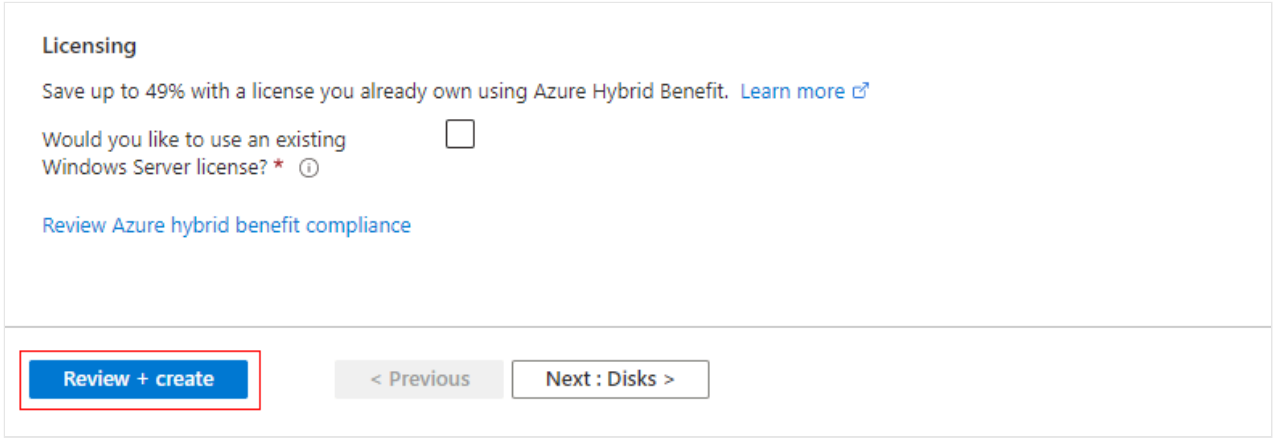
5. Under **Administrator account**, provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the defined complexity requirements.



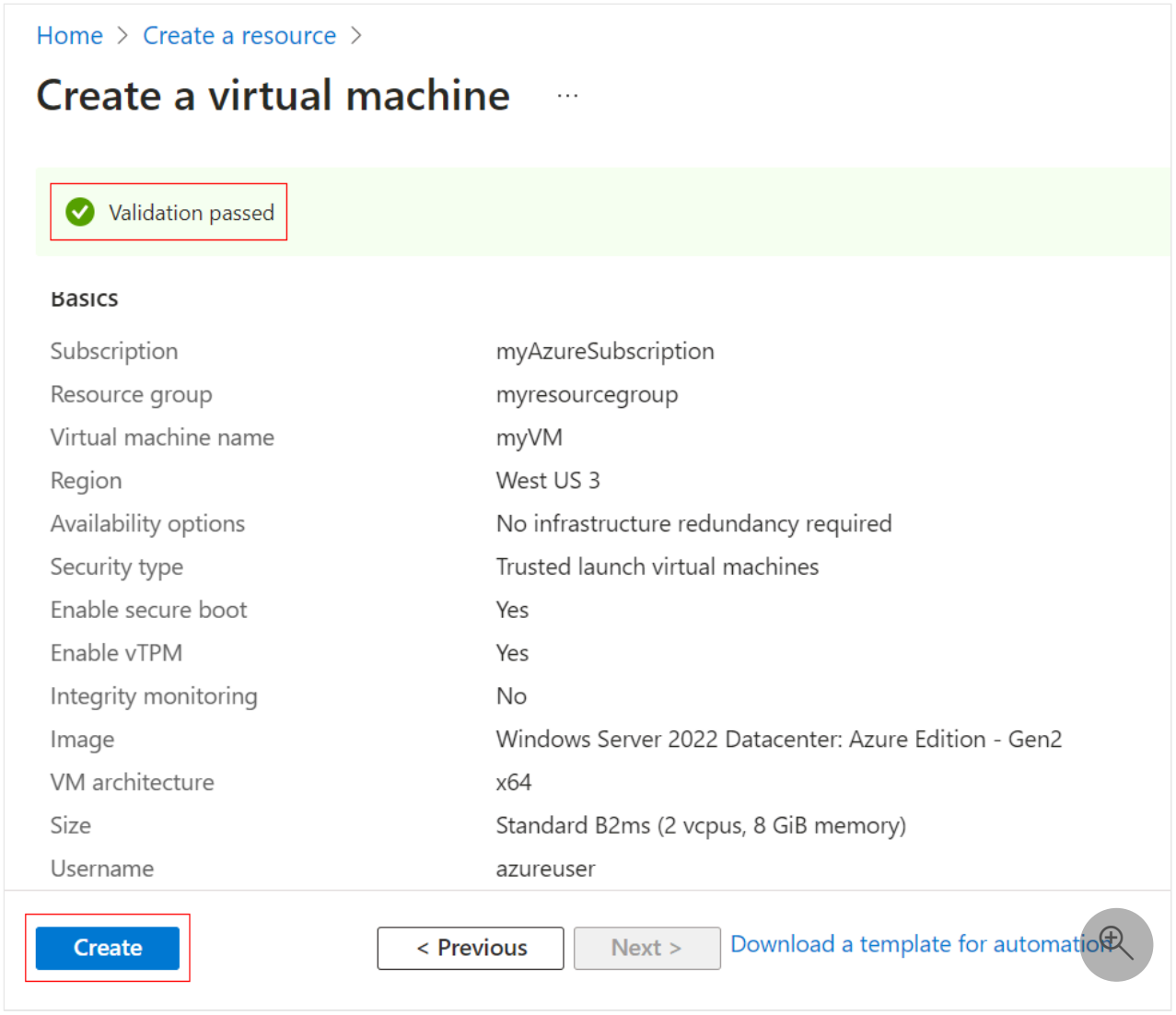
6. Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP (80)** from the drop-down.



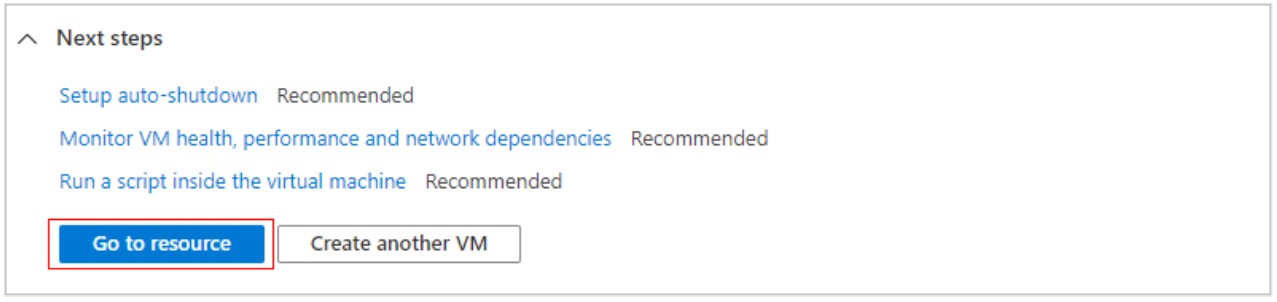
7. At the top of the page, select the **Management** tab.
8. In the **Auto-shutdown** section, select the **Enable auto-shutdown** option.
9. Select a **Shutdown time** and the **Time zone** to use (UTC is the default). Optionally you can choose to be notified by e-mail before the shutdown. For more information see Auto-shutdown.
10. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.



11. After validation runs, select the **Create** button at the bottom of the page.



12. After deployment is complete, select **Go to resource**.



Connect to virtual machine

Create a remote desktop connection to the virtual machine. These directions tell you how to connect to your VM from a Windows computer. On a Mac, you need an RDP client such as this Remote Desktop Client from the Mac App Store.

1. On the overview page for your virtual machine, select **Connect** and then select **Connect** again from the drop-down.

2. In the **Native RDP** page, select the **Check access** button to make sure port 3389 is available.
3. Select **Download RDP file** to download the connection file to your computer.
4. Open the downloaded RDP file and click **Connect** when prompted.
5. You may receive a certificate warning during the sign-in process. Click **Yes** or **Continue** to create the connection.

Install web server

To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

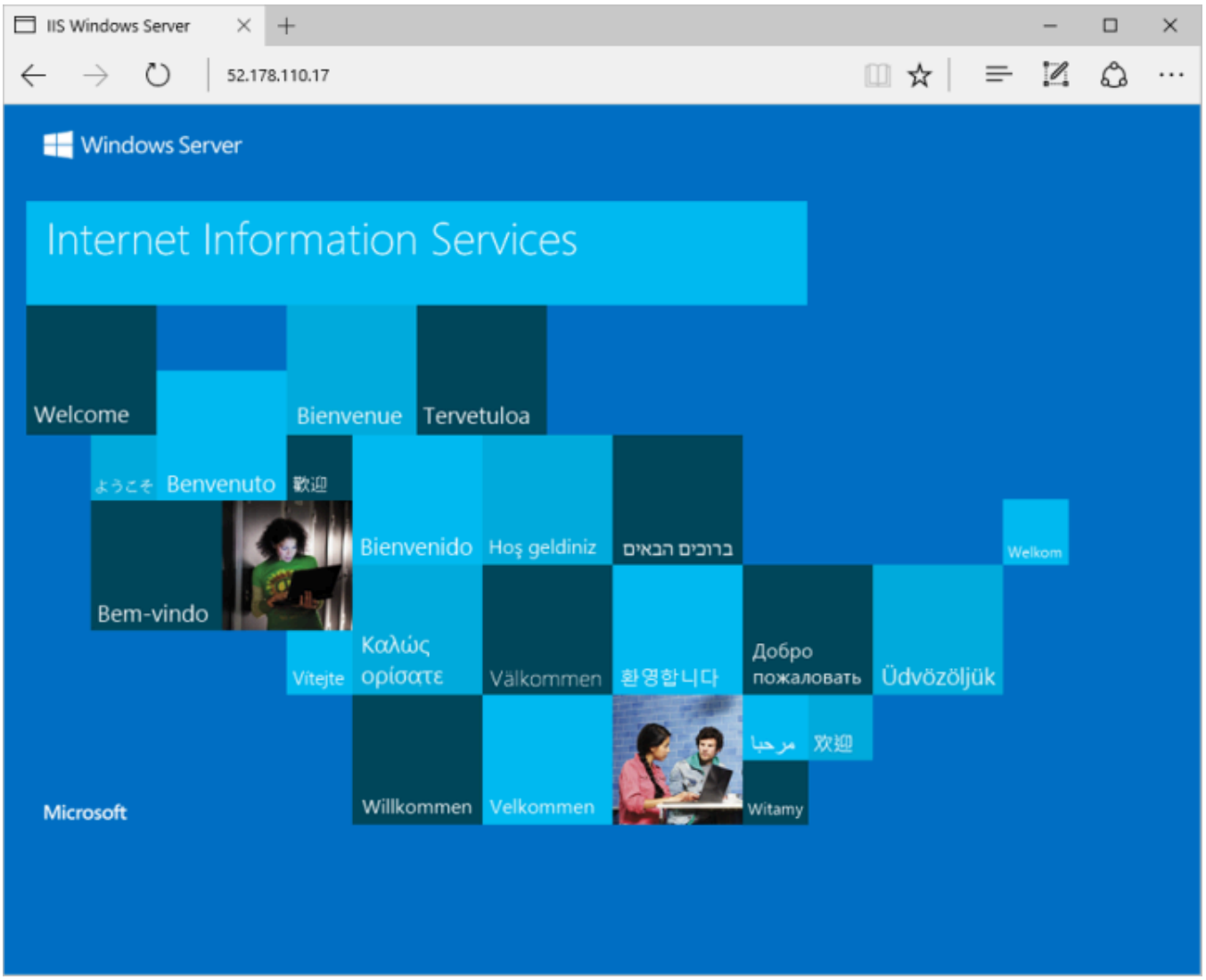
PowerShell

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

When done, close the RDP connection to the VM.

View the IIS welcome page

In the portal, select the VM and in the overview of the VM, hover over the IP address to show **Copy to clipboard**. Copy the IP address and paste it into a browser tab. The default IIS welcome page will open, and should look like this:



Clean up resources

Delete resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources.

1. On the Overview page for the VM, select the **Resource group** link.
2. At the top of the page for the resource group, select **Delete resource group**.
3. A page will open warning you that you are about to delete resources. Type the name of the resource group and select **Delete** to finish deleting the resources and the resource group.

Next steps

In this quickstart, you deployed a simple virtual machine, opened a network port for web traffic, and installed a basic web server. To learn more about Azure virtual machines, continue

to the tutorial for Windows VMs.

Azure Windows virtual machine tutorials

Last updated on 03/27/2026

Quickstart: Create a Windows virtual machine in Azure with PowerShell

Applies to: ✔ Windows VMs

The Azure PowerShell module is used to create and manage Azure resources from the PowerShell command line or in scripts. This quickstart shows you how to use the Azure PowerShell module to deploy a virtual machine (VM) in Azure that runs Windows Server 2016. You also bring Remote Desktop Portal (RDP) to the VM and install the IIS web server, to show the VM in action.

If you don't have an Azure subscription, create a free account before you begin.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Open Cloudshell** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the code blocks, paste them into the Cloud Shell, and press enter to run the them.

Create resource group

Create an Azure resource group with New-AzResourceGroup. A resource group is a logical container into which Azure resources are deployed and managed.

Azure PowerShell

```
New-AzResourceGroup -Name 'myResourceGroup' -Location 'centralus'
```

Create virtual machine

Create a VM with New-AzVM. Provide names for each of the resources and the `New-AzVM` cmdlet creates if they don't already exist.

When prompted, provide a username and password to be used as the sign-in credentials for the VM:

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName 'myResourceGroup' `
    -Name 'myVM' `
    -Location 'centralus' `
    -Image 'MicrosoftWindowsServer:WindowsServer:2022-datacenter-azure-
edition:latest' `
    -VirtualNetworkName 'myVnet' `
    -SubnetName 'mySubnet' `
    -SecurityGroupName 'myNetworkSecurityGroup' `
    -PublicIpAddressName 'myPublicIpAddress' `
    -OpenPorts 80,3389
```

Cost information isn't presented during the virtual machine creation process for PowerShell like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Install web server

To see your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

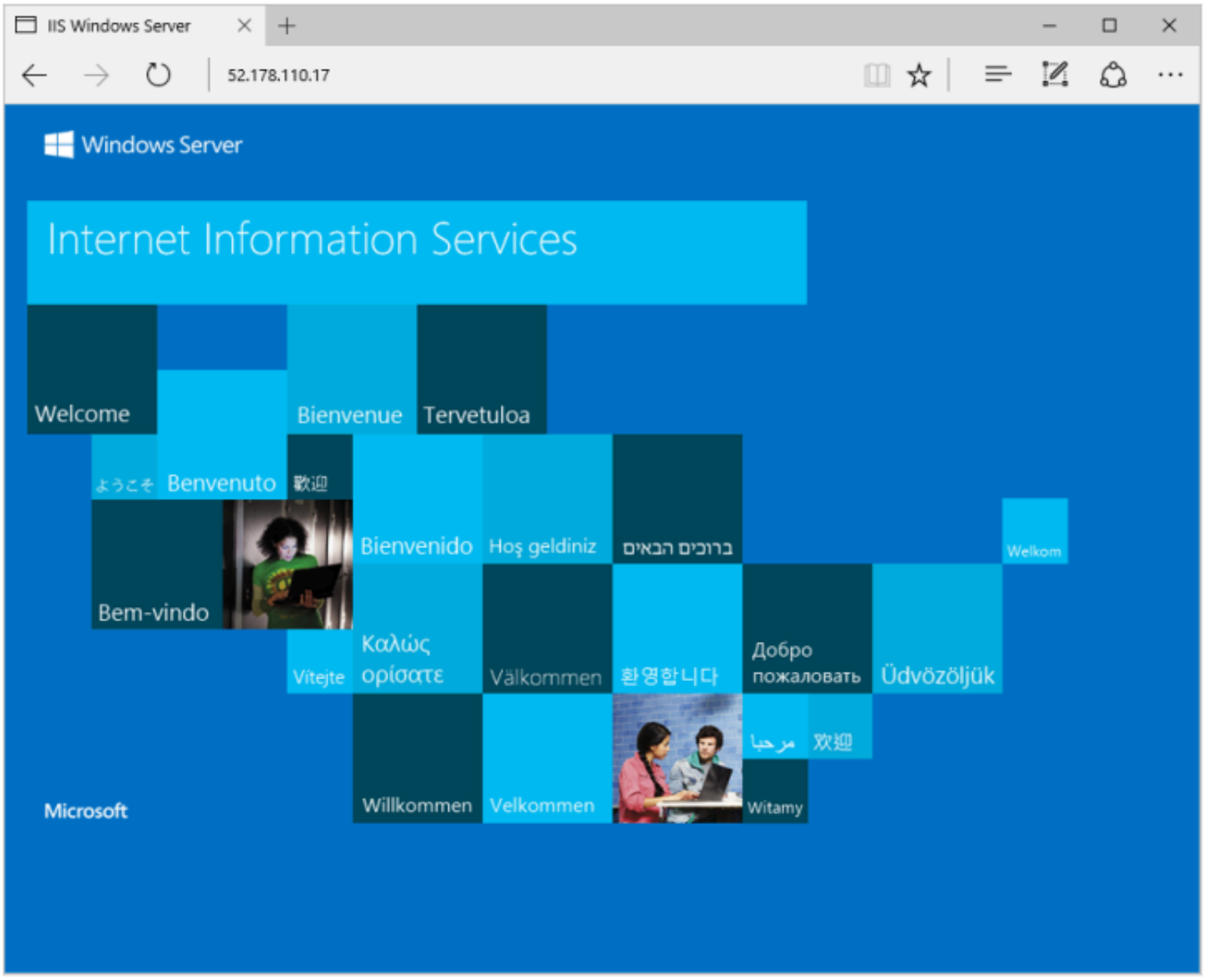
Azure PowerShell

```
Invoke-AzVMRunCommand -ResourceGroupName 'myResourceGroup' -VMName 'myVM' -
CommandId 'RunPowerShellScript' -ScriptString 'Install-WindowsFeature -Name Web-
Server -IncludeManagementTools'
```

The `-ScriptString` parameter requires version `4.27.0` or later of the `Az.Compute` module.

View the web server in action

With IIS installed and port 80 now open on your VM from the Internet, use a web browser of your choice to view the default IIS welcome page. Use the public IP address of the VM that you created. The following example shows the default IIS web site:



Clean up resources

When no longer needed, you can use the Remove-AzResourceGroup cmdlet to remove the resource group, VM, and all related resources:

Azure PowerShell

```
Remove-AzResourceGroup -Name 'myResourceGroup'
```

Next steps

In this quickstart, you deployed a simple virtual machine, opened a network port for web traffic, and installed a basic web server. To learn more about Azure virtual machines, continue to the tutorial for Windows VMs.

Azure Windows virtual machine tutorials

Last updated on 02/02/2026

Quickstart: Use Terraform to create a Windows VM

Article • 08/28/2024

Applies to: ✔ Windows VMs

This article shows you how to create a complete Windows environment and supporting resources with Terraform. Those resources include a virtual network, subnet, public IP address, and more.

Terraform enables the definition, preview, and deployment of cloud infrastructure. Using Terraform, you create configuration files using HCL syntax. The HCL syntax allows you to specify the cloud provider - such as Azure - and the elements that make up your cloud infrastructure. After you create your configuration files, you create an *execution plan* that allows you to preview your infrastructure changes before they're deployed. Once you verify the changes, you apply the execution plan to deploy the infrastructure.

In this article, you learn how to:

- ✔ Create a random value for the Azure resource group name using random_pet.
- ✔ Create an Azure resource group using azurerm_resource_group.
- ✔ Create a virtual network (VNET) using azurerm_virtual_network.
- ✔ Create a subnet using azurerm_subnet.
- ✔ Create a public IP using azurerm_public_ip.
- ✔ Create a network security group using azurerm_network_security_group.
- ✔ Create a network interface using azurerm_network_interface.
- ✔ Create an association between the network security group and the network interface using azurerm_network_interface_security_group_association.
- ✔ Generate a random value for a unique storage account name using random_id.
- ✔ Create a storage account for boot diagnostics using azurerm_storage_account.
- ✔ Create a Windows VM with an IIS web server using azurerm_windows_virtual_machine.
- ✔ Create a Windows VM extension using azurerm_virtual_machine_extension.

Prerequisites

- **Azure subscription**: If you don't have an Azure subscription, create a free account before you begin.

- Install and configure Terraform

Implement the Terraform code

> **Note**
>
> The sample code for this article is located in the Azure Terraform GitHub repo. You can view the log file containing the test results from current and previous versions of Terraform.
>
> See more articles and sample code showing how to use Terraform to manage Azure resources

1. Create a directory in which to test the sample Terraform code and make it the current directory.

2. Create a file named `providers.tf` and insert the following code:

   Terraform

   ```
   terraform {
     required_version = ">=1.0"

     required_providers {
       azurerm = {
         source  = "hashicorp/azurerm"
         version = "~>3.0"
       }
       random = {
         source  = "hashicorp/random"
         version = "~>3.0"
       }
     }
   }

   provider "azurerm" {
     features {}
   }
   ```

3. Create a file named `main.tf` and insert the following code:

   Terraform

   ```
   resource "azurerm_resource_group" "rg" {
     location = var.resource_group_location
     name     = "${random_pet.prefix.id}-rg"
   ```

```
}

# Create virtual network
resource "azurerm_virtual_network" "my_terraform_network" {
  name                = "${random_pet.prefix.id}-vnet"
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
}

# Create subnet
resource "azurerm_subnet" "my_terraform_subnet" {
  name                 = "${random_pet.prefix.id}-subnet"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name =
azurerm_virtual_network.my_terraform_network.name
  address_prefixes     = ["10.0.1.0/24"]
}

# Create public IPs
resource "azurerm_public_ip" "my_terraform_public_ip" {
  name                = "${random_pet.prefix.id}-public-ip"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  allocation_method   = "Dynamic"
}

# Create Network Security Group and rules
resource "azurerm_network_security_group" "my_terraform_nsg" {
  name                = "${random_pet.prefix.id}-nsg"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  security_rule {
    name                       = "RDP"
    priority                   = 1000
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "*"
    source_port_range          = "*"
    destination_port_range     = "3389"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }
  security_rule {
    name                       = "web"
    priority                   = 1001
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "80"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }
```

```
}

# Create network interface
resource "azurerm_network_interface" "my_terraform_nic" {
  name                = "${random_pet.prefix.id}-nic"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                          = "my_nic_configuration"
    subnet_id                     =
azurerm_subnet.my_terraform_subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          =
azurerm_public_ip.my_terraform_public_ip.id
  }
}

# Connect the security group to the network interface
resource "azurerm_network_interface_security_group_association"
"example" {
  network_interface_id      =
azurerm_network_interface.my_terraform_nic.id
  network_security_group_id =
azurerm_network_security_group.my_terraform_nsg.id
}

# Create storage account for boot diagnostics
resource "azurerm_storage_account" "my_storage_account" {
  name                     = "diag${random_id.random_id.hex}"
  location                 = azurerm_resource_group.rg.location
  resource_group_name      = azurerm_resource_group.rg.name
  account_tier             = "Standard"
  account_replication_type = "LRS"
}


# Create virtual machine
resource "azurerm_windows_virtual_machine" "main" {
  name                  = "${var.prefix}-vm"
  admin_username        = "azureuser"
  admin_password        = random_password.password.result
  location              = azurerm_resource_group.rg.location
  resource_group_name   = azurerm_resource_group.rg.name
  network_interface_ids =
[azurerm_network_interface.my_terraform_nic.id]
  size                  = "Standard_DS1_v2"

  os_disk {
    name                 = "myOsDisk"
    caching              = "ReadWrite"
    storage_account_type = "Premium_LRS"
  }

  source_image_reference {
```

```
    publisher = "MicrosoftWindowsServer"
    offer     = "WindowsServer"
    sku       = "2022-datacenter-azure-edition"
    version   = "latest"
  }


  boot_diagnostics {
    storage_account_uri =
azurerm_storage_account.my_storage_account.primary_blob_endpoint
  }
}

# Install IIS web server to the virtual machine
resource "azurerm_virtual_machine_extension" "web_server_install" {
  name                       = "${random_pet.prefix.id}-wsi"
  virtual_machine_id         = azurerm_windows_virtual_machine.main.id
  publisher                  = "Microsoft.Compute"
  type                       = "CustomScriptExtension"
  type_handler_version       = "1.8"
  auto_upgrade_minor_version = true

  settings = <<SETTINGS
    {
      "commandToExecute": "powershell -ExecutionPolicy Unrestricted
Install-WindowsFeature -Name Web-Server -IncludeAllSubFeature -
IncludeManagementTools"
    }
  SETTINGS
}

# Generate random text for a unique storage account name
resource "random_id" "random_id" {
  keepers = {
    # Generate a new ID only when a new resource group is defined
    resource_group = azurerm_resource_group.rg.name
  }

  byte_length = 8
}

resource "random_password" "password" {
  length      = 20
  min_lower   = 1
  min_upper   = 1
  min_numeric = 1
  min_special = 1
  special     = true
}

resource "random_pet" "prefix" {
  prefix = var.prefix
  length = 1
}
```

4. Create a file named `variables.tf` and insert the following code:

```terraform
variable "resource_group_location" {
  default     = "eastus"
  description = "Location of the resource group."
}

variable "prefix" {
  type        = string
  default     = "win-vm-iis"
  description = "Prefix of the resource name"
}
```

5. Create a file named `outputs.tf` and insert the following code:

```terraform
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}

output "public_ip_address" {
  value = azurerm_windows_virtual_machine.main.public_ip_address
}

output "admin_password" {
  sensitive = true
  value     = azurerm_windows_virtual_machine.main.admin_password
}
```

Initialize Terraform

Run terraform init to initialize the Terraform deployment. This command downloads the Azure provider required to manage your Azure resources.

```console
terraform init -upgrade
```

Key points:

- The `-upgrade` parameter upgrades the necessary provider plugins to the newest version that complies with the configuration's version constraints.

Create a Terraform execution plan

Run terraform plan to create an execution plan.

```
Console

terraform plan -out main.tfplan
```

Key points:

- The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

Apply a Terraform execution plan

Run terraform apply to apply the execution plan to your cloud infrastructure.

```
Console

terraform apply main.tfplan
```

Key points:

- The example `terraform apply` command assumes you previously ran `terraform plan -out main.tfplan`.
- If you specified a different filename for the `-out` parameter, use that same filename in the call to `terraform apply`.
- If you didn't use the `-out` parameter, call `terraform apply` without any parameters.

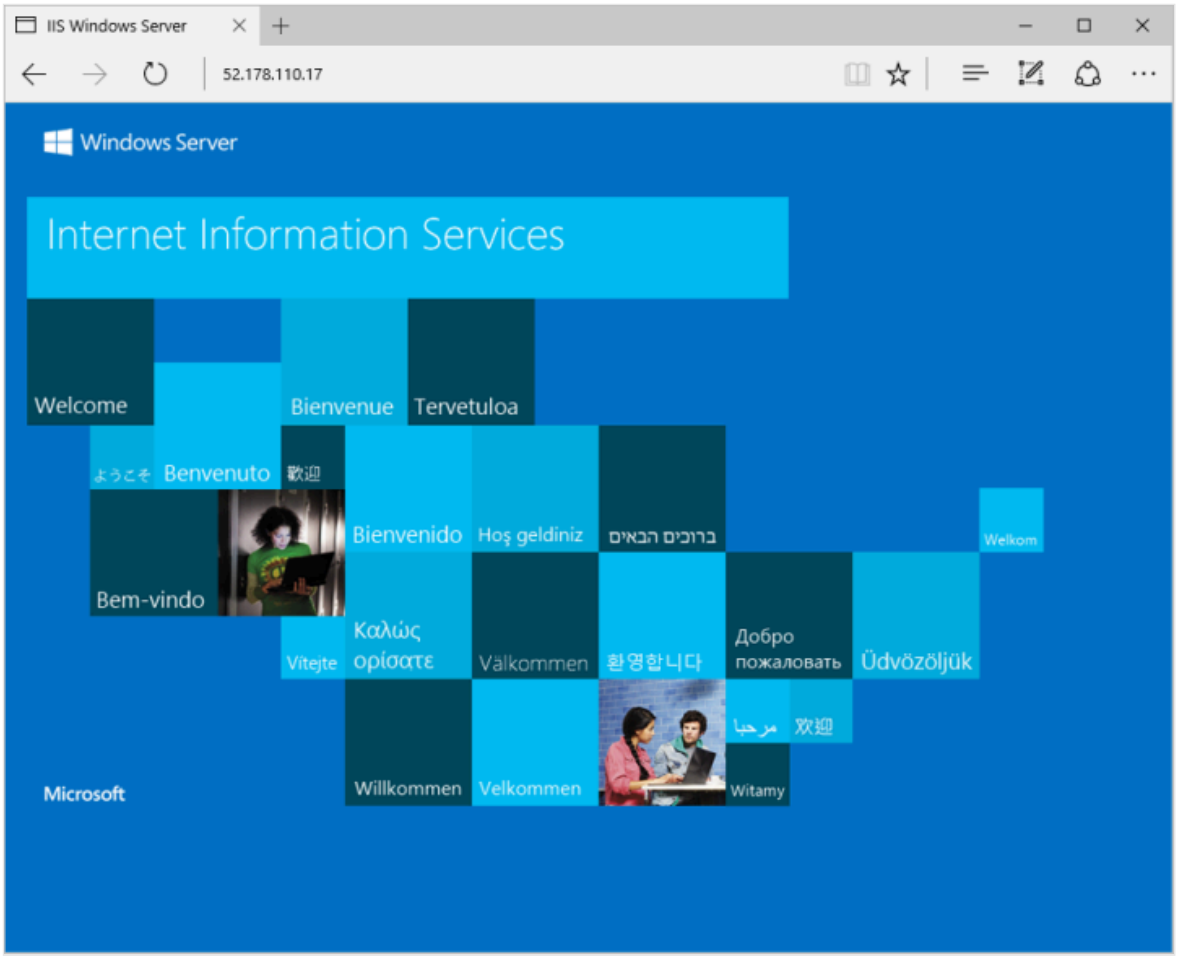
Cost information isn't presented during the virtual machine creation process for Terraform like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Verify the results

1. Run the following command to get the VM's public IP address and make note of it:

```
Azure CLI

echo $(terraform output -raw public_ip_address)
```

2. With IIS installed and port 80 now open on your VM from the Internet, use a web browser of your choice to view the default IIS welcome page. Use the public IP address of your VM obtained from the previous command. The following example shows the default IIS web site:



Clean up resources

When you no longer need the resources created via Terraform, do the following steps:

1. Run terraform plan and specify the `destroy` flag.

```
Console

terraform plan -destroy -out main.destroy.tfplan
```

Key points:

- The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

2. Run terraform apply to apply the execution plan.

```
Console

terraform apply main.destroy.tfplan
```

Troubleshoot Terraform on Azure

Troubleshoot common problems when using Terraform on Azure

Next steps

In this quickstart, you deployed a simple virtual machine using Terraform. To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Azure Windows virtual machine tutorials

Note: The author created this article with assistance from AI. Learn more

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a Windows VM cluster in Azure using Terraform

Article • 08/27/2024

Applies to: ✔ Windows VMs

This article shows you how to create a Windows VM cluster (containing three Windows VM instances) in Azure using Terraform.

- ✔ Create a random value for the Azure resource group name using random_pet.
- ✔ Create an Azure resource group using azurerm_resource_group.
- ✔ Create a random value for the Windows VM host name random_string.
- ✔ Create a random password for the Windows VMs using random_password.
- ✔ Create a Windows VM using the compute module.
- ✔ Create a virtual network along with subnet using the network module.

Prerequisites

- Install and configure Terraform

Implement the Terraform code

> **Note**
>
> The sample code for this article is located in the Azure Terraform GitHub repo. You can view the log file containing the test results from current and previous versions of Terraform.
>
> See more articles and sample code showing how to use Terraform to manage Azure resources

1. Create a directory in which to test the sample Terraform code and make it the current directory.

2. Create a file named `providers.tf` and insert the following code:

```
Terraform
```

```
terraform {
  required_version = ">=1.0"
```

```
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~>3.0"
    }
    random = {
      source  = "hashicorp/random"
      version = "~>3.0"
    }
  }
}
provider "azurerm" {
  features {}
}
```

3. Create a file named `main.tf` and insert the following code:

Terraform

```
resource "random_pet" "rg_name" {
  prefix = var.resource_group_name_prefix
}

resource "azurerm_resource_group" "rg" {
  name     = random_pet.rg_name.id
  location = var.resource_group_location
}

resource "random_string" "windows_server_vm_hostname" {
  length  = 8
  lower   = true
  upper   = false
  special = false
}

resource "random_pet" "windows_server_public_ip_dns" {
  prefix = "dns"
}

resource "random_password" "password" {
  length  = 16
  special = true
  lower   = true
  upper   = true
  numeric = true
}

# The following module is a Terraform Verified Module.
# For more information about Verified Modules, see
# https://github.com/azure/terraform-azure-modules/
module "windows_server" {
  count                        = 3 # Define 3 Windows Server VMs
  source                       = "Azure/compute/azurerm"
```

```
  resource_group_name           = azurerm_resource_group.rg.name
  vnet_subnet_id                = module.network.vnet_subnets[0]
  is_windows_image              = true
  vm_hostname                   =
"vm-${random_string.windows_server_vm_hostname.result}-${count.index}"
  delete_os_disk_on_termination = true
  admin_password                = random_password.password.result
  vm_os_simple                  = "WindowsServer"
  public_ip_dns                 =
["${random_pet.windows_server_public_ip_dns.id}-${count.index}"]
}

# The following module is a Terraform Verified Module.
# For more information about Verified Modules, see
# https://github.com/azure/terraform-azure-modules/
module "network" {
  source              = "Azure/network/azurerm"
  resource_group_name = azurerm_resource_group.rg.name
  version             = "5.2.0"
  subnet_prefixes     = ["10.0.1.0/24"]
  subnet_names        = ["subnet1"]
  use_for_each        = true
}
```

4. Create a file named `variables.tf` and insert the following code:

Terraform

```
variable "resource_group_location" {
  type        = string
  default     = "eastus"
  description = "Location for all resources."
}

variable "resource_group_name_prefix" {
  type        = string
  default     = "rg"
  description = "Prefix of the resource group name that's combined with
a random value so name is unique in your Azure subscription."
}
```

5. Create a file named `outputs.tf` and insert the following code:

Terraform

```
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}

output "windows_vm_public_names" {
  value = module.windows_server[*].public_ip_dns_name
```

```
}

output "vm_public_ip_addresses" {
  value = module.windows_server[*].public_ip_address
}

output "vm_private_ip_addresses" {
  value = module.windows_server[*].network_interface_private_ip
}

output "vm_hostnames" {
  value = module.windows_server[*].vm_names
}
```

Initialize Terraform

Run terraform init to initialize the Terraform deployment. This command downloads the Azure provider required to manage your Azure resources.

Console

```
terraform init -upgrade
```

Key points:

- The `-upgrade` parameter upgrades the necessary provider plugins to the newest version that complies with the configuration's version constraints.

Create a Terraform execution plan

Run terraform plan to create an execution plan.

Console

```
terraform plan -out main.tfplan
```

Key points:

- The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.

- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

Apply a Terraform execution plan

Run terraform apply to apply the execution plan to your cloud infrastructure.

Console

```
terraform apply main.tfplan
```

Key points:

- The example `terraform apply` command assumes you previously ran `terraform plan -out main.tfplan`.
- If you specified a different filename for the `-out` parameter, use that same filename in the call to `terraform apply`.
- If you didn't use the `-out` parameter, call `terraform apply` without any parameters.

Cost information isn't presented during the virtual machine creation process for Terraform like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Verify the results

Azure CLI

1. Get the Azure resource group name.

   Console

   ```
   resource_group_name=$(terraform output -raw resource_group_name)
   ```

2. Run az vm list with a JMESPath query to display the names of the virtual machines created in the resource group.

   Azure CLI

   ```
   az vm list \
     --resource-group $resource_group_name \
   ```

```
    --query "[].{\"VM Name\":name}" -o table
```

Clean up resources

When you no longer need the resources created via Terraform, do the following steps:

1. Run terraform plan and specify the `destroy` flag.

   Console

   ```
   terraform plan -destroy -out main.destroy.tfplan
   ```

   Key points:

   - The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
   - The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

2. Run terraform apply to apply the execution plan.

   Console

   ```
   terraform apply main.destroy.tfplan
   ```

Troubleshoot Terraform on Azure

Troubleshoot common problems when using Terraform on Azure

Next steps

Azure Windows virtual machine tutorials

ⓘ **Note:** The author created this article with assistance from AI. Learn more

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a Windows virtual machine using a Bicep file

Applies to: ✔ Windows VMs

This quickstart shows you how to use a Bicep file to deploy a Windows virtual machine (VM) in Azure. Availability of VM sizes, images, and features can vary by Azure region and subscription SKU.

Bicep is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. It provides concise syntax, reliable type safety, and support for code reuse. Bicep offers the best authoring experience for your infrastructure-as-code solutions in Azure.

Prerequisites

- If you don't have an Azure subscription, create a free account before you begin.
- Install Bicep tools
- Install Azure CLI (version 2.20.0 or later) or Azure PowerShell (Az module 5.6.0 or later).
- Signed in to Azure with the correct subscription selected. Use Azure CLI's az login command, or Azure PowerShell's Connect-AzAccount.

Review the Bicep file

The Bicep file used in this quickstart is from Azure Quickstart Templates. The template relies on the following parameters:

Expand table

Parameter	Default value	Allowed values	Constraints
`adminUsername`	*No default (required)*	Any valid Windows admin username	Required. Cannot use reserved usernames such as `admin` or `administrator`.
`vmSize`	`Standard_DS1_v2`	Region-dependent Azure VM sizes	Size availability depends on the selected region and subscription quota.
`location`	Resource group location	Any Azure region	Some VM sizes and images are not available in all regions.
`windowsOSVersion`	`2019-Datacenter`	Supported Windows Server images in Azure	Image availability varies by region.

Parameter	Default value	Allowed values	Constraints
		Marketplace	

Bicep

```
@description('Username for the Virtual Machine.')
param adminUsername string

@description('Password for the Virtual Machine.')
@minLength(12)
@secure()
param adminPassword string

@description('Unique DNS Name for the Public IP used to access the Virtual
Machine.')
param dnsLabelPrefix string = toLower('${vmName}-${uniqueString(resourceGroup().id,
vmName)}')

@description('Name for the Public IP used to access the Virtual Machine.')
param publicIpName string = 'myPublicIP'

@description('Allocation method for the Public IP used to access the Virtual
Machine.')
param publicIPAllocationMethod string = 'Static'

@description('SKU for the Public IP used to access the Virtual Machine.')
param publicIpSku string = 'Standard'

@description('The Windows version for the VM. This will pick a fully patched image
of this given Windows version.')
@allowed([
  '2016-datacenter-gensecond'
  '2016-datacenter-server-core-g2'
  '2016-datacenter-server-core-smalldisk-g2'
  '2016-datacenter-smalldisk-g2'
  '2016-datacenter-with-containers-g2'
  '2016-datacenter-zhcn-g2'
  '2019-datacenter-core-g2'
  '2019-datacenter-core-smalldisk-g2'
  '2019-datacenter-core-with-containers-g2'
  '2019-datacenter-core-with-containers-smalldisk-g2'
  '2019-datacenter-gensecond'
  '2019-datacenter-smalldisk-g2'
  '2019-datacenter-with-containers-g2'
  '2019-datacenter-with-containers-smalldisk-g2'
  '2019-datacenter-zhcn-g2'
  '2022-datacenter-azure-edition'
  '2022-datacenter-azure-edition-core'
  '2022-datacenter-azure-edition-core-smalldisk'
  '2022-datacenter-azure-edition-smalldisk'
  '2022-datacenter-core-g2'
  '2022-datacenter-core-smalldisk-g2'
  '2022-datacenter-g2'
```

```
  '2022-datacenter-smalldisk-g2'
])
param OSVersion string = '2022-datacenter-azure-edition'

@description('Size of the virtual machine.')
param vmSize string = 'Standard_D2s_v5'

@description('Location for all resources.')
param location string = resourceGroup().location

@description('Name of the virtual machine.')
param vmName string = 'simple-vm'

@description('Security Type of the Virtual Machine.')
@allowed([
  'Standard'
  'TrustedLaunch'
])
param securityType string = 'TrustedLaunch'

var storageAccountName = 'bootdiags${uniqueString(resourceGroup().id)}'
var nicName = 'myVMNic'
var addressPrefix = '10.0.0.0/16'
var subnetName = 'Subnet'
var subnetPrefix = '10.0.0.0/24'
var virtualNetworkName = 'MyVNET'
var networkSecurityGroupName = 'default-NSG'
var securityProfileJson = {
  uefiSettings: {
    secureBootEnabled: true
    vTpmEnabled: true
  }
  securityType: securityType
}
var extensionName = 'GuestAttestation'
var extensionPublisher = 'Microsoft.Azure.Security.WindowsAttestation'
var extensionVersion = '1.0'
var maaTenantName = 'GuestAttestation'
var maaEndpoint = substring('emptyString', 0, 0)

resource storageAccount 'Microsoft.Storage/storageAccounts@2022-05-01' = {
  name: storageAccountName
  location: location
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'Storage'
}

resource publicIp 'Microsoft.Network/publicIPAddresses@2022-05-01' = {
  name: publicIpName
  location: location
  sku: {
    name: publicIpSku
  }
```

```
  properties: {
    publicIPAllocationMethod: publicIPAllocationMethod
    dnsSettings: {
      domainNameLabel: dnsLabelPrefix
    }
  }
}

resource networkSecurityGroup 'Microsoft.Network/networkSecurityGroups@2022-05-01' =
{
  name: networkSecurityGroupName
  location: location
  properties: {
    securityRules: [
      {
        name: 'default-allow-3389'
        properties: {
          priority: 1000
          access: 'Allow'
          direction: 'Inbound'
          destinationPortRange: '3389'
          protocol: 'Tcp'
          sourcePortRange: '*'
          sourceAddressPrefix: '*'
          destinationAddressPrefix: '*'
        }
      }
    ]
  }
}

resource virtualNetwork 'Microsoft.Network/virtualNetworks@2022-05-01' = {
  name: virtualNetworkName
  location: location
  properties: {
    addressSpace: {
      addressPrefixes: [
        addressPrefix
      ]
    }
    subnets: [
      {
        name: subnetName
        properties: {
          addressPrefix: subnetPrefix
          networkSecurityGroup: {
            id: networkSecurityGroup.id
          }
        }
      }
    ]
  }
}

resource nic 'Microsoft.Network/networkInterfaces@2022-05-01' = {
```

```
  name: nicName
  location: location
  properties: {
    ipConfigurations: [
      {
        name: 'ipconfig1'
        properties: {
          privateIPAllocationMethod: 'Dynamic'
          publicIPAddress: {
            id: publicIp.id
          }
          subnet: {
            id: resourceId('Microsoft.Network/virtualNetworks/subnets',
virtualNetworkName, subnetName)
          }
        }
      }
    ]
  }
  dependsOn: [

    virtualNetwork
  ]
}

resource vm 'Microsoft.Compute/virtualMachines@2022-03-01' = {
  name: vmName
  location: location
  properties: {
    hardwareProfile: {
      vmSize: vmSize
    }
    osProfile: {
      computerName: vmName
      adminUsername: adminUsername
      adminPassword: adminPassword
    }
    storageProfile: {
      imageReference: {
        publisher: 'MicrosoftWindowsServer'
        offer: 'WindowsServer'
        sku: OSVersion
        version: 'latest'
      }
      osDisk: {
        createOption: 'FromImage'
        managedDisk: {
          storageAccountType: 'StandardSSD_LRS'
        }
      }
      dataDisks: [
        {
          diskSizeGB: 1023
          lun: 0
          createOption: 'Empty'
```

```
        }
      ]
    }
    networkProfile: {
      networkInterfaces: [
        {
          id: nic.id
        }
      ]
    }
    diagnosticsProfile: {
      bootDiagnostics: {
        enabled: true
        storageUri: storageAccount.properties.primaryEndpoints.blob
      }
    }
    securityProfile: ((securityType == 'TrustedLaunch') ? securityProfileJson :
null)
  }
}

resource vmExtension 'Microsoft.Compute/virtualMachines/extensions@2022-03-01' = if
((securityType == 'TrustedLaunch') &&
((securityProfileJson.uefiSettings.secureBootEnabled == true) &&
(securityProfileJson.uefiSettings.vTpmEnabled == true))) {
  parent: vm
  name: extensionName
  location: location
  properties: {
    publisher: extensionPublisher
    type: extensionName
    typeHandlerVersion: extensionVersion
    autoUpgradeMinorVersion: true
    enableAutomaticUpgrade: true
    settings: {
      AttestationConfig: {
        MaaSettings: {
          maaEndpoint: maaEndpoint
          maaTenantName: maaTenantName
        }
      }
    }
  }
}

output hostname string = publicIp.properties.dnsSettings.fqdn
```

Several resources are defined in the Bicep file:

- Microsoft.Network/virtualNetworks/subnets: create a subnet.
- Microsoft.Storage/storageAccounts: create a storage account.
- Microsoft.Network/publicIPAddresses: create a public IP address.
- Microsoft.Network/networkSecurityGroups: create a network security group.

- Microsoft.Network/virtualNetworks: create a virtual network.
- Microsoft.Network/networkInterfaces: create a NIC.
- Microsoft.Compute/virtualMachines: create a virtual machine.

Deploy the Bicep file

1. Save the Bicep file as **main.bicep** to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell.

CLI

Azure CLI

```
exampleRG=myResourceGroupName
adminUserName=myAdminUserName

az group create --name exampleRG --location eastus
az deployment group create --resource-group exampleRG --template-file
main.bicep --parameters adminUserName=<admin-username>
```

> **Note**
>
> Replace <**adminUserName**> with a unique username. You'll also be prompted to enter adminPassword. The minimum password length is 12 characters.

When the deployment finishes, you should see a messaged indicating the deployment succeeded. Use Azure CLI to verify that the virtual machine was created successfully by checking that the deployment provisioning state is `Succeeded`.

Azure CLI

```
 az deployment group show --resource-group exampleRG --name main --query
properties.provisioningState
```

Cost information isn't presented during the virtual machine creation process for Bicep like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

Review deployed resources

Use the Azure portal, Azure CLI, or Azure PowerShell to list the deployed resources in the resource group.

CLI

Azure CLI

```
az resource list --resource-group exampleRG
```

Clean up resources

When no longer needed, use the Azure portal, Azure CLI, or Azure PowerShell to delete the VM and all of the resources in the resource group.

CLI

Azure CLI

```
az group delete --name exampleRG
```

Next steps

In this quickstart, you deployed a simple virtual machine using a Bicep file. To learn more about Azure virtual machines, continue to the tutorial for Linux VMs.

Azure Windows virtual machine tutorials

Last updated on 01/22/2026

Quickstart: Create a Windows virtual machine using a template

Article • 08/22/2024

Applies to: ✔ Windows VMs

This quickstart shows you how to use an Azure Resource Manager template to deploy a Windows virtual machine (VM) in Azure.

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using templates, select the **Deploy to Azure** button. The template opens in the Azure portal.



Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

Review the template

The template used in this quickstart is from Azure Quickstart Templates.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "metadata": {
    "_generator": {
      "name": "bicep",
      "version": "0.16.2.56959",
      "templateHash": "14427937023370378081"
    }
  },
  "parameters": {
    "adminUsername": {
      "type": "string",
```

```
      "metadata": {
        "description": "Username for the Virtual Machine."
      }
    },
    "adminPassword": {
      "type": "securestring",
      "minLength": 12,
      "metadata": {
        "description": "Password for the Virtual Machine."
      }
    },
    "dnsLabelPrefix": {
      "type": "string",
      "defaultValue": "[toLower(format('{0}-{1}', parameters('vmName'),
uniqueString(resourceGroup().id, parameters('vmName'))))]",
      "metadata": {
        "description": "Unique DNS Name for the Public IP used to access the
Virtual Machine."
      }
    },
    "publicIpName": {
      "type": "string",
      "defaultValue": "myPublicIP",
      "metadata": {
        "description": "Name for the Public IP used to access the Virtual
Machine."
      }
    },
    "publicIPAllocationMethod": {
      "type": "string",
      "defaultValue": "Dynamic",
      "allowedValues": [
        "Dynamic",
        "Static"
      ],
      "metadata": {
        "description": "Allocation method for the Public IP used to access
the Virtual Machine."
      }
    },
    "publicIpSku": {
      "type": "string",
      "defaultValue": "Basic",
      "allowedValues": [
        "Basic",
        "Standard"
      ],
      "metadata": {
        "description": "SKU for the Public IP used to access the Virtual
Machine."
      }
    },
    "OSVersion": {
      "type": "string",
      "defaultValue": "2022-datacenter-azure-edition",
```

```
      "allowedValues": [
        "2016-datacenter-gensecond",
        "2016-datacenter-server-core-g2",
        "2016-datacenter-server-core-smalldisk-g2",
        "2016-datacenter-smalldisk-g2",
        "2016-datacenter-with-containers-g2",
        "2016-datacenter-zhcn-g2",
        "2019-datacenter-core-g2",
        "2019-datacenter-core-smalldisk-g2",
        "2019-datacenter-core-with-containers-g2",
        "2019-datacenter-core-with-containers-smalldisk-g2",
        "2019-datacenter-gensecond",
        "2019-datacenter-smalldisk-g2",
        "2019-datacenter-with-containers-g2",
        "2019-datacenter-with-containers-smalldisk-g2",
        "2019-datacenter-zhcn-g2",
        "2022-datacenter-azure-edition",
        "2022-datacenter-azure-edition-core",
        "2022-datacenter-azure-edition-core-smalldisk",
        "2022-datacenter-azure-edition-smalldisk",
        "2022-datacenter-core-g2",
        "2022-datacenter-core-smalldisk-g2",
        "2022-datacenter-g2",
        "2022-datacenter-smalldisk-g2"
      ],
      "metadata": {
        "description": "The Windows version for the VM. This will pick a
fully patched image of this given Windows version."
      }
    },
    "vmSize": {
      "type": "string",
      "defaultValue": "Standard_D2s_v5",
      "metadata": {
        "description": "Size of the virtual machine."
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Location for all resources."
      }
    },
    "vmName": {
      "type": "string",
      "defaultValue": "simple-vm",
      "metadata": {
        "description": "Name of the virtual machine."
      }
    },
    "securityType": {
      "type": "string",
      "defaultValue": "TrustedLaunch",
      "allowedValues": [
```

```
        "Standard",
        "TrustedLaunch"
      ],
      "metadata": {
        "description": "Security Type of the Virtual Machine."
      }
    }
  },
  "variables": {
    "storageAccountName": "[format('bootdiags{0}',
uniqueString(resourceGroup().id))]",
    "nicName": "myVMNic",
    "addressPrefix": "10.0.0.0/16",
    "subnetName": "Subnet",
    "subnetPrefix": "10.0.0.0/24",
    "virtualNetworkName": "MyVNET",
    "networkSecurityGroupName": "default-NSG",
    "securityProfileJson": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "[parameters('securityType')]"
    },
    "extensionName": "GuestAttestation",
    "extensionPublisher": "Microsoft.Azure.Security.WindowsAttestation",
    "extensionVersion": "1.0",
    "maaTenantName": "GuestAttestation",
    "maaEndpoint": "[substring('emptyString', 0, 0)]"
  },
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2022-05-01",
      "name": "[variables('storageAccountName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "Standard_LRS"
      },
      "kind": "Storage"
    },
    {
      "type": "Microsoft.Network/publicIPAddresses",
      "apiVersion": "2022-05-01",
      "name": "[parameters('publicIpName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "[parameters('publicIpSku')]"
      },
      "properties": {
        "publicIPAllocationMethod": "
[parameters('publicIPAllocationMethod')]",
        "dnsSettings": {
          "domainNameLabel": "[parameters('dnsLabelPrefix')]"
        }
```

```
      }
    },
    {
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2022-05-01",
      "name": "[variables('networkSecurityGroupName')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "default-allow-3389",
            "properties": {
              "priority": 1000,
              "access": "Allow",
              "direction": "Inbound",
              "destinationPortRange": "3389",
              "protocol": "Tcp",
              "sourcePortRange": "*",
              "sourceAddressPrefix": "*",
              "destinationAddressPrefix": "*"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2022-05-01",
      "name": "[variables('virtualNetworkName')]",
      "location": "[parameters('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets": [
          {
            "name": "[variables('subnetName')]",
            "properties": {
              "addressPrefix": "[variables('subnetPrefix')]",
              "networkSecurityGroup": {
                "id": "
[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
              }
            }
          }
        ]
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
      ]
    },
```

```
    {
      "type": "Microsoft.Network/networkInterfaces",
      "apiVersion": "2022-05-01",
      "name": "[variables('nicName')]",
      "location": "[parameters('location')]",
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": {
                "id": "[resourceId('Microsoft.Network/publicIPAddresses',
parameters('publicIpName'))]"
              },
              "subnet": {
                "id": "
[resourceId('Microsoft.Network/virtualNetworks/subnets',
variables('virtualNetworkName'), variables('subnetName'))]"
              }
            }
          }
        ]
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses',
parameters('publicIpName'))]",
        "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]"
      ]
    },
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2022-03-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "hardwareProfile": {
          "vmSize": "[parameters('vmSize')]"
        },
        "osProfile": {
          "computerName": "[parameters('vmName')]",
          "adminUsername": "[parameters('adminUsername')]",
          "adminPassword": "[parameters('adminPassword')]"
        },
        "storageProfile": {
          "imageReference": {
            "publisher": "MicrosoftWindowsServer",
            "offer": "WindowsServer",
            "sku": "[parameters('OSVersion')]",
            "version": "latest"
          },
          "osDisk": {
            "createOption": "FromImage",
            "managedDisk": {
```

```
              "storageAccountType": "StandardSSD_LRS"
            }
          },
          "dataDisks": [
            {
              "diskSizeGB": 1023,
              "lun": 0,
              "createOption": "Empty"
            }
          ]
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('nicName'))]"
            }
          ]
        },
        "diagnosticsProfile": {
          "bootDiagnostics": {
            "enabled": true,
            "storageUri": "
[reference(resourceId('Microsoft.Storage/storageAccounts',
variables('storageAccountName')), '2022-05-01').primaryEndpoints.blob]"
          }
        },
        "securityProfile": "[if(equals(parameters('securityType'),
'TrustedLaunch'), variables('securityProfileJson'), null())]"
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkInterfaces',
variables('nicName'))]",
        "[resourceId('Microsoft.Storage/storageAccounts',
variables('storageAccountName'))]"
      ]
    },
    {
      "condition": "[and(equals(parameters('securityType'),
'TrustedLaunch'),
and(equals(variables('securityProfileJson').uefiSettings.secureBootEnabled,
true()), equals(variables('securityProfileJson').uefiSettings.vTpmEnabled,
true())))]",
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "apiVersion": "2022-03-01",
      "name": "[format('{0}/{1}', parameters('vmName'),
variables('extensionName'))]",
      "location": "[parameters('location')]",
      "properties": {
        "publisher": "[variables('extensionPublisher')]",
        "type": "[variables('extensionName')]",
        "typeHandlerVersion": "[variables('extensionVersion')]",
        "autoUpgradeMinorVersion": true,
        "enableAutomaticUpgrade": true,
        "settings": {
```

```
          "AttestationConfig": {
            "MaaSettings": {
              "maaEndpoint": "[variables('maaEndpoint')]",
              "maaTenantName": "[variables('maaTenantName')]"
            }
          }
        }
      },
      "dependsOn": [
        "[resourceId('Microsoft.Compute/virtualMachines',
parameters('vmName'))]"
      ]
    }
  ],
  "outputs": {
    "hostname": {
      "type": "string",
      "value": "[reference(resourceId('Microsoft.Network/publicIPAddresses',
parameters('publicIpName')), '2022-05-01').dnsSettings.fqdn]"
    }
  }
}
```

Several resources are defined in the template:

- Microsoft.Network/virtualNetworks/subnets: create a subnet.
- Microsoft.Storage/storageAccounts: create a storage account.
- Microsoft.Network/publicIPAddresses: create a public IP address.
- Microsoft.Network/networkSecurityGroups: create a network security group.
- Microsoft.Network/virtualNetworks: create a virtual network.
- Microsoft.Network/networkInterfaces: create a NIC.
- Microsoft.Compute/virtualMachines: create a virtual machine.

Deploy the template

1. Select the following image to sign in to Azure and open a template. The template creates a key vault and a secret.

2. Select or enter the following values. Use the default values, when available.

   - **Subscription**: select an Azure subscription.
   - **Resource group**: select an existing resource group from the drop-down, or select **Create new**, enter a unique name for the resource group, and then select **OK**.

- **Region**: select a region. For example, **Central US**.
- **Admin username**: provide a username, such as *azureuser*.
- **Admin password**: provide a password to use for the admin account. The password must be at least 12 characters long and meet the defined complexity requirements.
- **DNS label prefix**: enter a unique identifier to use as part of the DNS label.
- **OS version**: select which OS version you want to run on the VM.
- **VM size**: select the size to use for the VM.
- **Location**: the default is the same location as the resource group, if it already exists.

3. Select **Review + create**. After validation completes, select **Create** to create and deploy the VM.

Cost information isn't presented during the virtual machine creation process for ARM templates like it is for the Azure portal. If you want to learn more about how cost works for virtual machines, see the Cost optimization Overview page.

The Azure portal is used to deploy the template. In addition to the Azure portal, you can also use the Azure PowerShell, Azure CLI, and REST API. To learn other deployment methods, see Deploy templates.

Review deployed resources

You can use the Azure portal to check on the VM and other resource that were created. After the deployment is finished, select **Resource groups** to see the VM and other resources.

Clean up resources

When no longer needed, delete the resource group, which deletes the VM and all of the resources in the resource group.

1. Select the **Resource group**.
2. On the page for the resource group, select **Delete resource group**.
3. When prompted, type the name of the resource group and then select **Delete**.

Next steps

In this quickstart, you deployed a simple virtual machine using a Resource Manager template. To learn more about Azure virtual machines, continue to the tutorial for

managing VMs.

Azure Windows virtual machine tutorials

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machines in a scale set using Azure portal

Article • 04/28/2025

This article steps through using Azure portal to create a Virtual Machine Scale Set.

Sign in to Azure

Sign in to the Azure portal.

Create a Virtual Machine Scale Set

You can deploy a scale set with a Windows Server image or Linux image such as RHEL, Ubuntu, or SLES.

1. In the Azure portal search bar, search for and select **Virtual Machine Scale Sets**.
2. Select **Create** on the **Virtual Machine Scale Sets** page.
3. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and create a new resource group called *myVMSSResourceGroup*.
4. Under **Scale set details**, set *myScaleSet* for your scale set name and select a **Region** that is close to your area.
5. Under **Orchestration**, select *Flexible* Orchestration Mode.
6. Under **Security type**, select *Trusted launch virtual machines*.
7. Under **Scaling**, keep it the Scaling mode as *Manual*.
8. Under **Instance count**, set the **initial instance count** field to *5*. You can set this number up to 1000
9. Under **Instance details**, select a marketplace image for **Image**. Select any of the Supported Distros.
10. Under **Administrator account** configure the admin username and set up an associated password or SSH public key.
    - A **Password** must be at least 12 characters long and meet three out of the four following complexity requirements: one lower case character, one upper case

character, one number, and one special character. For more information, see username and password requirements.

- If you select a Linux OS disk image, you can instead choose **SSH public key**. You can use an existing key or create a new one. In this example, we will have Azure generate a new key pair for us. For more information on generating key pairs, see create and use SSH keys.

Create a Virtual Machine Scale Set (VMSS) ...

Basics | Spot | Disks | Networking | Management | Health | Advanced | Tags | Review + create

At no added cost, a VMSS offers automatic scaling and performance optimization, infrastructure flexibility, and options to mix VM sizes, zones, and fault domains—all with simple, centralized group VM management. Already have VMs? Just create a new flexible VMSS, and attach existing VMs to your new scale set to get enhanced availability, resiliency, and capacity and cost optimization. Learn more about VMSS

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Visual Studio Enterprise Subscription

Resource group * : (New) myVMSSResourceGroup

Create new

Scale set details

Virtual machine scale set name * : myScaleSet

Region * : (US) East US

Availability zone : None

Orchestration

A scale set has a "scale set model" that defines the attributes of virtual machine instances (size, number of data disks, etc). As the number of instances in the scale set changes, new instances are added based on the scale set model.
Learn more about the scale set model

Orchestration mode * :
- (●) **Flexible:** achieve high availability at scale with identical or multiple virtual machine types
- ( ) **Uniform:** optimized for large scale stateless workloads

Security type : Trusted launch virtual machines

Configure security features

Scaling

Scaling mode :
- (●) **Manually update the capacity:** Maintain a fixed amount of instances.
- ( ) Autoscaling: Scaling based on a CPU metric, on any schedule.
- ( ) **No scaling profile:** manual attach virtual machines after deployment

Subscription needs Microsoft.Insights registration to use autoscaling. Learn more

Instance count * : 5

Configure scaling options

Instance details

Image * : Ubuntu Server 20.04 LTS - x64 Gen2

See all images | Configure VM generation

VM architecture :
- ( ) Arm64
- (●) x64

Size * :

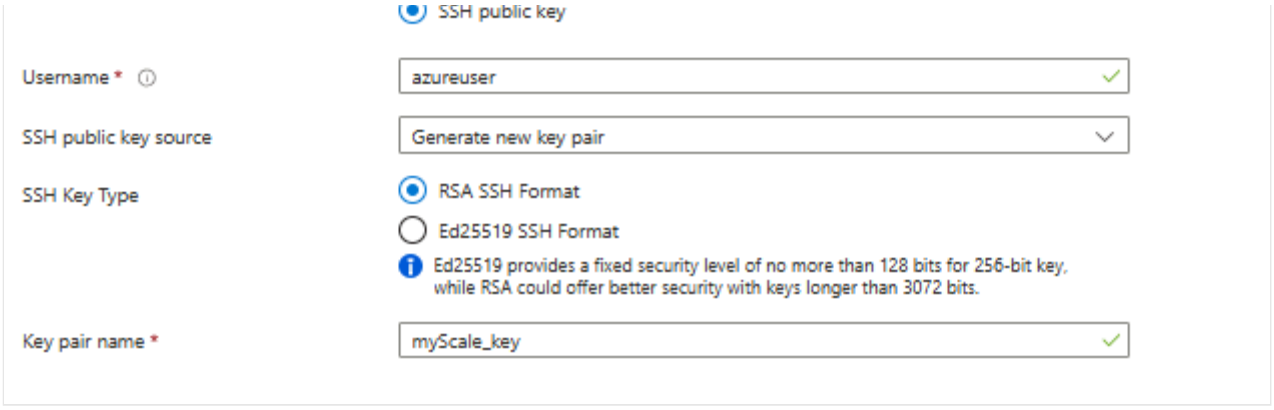
Selected sizes
Standard D2s v3
Select up to 5 sizes

Enable Hibernation : [ ]

Hibernate does not currently support Trusted launch and Confidential virtual machines for Linux images. Learn more

Administrator account

Authentication type :
- ( ) Password



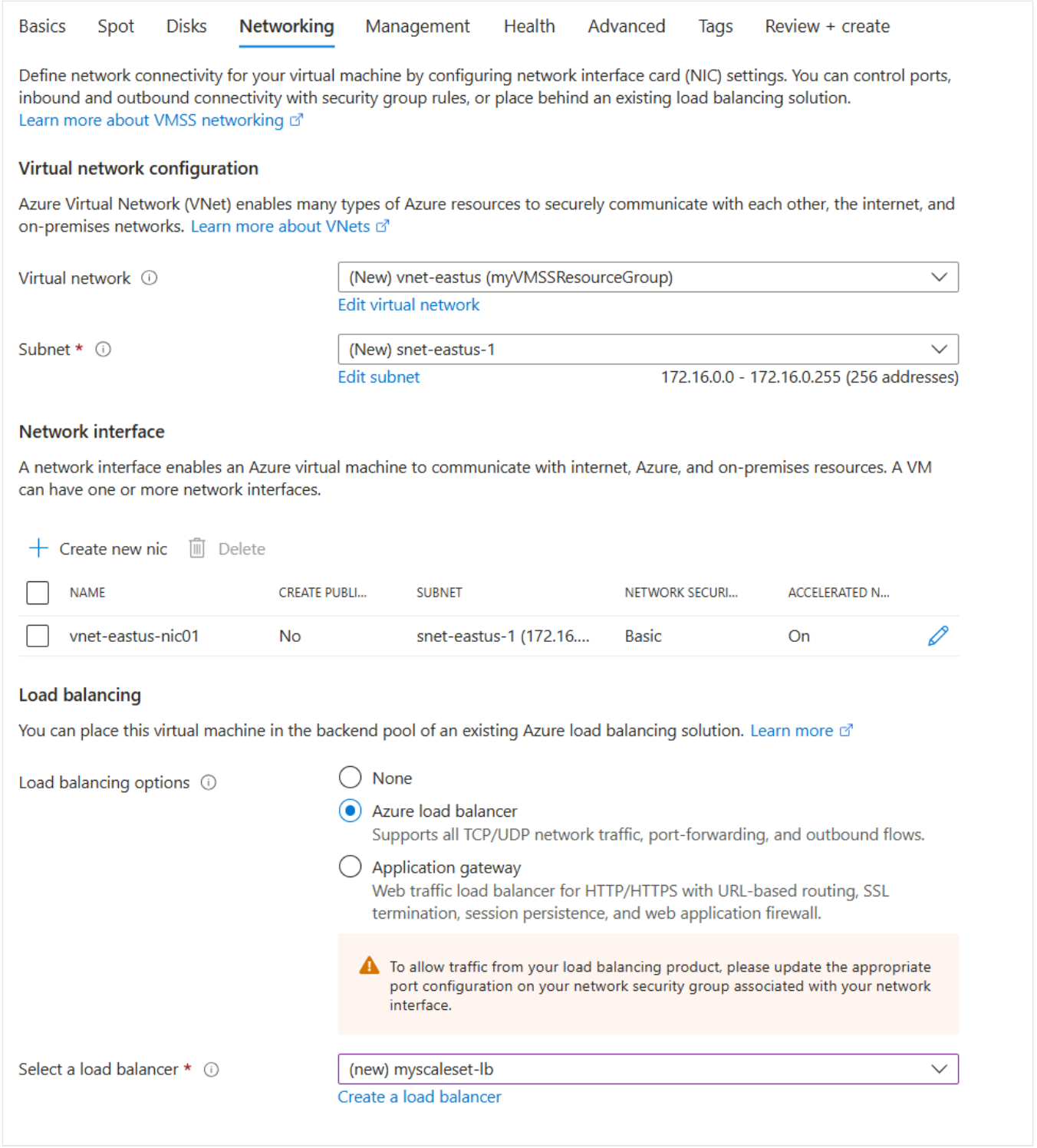
11. Select **Next: Spot** to move to the Spot virtual machine configuration options. For this quickstart, leave the default Spot configurations.

12. Select **Next: Disks** to move the disk configuration options. For this quickstart, leave the default disk configurations.

13. Select **Next: Networking** to move the networking configuration options.

14. On the **Networking** page, under **Load balancing**, select **Azure load balancer**.

15. In **Select a load balancer**, select a load balancer or create a new one.



16. When you're done, select **Review + create**.

17. After it passes validation, select **Create** to deploy the scale set.

Clean up resources

When no longer needed, delete the resource group, scale set, and all related resources. To do so, select the resource group for the scale set and then select **Delete**.

Next steps

Learn how to create a Flexible scale with Azure CLI.

Create virtual machines in a scale set using Azure CLI

Article • 09/19/2024

Deploy to Azure

This article steps through using the Azure CLI to create a Virtual Machine Scale Set.

Make sure that you've installed the latest Azure CLI and are logged in to an Azure account with az login.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, select **Open Cloud Shell** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/cli. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Define environment variables

Define environment variables as follows.

Bash

```
export RANDOM_ID="$(openssl rand -hex 3)"
export MY_RESOURCE_GROUP_NAME="myVMSSResourceGroup$RANDOM_ID"
export REGION=EastUS
export MY_VMSS_NAME="myVMSS$RANDOM_ID"
export MY_USERNAME=azureuser
export MY_VM_IMAGE="Ubuntu2204"
export MY_VNET_NAME="myVNet$RANDOM_ID"
export NETWORK_PREFIX="$(($RANDOM % 254 + 1))"
export MY_VNET_PREFIX="10.$NETWORK_PREFIX.0.0/16"
export MY_VM_SN_NAME="myVMSN$RANDOM_ID"
export MY_VM_SN_PREFIX="10.$NETWORK_PREFIX.0.0/24"
export MY_APPGW_SN_NAME="myAPPGWSN$RANDOM_ID"
export MY_APPGW_SN_PREFIX="10.$NETWORK_PREFIX.1.0/24"
export MY_APPGW_NAME="myAPPGW$RANDOM_ID"
export MY_APPGW_PUBLIC_IP_NAME="myAPPGWPublicIP$RANDOM_ID"
```

Create a resource group

A resource group is a logical container into which Azure resources are deployed and managed. All resources must be placed in a resource group. The following command creates a resource group with the previously defined $MY_RESOURCE_GROUP_NAME and $REGION parameters.

Bash

```
az group create --name $MY_RESOURCE_GROUP_NAME --location $REGION -o JSON
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx",
  "location": "eastus",
  "managedBy": null,
  "name": "myVMSSResourceGroupxxxxxx",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```


Create network resources

Now you'll create network resources. In this step you're going to create a virtual network, one subnet 1 for Application Gateway, and one subnet for VMs. You also need to have a public IP to attach your Application Gateway to reach your web application from the internet.

Create virtual network and subnet

Bash

```
az network vnet create  --name $MY_VNET_NAME  --resource-group
$MY_RESOURCE_GROUP_NAME --location $REGION  --address-prefix $MY_VNET_PREFIX
--subnet-name $MY_VM_SN_NAME --subnet-prefix $MY_VM_SN_PREFIX -o JSON
```

Results:

JSON


```
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.X.0.0/16"
      ]
    },
    "enableDdosProtection": false,
    "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx",
    "location": "eastus",
    "name": "myVNetxxxxxx",
    "provisioningState": "Succeeded",
    "resourceGroup": "myVMSSResourceGroupxxxxxx",
    "resourceGuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "subnets": [
      {
        "addressPrefix": "10.X.0.0/24",
        "delegations": [],
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx/subnets/myVMSNxxxxxx",
        "name": "myVMSNxxxxxx",
        "privateEndpointNetworkPolicies": "Disabled",
        "privateLinkServiceNetworkPolicies": "Enabled",
        "provisioningState": "Succeeded",
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type": "Microsoft.Network/virtualNetworks/subnets"
      }
    ],
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
```


Create Application Gateway resources

Azure Application Gateway requires a dedicated subnet within your virtual network. The following command creates a subnet named $MY_APPGW_SN_NAME with a specified address prefix named $MY_APPGW_SN_PREFIX in your virtual network $MY_VNET_NAME.

Bash

```
az network vnet subnet create  --name $MY_APPGW_SN_NAME  --resource-group
$MY_RESOURCE_GROUP_NAME --vnet-name  $MY_VNET_NAME --address-prefix
$MY_APPGW_SN_PREFIX -o JSON
```

Results:

JSON


```
{
  "addressPrefix": "10.66.1.0/24",
  "delegations": [],
  "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx/subnets/myAPPGWSNxxxxxx",
  "name": "myAPPGWSNxxxxxx",
  "privateEndpointNetworkPolicies": "Disabled",
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
  "resourceGroup": "myVMSSResourceGroupxxxxxx",
  "type": "Microsoft.Network/virtualNetworks/subnets"
}
```


The following command creates a standard, zone redundant, static, public IPv4 in your resource group.

Bash

```
az network public-ip create  --resource-group $MY_RESOURCE_GROUP_NAME --name
$MY_APPGW_PUBLIC_IP_NAME --sku Standard   --location $REGION  --allocation-
method static --version IPv4 --zone 1 2 3 -o JSON
```

Results:

JSON


```
{
  "publicIp": {
    "ddosSettings": {
      "protectionMode": "VirtualNetworkInherited"
    },
    "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/publicIPAddresses//myAPPGWPublicIPxxxxxx",
    "idleTimeoutInMinutes": 4,
    "ipAddress": "X.X.X.X",
    "ipTags": [],
    "location": "eastus",
```

```
    "name": "/myAPPGWPublicIPxxxxxx",
    "provisioningState": "Succeeded",
    "publicIPAddressVersion": "IPv4",
    "publicIPAllocationMethod": "Static",
    "resourceGroup": "myVMSSResourceGroupxxxxxx",
    "resourceGuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "sku": {
      "name": "Standard",
      "tier": "Regional"
    },
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": [
      "1",
      "2",
      "3"
    ]
  }
}
```


In this step, you create an Application Gateway that you're going to integrate with your Virtual Machine Scale Set. This example creates a zone redundant Application Gateway with Standard_v2 SKU and enables Http communication for the Application Gateway. The public IP $MY_APPGW_PUBLIC_IP_NAME created in previous step is attached to the Application Gateway.

Bash

```
az network application-gateway create   --name $MY_APPGW_NAME --location
$REGION --resource-group $MY_RESOURCE_GROUP_NAME --vnet-name $MY_VNET_NAME -
-subnet $MY_APPGW_SN_NAME --capacity 2  --zones 1 2 3 --sku Standard_v2   --
http-settings-cookie-based-affinity Disabled   --frontend-port 80 --http-
settings-port 80   --http-settings-protocol Http --public-ip-address
$MY_APPGW_PUBLIC_IP_NAME --priority 1001 -o JSON
```

JSON


```
{
  "applicationGateway": {
    "backendAddressPools": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/backendAddressPools/appGatewayBacken
dPool",
        "name": "appGatewayBackendPool",
        "properties": {
          "backendAddresses": [],
          "provisioningState": "Succeeded",
          "requestRoutingRules": [
```

```
            {
              "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/requestRoutingRules/rule1",
              "resourceGroup": "myVMSSResourceGroupxxxxxx"
            }
          ]
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type": "Microsoft.Network/applicationGateways/backendAddressPools"
      }
    ],
    "backendHttpSettingsCollection": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/backendHttpSettingsCollection/appGat
ewayBackendHttpSettings",
        "name": "appGatewayBackendHttpSettings",
        "properties": {
          "connectionDraining": {
            "drainTimeoutInSec": 1,
            "enabled": false
          },
          "cookieBasedAffinity": "Disabled",
          "pickHostNameFromBackendAddress": false,
          "port": 80,
          "protocol": "Http",
          "provisioningState": "Succeeded",
          "requestRoutingRules": [
            {
              "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/requestRoutingRules/rule1",
              "resourceGroup": "myVMSSResourceGroupxxxxxx"
            }
          ],
          "requestTimeout": 30
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type":
"Microsoft.Network/applicationGateways/backendHttpSettingsCollection"
      }
    ],
    "backendSettingsCollection": [],
    "frontendIPConfigurations": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/frontendIPConfigurations/appGatewayF
rontendIP",
        "name": "appGatewayFrontendIP",
        "properties": {
```

```
          "httpListeners": [
            {
              "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/httpListeners/appGatewayHttpListener
",
              "resourceGroup": "myVMSSResourceGroupxxxxxx"
            }
          ],
          "privateIPAllocationMethod": "Dynamic",
          "provisioningState": "Succeeded",
          "publicIPAddress": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/publicIPAddresses/myAPPGWPublicIPxxxxxx",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          }
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type":
"Microsoft.Network/applicationGateways/frontendIPConfigurations"
      }
    ],
    "frontendPorts": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/frontendPorts/appGatewayFrontendPort
",
        "name": "appGatewayFrontendPort",
        "properties": {
          "httpListeners": [
            {
              "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/httpListeners/appGatewayHttpListener
",
              "resourceGroup": "myVMSSResourceGroupxxxxxx"
            }
          ],
          "port": 80,
          "provisioningState": "Succeeded"
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type": "Microsoft.Network/applicationGateways/frontendPorts"
      }
    ],
    "gatewayIPConfigurations": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/gatewayIPConfigurations/appGatewayFr
ontendIP",
```

```
        "name": "appGatewayFrontendIP",
        "properties": {
          "provisioningState": "Succeeded",
          "subnet": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx/subnets/myAPPGWSNxxxxxx",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          }
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type":
"Microsoft.Network/applicationGateways/gatewayIPConfigurations"
      }
    ],
    "httpListeners": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/httpListeners/appGatewayHttpListener
",
        "name": "appGatewayHttpListener",
        "properties": {
          "frontendIPConfiguration": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/frontendIPConfigurations/appGatewayF
rontendIP",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          },
          "frontendPort": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/frontendPorts/appGatewayFrontendPort
",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          },
          "hostNames": [],
          "protocol": "Http",
          "provisioningState": "Succeeded",
          "requestRoutingRules": [
            {
              "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/requestRoutingRules/rule1",
              "resourceGroup": "myVMSSResourceGroupxxxxxx"
            }
          ],
          "requireServerNameIndication": false
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type": "Microsoft.Network/applicationGateways/httpListeners"
      }
    ],
```

```
    "listeners": [],
    "loadDistributionPolicies": [],
    "operationalState": "Running",
    "privateEndpointConnections": [],
    "privateLinkConfigurations": [],
    "probes": [],
    "provisioningState": "Succeeded",
    "redirectConfigurations": [],
    "requestRoutingRules": [
      {
        "etag": "W/\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/requestRoutingRules/rule1",
        "name": "rule1",
        "properties": {
          "backendAddressPool": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/backendAddressPools/appGatewayBacken
dPool",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          },
          "backendHttpSettings": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/backendHttpSettingsCollection/appGat
ewayBackendHttpSettings",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          },
          "httpListener": {
            "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Ne
twork/applicationGateways/myAPPGWxxxxxx/httpListeners/appGatewayHttpListener
",
            "resourceGroup": "myVMSSResourceGroupxxxxxx"
          },
          "priority": 1001,
          "provisioningState": "Succeeded",
          "ruleType": "Basic"
        },
        "resourceGroup": "myVMSSResourceGroupxxxxxx",
        "type": "Microsoft.Network/applicationGateways/requestRoutingRules"
      }
    ],
    "resourceGuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "rewriteRuleSets": [],
    "routingRules": [],
    "sku": {
      "capacity": 2,
      "family": "Generation_1",
      "name": "Standard_v2",
      "tier": "Standard_v2"
    },
    "sslCertificates": [],
```

```
    "sslProfiles": [],
    "trustedClientCertificates": [],
    "trustedRootCertificates": [],
    "urlPathMaps": []
  }
}
```

Create a Virtual Machine Scale Set

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Now create a Virtual Machine Scale Set with az vmss create. The following example creates a zone redundant scale set with an instance count of *2* with public IP in subnet $MY_VM_SN_NAME within your resource group $MY_RESOURCE_GROUP_NAME, integrates the Application Gateway, and generates SSH keys. Make sure to save the SSH keys if you need to log into your VMs via ssh.

Bash

```
az vmss create --name $MY_VMSS_NAME --resource-group $MY_RESOURCE_GROUP_NAME
--image $MY_VM_IMAGE --admin-username $MY_USERNAME --generate-ssh-keys --
public-ip-per-vm --orchestration-mode Uniform --instance-count 2 --zones 1 2
3 --vnet-name $MY_VNET_NAME --subnet $MY_VM_SN_NAME --vm-sku Standard_DS2_v2
--upgrade-policy-mode Automatic --app-gateway $MY_APPGW_NAME --backend-pool-
name appGatewayBackendPool -o JSON
```

Results:

JSON

```
{
  "vmss": {
    "doNotRunExtensionsOnOverprovisionedVMs": false,
    "orchestrationMode": "Uniform",
    "overprovision": true,
    "platformFaultDomainCount": 1,
    "provisioningState": "Succeeded",
    "singlePlacementGroup": false,
    "timeCreated": "20xx-xx-xxTxx:xx:xx.xxxxxx+00:00",
```

```
    "uniqueId": "xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx",
    "upgradePolicy": {
      "mode": "Automatic",
      "rollingUpgradePolicy": {
        "maxBatchInstancePercent": 20,
        "maxSurge": false,
        "maxUnhealthyInstancePercent": 20,
        "maxUnhealthyUpgradedInstancePercent": 20,
        "pauseTimeBetweenBatches": "PT0S",
        "rollbackFailedInstancesOnPolicyBreach": false
      }
    },
    "virtualMachineProfile": {
      "networkProfile": {
        "networkInterfaceConfigurations": [
          {
            "name": "myvmsa53cNic",
            "properties": {
              "disableTcpStateTracking": false,
              "dnsSettings": {
                "dnsServers": []
              },
              "enableAcceleratedNetworking": false,
              "enableIPForwarding": false,
              "ipConfigurations": [
                {
                  "name": "myvmsa53cIPConfig",
                  "properties": {
                    "applicationGatewayBackendAddressPools": [
                      {
                        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxx/providers/Microsoft.Net
work/applicationGateways/myAPPGW7xxxxx/backendAddressPools/appGatewayBackend
Pool",
                        "resourceGroup": "myVMSSResourceGroupxxxxxx"
                      }
                    ],
                    "privateIPAddressVersion": "IPv4",
                    "publicIPAddressConfiguration": {
                      "name": "instancepublicip",
                      "properties": {
                        "idleTimeoutInMinutes": 10,
                        "ipTags": [],
                        "publicIPAddressVersion": "IPv4"
                      }
                    },
                    "subnet": {
                      "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxx/providers/Microsoft.Net
work/virtualNetworks/myVNetxxxxx/subnets/myVMSN7xxxxx",
                      "resourceGroup": "myVMSSResourceGroupxxxxxxx"
                    }
                  }
                }
              ],
```

```
              "primary": true
            }
          }
        ]
      },
      "osProfile": {
        "adminUsername": "azureuser",
        "allowExtensionOperations": true,
        "computerNamePrefix": "myvmsa53c",
        "linuxConfiguration": {
          "disablePasswordAuthentication": true,
          "enableVMAgentPlatformUpdates": false,
          "provisionVMAgent": true,
          "ssh": {
            "publicKeys": [
              {
                "keyData": "ssh-rsa xxxxxxxx",
                "path": "/home/azureuser/.ssh/authorized_keys"
              }
            ]
          }
        },
        "requireGuestProvisionSignal": true,
        "secrets": []
      },
      "storageProfile": {
        "diskControllerType": "SCSI",
        "imageReference": {
          "offer": "0001-com-ubuntu-server-jammy",
          "publisher": "Canonical",
          "sku": "22_04-lts-gen2",
          "version": "latest"
        },
        "osDisk": {
          "caching": "ReadWrite",
          "createOption": "FromImage",
          "diskSizeGB": 30,
          "managedDisk": {
            "storageAccountType": "Premium_LRS"
          },
          "osType": "Linux"
        }
      },
      "timeCreated": "20xx-xx-xxTxx:xx:xx.xxxxxx+00:00"
    },
    "zoneBalance": false
  }
}
```


Install ngnix with Virtual Machine Scale Sets extensions

The following command uses the Virtual Machine Scale Sets extension to run a custom script that installs ngnix and publishes a page that shows the hostname of the Virtual Machine that your HTTP requests hits.

Bash

```
az vmss extension set --publisher Microsoft.Azure.Extensions --version 2.0
--name CustomScript --resource-group $MY_RESOURCE_GROUP_NAME --vmss-name
$MY_VMSS_NAME --settings '{ "fileUris":
["https://raw.githubusercontent.com/Azure-Samples/compute-automation-
configurations/master/automate_nginx.sh"], "commandToExecute":
"./automate_nginx.sh" }' -o JSON
```

Results:

JSON


```
{
  "additionalCapabilities": null,
  "automaticRepairsPolicy": null,
  "constrainedMaximumCapacity": null,
  "doNotRunExtensionsOnOverprovisionedVMs": false,
  "extendedLocation": null,
  "hostGroup": null,
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxx/providers/Microsoft.Com
pute/virtualMachineScaleSets/myVMSSxxxxx",
  "identity": null,
  "location": "eastus",
  "name": "myVMSSxxxx",
  "orchestrationMode": "Uniform",
  "overprovision": true,
  "plan": null,
  "platformFaultDomainCount": 1,
  "priorityMixPolicy": null,
  "provisioningState": "Succeeded",
  "proximityPlacementGroup": null,
  "resourceGroup": "myVMSSResourceGroupxxxxx",
  "scaleInPolicy": null,
  "singlePlacementGroup": false,
  "sku": {
    "capacity": 2,
    "name": "Standard_DS2_v2",
    "tier": "Standard"
  },
  "spotRestorePolicy": null,
  "tags": {},
  "timeCreated": "20xx-xx-xxTxx:xx:xx.xxxxxx+00:00",
  "type": "Microsoft.Compute/virtualMachineScaleSets",
  "uniqueId": "xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx",
  "upgradePolicy": {
    "automaticOsUpgradePolicy": null,
```

```
    "mode": "Automatic",
    "rollingUpgradePolicy": {
      "enableCrossZoneUpgrade": null,
      "maxBatchInstancePercent": 20,
      "maxSurge": false,
      "maxUnhealthyInstancePercent": 20,
      "maxUnhealthyUpgradedInstancePercent": 20,
      "pauseTimeBetweenBatches": "PT0S",
      "prioritizeUnhealthyInstances": null,
      "rollbackFailedInstancesOnPolicyBreach": false
    }
  },
  "virtualMachineProfile": {
    "applicationProfile": null,
    "billingProfile": null,
    "capacityReservation": null,
    "diagnosticsProfile": null,
    "evictionPolicy": null,
    "extensionProfile": {
      "extensions": [
        {
          "autoUpgradeMinorVersion": true,
          "enableAutomaticUpgrade": null,
          "forceUpdateTag": null,
          "id": null,
          "name": "CustomScript",
          "protectedSettings": null,
          "protectedSettingsFromKeyVault": null,
          "provisionAfterExtensions": null,
          "provisioningState": null,
          "publisher": "Microsoft.Azure.Extensions",
          "settings": {
            "commandToExecute": "./automate_nginx.sh",
            "fileUris": [
              "https://raw.githubusercontent.com/Azure-Samples/compute-
automation-configurations/master/automate_nginx.sh"
            ]
          },
          "suppressFailures": null,
          "type": null,
          "typeHandlerVersion": "2.0",
          "typePropertiesType": "CustomScript"
        }
      ],
      "extensionsTimeBudget": null
    },
    "hardwareProfile": null,
    "licenseType": null,
    "networkProfile": {
      "healthProbe": null,
      "networkApiVersion": null,
      "networkInterfaceConfigurations": [
        {
          "deleteOption": null,
          "disableTcpStateTracking": false,
```

```
          "dnsSettings": {
            "dnsServers": []
          },
          "enableAcceleratedNetworking": false,
          "enableFpga": null,
          "enableIpForwarding": false,
          "ipConfigurations": [
            {
              "applicationGatewayBackendAddressPools": [
                {
                  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxx/providers/Microsoft.Net
work/applicationGateways/myAPPGWxxxx/backendAddressPools/appGatewayBackendPo
ol",
                  "resourceGroup": "myVMSSResourceGroupxxxxxx"
                }
              ],
              "applicationSecurityGroups": null,
              "loadBalancerBackendAddressPools": null,
              "loadBalancerInboundNatPools": null,
              "name": "myvmsdxxxIPConfig",
              "primary": null,
              "privateIpAddressVersion": "IPv4",
              "publicIpAddressConfiguration": null,
              "subnet": {
                "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxx/providers/Microsoft.Net
work/virtualNetworks/myVNetxxxxx/subnets/myVMSNxxxxx",
                "resourceGroup": "myVMSSResourceGroupaxxxxx"
              }
            }
          ],
          "name": "myvmsxxxxxx",
          "networkSecurityGroup": null,
          "primary": true
        }
      ]
    },
    "osProfile": {
      "adminPassword": null,
      "adminUsername": "azureuser",
      "allowExtensionOperations": true,
      "computerNamePrefix": "myvmsdxxx",
      "customData": null,
      "linuxConfiguration": {
        "disablePasswordAuthentication": true,
        "enableVmAgentPlatformUpdates": false,
        "patchSettings": null,
        "provisionVmAgent": true,
        "ssh": {
          "publicKeys": [
            {
              "keyData": "ssh-rsa xxxxxxxx",
              "path": "/home/azureuser/.ssh/authorized_keys"
            }
```

```
            ]
          }
        },
        "requireGuestProvisionSignal": true,
        "secrets": [],
        "windowsConfiguration": null
      },
      "priority": null,
      "scheduledEventsProfile": null,
      "securityPostureReference": null,
      "securityProfile": null,
      "serviceArtifactReference": null,
      "storageProfile": {
        "dataDisks": null,
        "diskControllerType": "SCSI",
        "imageReference": {
          "communityGalleryImageId": null,
          "exactVersion": null,
          "id": null,
          "offer": "0001-com-ubuntu-server-jammy",
          "publisher": "Canonical",
          "sharedGalleryImageId": null,
          "sku": "22_04-lts-gen2",
          "version": "latest"
        },
        "osDisk": {
          "caching": "ReadWrite",
          "createOption": "FromImage",
          "deleteOption": null,
          "diffDiskSettings": null,
          "diskSizeGb": 30,
          "image": null,
          "managedDisk": {
            "diskEncryptionSet": null,
            "securityProfile": null,
            "storageAccountType": "Premium_LRS"
          },
          "name": null,
          "osType": "Linux",
          "vhdContainers": null,
          "writeAcceleratorEnabled": null
        }
      },
      "userData": null
    },
    "zoneBalance": false,
    "zones": [
      "1",
      "2",
      "3"
    ]
}
```

Define an autoscale profile

To enable autoscale on a scale set, first define an autoscale profile. This profile defines the default, minimum, and maximum scale set capacity. These limits let you control cost by not continually creating VM instances and balance acceptable performance with a minimum number of instances that remain in a scale-in event. The following example sets the default, minimum capacity of two VM instances, and a maximum capacity of 10:

Bash

```
az monitor autoscale create --resource-group $MY_RESOURCE_GROUP_NAME --
resource  $MY_VMSS_NAME --resource-type
Microsoft.Compute/virtualMachineScaleSets --name autoscale --min-count 2 --
max-count 10 --count 2
```

Results:

JSON


```
{
  "enabled": true,
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxx/providers/microsoft.ins
ights/autoscalesettings/autoscale",
  "location": "eastus",
  "name": "autoscale",
  "namePropertiesName": "autoscale",
  "notifications": [
    {
      "email": {
        "customEmails": [],
        "sendToSubscriptionAdministrator": false,
        "sendToSubscriptionCoAdministrators": false
      },
      "webhooks": []
    }
  ],
  "predictiveAutoscalePolicy": {
    "scaleLookAheadTime": null,
    "scaleMode": "Disabled"
  },
  "profiles": [
    {
      "capacity": {
        "default": "2",
        "maximum": "10",
        "minimum": "2"
      },
      "fixedDate": null,
      "name": "default",
```

```
      "recurrence": null,
      "rules": []
    }
  ],
  "resourceGroup": "myVMSSResourceGroupxxxxx",
  "systemData": null,
  "tags": {},
  "targetResourceLocation": null,
  "targetResourceUri": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Co
mpute/virtualMachineScaleSets/myVMSSxxxxxx",
  "type": "Microsoft.Insights/autoscaleSettings"
}
```


Create a rule to autoscale out

The following command creates a rule that increases the number of VM instances in a scale set when the average CPU load is greater than 70% over a 5-minute period. When the rule triggers, the number of VM instances increases by three.

Bash

```
az monitor autoscale rule create --resource-group $MY_RESOURCE_GROUP_NAME --
autoscale-name autoscale --condition "Percentage CPU > 70 avg 5m" --scale
out 3
```

Results:

JSON


```
{
  "metricTrigger": {
    "dimensions": [],
    "dividePerInstance": null,
    "metricName": "Percentage CPU",
    "metricNamespace": null,
    "metricResourceLocation": null,
    "metricResourceUri": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Co
mpute/virtualMachineScaleSets/myVMSSxxxxxx",
    "operator": "GreaterThan",
    "statistic": "Average",
    "threshold": "70",
    "timeAggregation": "Average",
    "timeGrain": "PT1M",
    "timeWindow": "PT5M"
  },
  "scaleAction": {
    "cooldown": "PT5M",
```

```
    "direction": "Increase",
    "type": "ChangeCount",
    "value": "3"
  }
}
```


Create a rule to autoscale in

Create another rule with `az monitor autoscale rule create` that decreases the number of VM instances in a scale set when the average CPU load then drops below 30% over a 5-minute period. The following example defines the rule to scale in the number of VM instances by one.

Bash

```
az monitor autoscale rule create --resource-group  $MY_RESOURCE_GROUP_NAME --autoscale-name autoscale --condition "Percentage CPU < 30 avg 5m" --scale in 1
```

Results:

JSON


```
{
  "metricTrigger": {
    "dimensions": [],
    "dividePerInstance": null,
    "metricName": "Percentage CPU",
    "metricNamespace": null,
    "metricResourceLocation": null,
    "metricResourceUri": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/myVMSSResourceGroupxxxxxx/providers/Microsoft.Compute/virtualMachineScaleSets/myVMSSxxxxxx",
    "operator": "LessThan",
    "statistic": "Average",
    "threshold": "30",
    "timeAggregation": "Average",
    "timeGrain": "PT1M",
    "timeWindow": "PT5M"
  },
  "scaleAction": {
    "cooldown": "PT5M",
    "direction": "Decrease",
    "type": "ChangeCount",
    "value": "1"
  }
}
```

Test the page

The following command shows you the public IP of your Application Gateway. Paste the IP address into a browser page for testing.

Bash

```
az network public-ip show --resource-group $MY_RESOURCE_GROUP_NAME --name
$MY_APPGW_PUBLIC_IP_NAME --query [ipAddress]  --output tsv
```

Clean up resources (optional)

To avoid Azure charges, you should clean up unneeded resources. When you no longer need your scale set and other resources, delete the resource group and all its resources with az group delete. The `--no-wait` parameter returns control to the prompt without waiting for the operation to complete. The `--yes` parameter confirms that you wish to delete the resources without another prompt to do so. This tutorial cleans up resources for you.

Next steps

- Learn how to create a scale set in the Azure portal.
- Learn about Virtual Machine Scale Sets.
- Automatically scale a Virtual Machine Scale Set with the Azure CLI

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machines in a scale set using PowerShell

Article • 08/22/2024

This article steps through using PowerShell to create a Virtual Machine Scale Set.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create resource group

Create an Azure resource group with New-AzResourceGroup. A resource group is a logical container into which Azure resources are deployed and managed.

Azure PowerShell

```
New-AzResourceGroup -Name 'myVMSSResourceGroup' -Location 'EastUS'
```

Create a Virtual Machine Scale Set

Now create a Virtual Machine Scale Set with New-AzVmss. The following example creates a scale set with an instance count of *two* running Windows Server 2019 Datacenter edition.

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Azure PowerShell

```
New-AzVmss `
    -ResourceGroup "myVMSSResourceGroup" `
    -Name "myScaleSet" `
    -OrchestrationMode "Flexible" `
    -Location "East US" `
    -InstanceCount "2" `
    -ImageName "Win2019Datacenter"
```

Clean up resources

When you delete a resource group, all resources contained within, such as the VM instances, virtual network, and disks, are also deleted. The `-Force` parameter confirms that you wish to delete the resources without another prompt to do so. The `-AsJob` parameter returns control to the prompt without waiting for the operation to complete.

Azure PowerShell

```
Remove-AzResourceGroup -Name "myResourceGroup" -Force -AsJob
```

Next steps

Learn how to create a scale set in the Azure portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a Windows Virtual Machine Scale Set with Bicep

Article • 08/22/2024

A Virtual Machine Scale Set allows you to deploy and manage a set of auto-scaling virtual machines. You can scale the number of VMs in the Virtual Machine Scale Set manually, or define rules to autoscale based on resource usage like CPU, memory demand, or network traffic. An Azure load balancer then distributes traffic to the VM instances in the Virtual Machine Scale Set. In this quickstart, you create a Virtual Machine Scale Set and deploy a sample application with Bicep.

Bicep is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. It provides concise syntax, reliable type safety, and support for code reuse. Bicep offers the best authoring experience for your infrastructure-as-code solutions in Azure.

Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

Review the Bicep file

The Bicep file used in this quickstart is from Azure Quickstart Templates.

Bicep

```
@description('String used as a base for naming resources. Must be 3-61
characters in length and globally unique across Azure. A hash is prepended
to this string for some resources, and resource-specific information is
appended.')
@minLength(3)
@maxLength(61)
param vmssName string

@description('Size of VMs in the VM Scale Set.')
param vmSku string = 'Standard_D2s_v3'

@description('The Windows version for the VM. This will pick a fully patched
image of this given Windows version. Allowed values: 2008-R2-SP1, 2012-
Datacenter, 2012-R2-Datacenter & 2016-Datacenter, 2019-Datacenter.')
@allowed([
  '2019-DataCenter-GenSecond'
  '2016-DataCenter-GenSecond'
  '2022-datacenter-azure-edition'
```

```
])
param windowsOSVersion string = '2022-datacenter-azure-edition'

@description('Security Type of the Virtual Machine.')
@allowed([
  'Standard'
  'TrustedLaunch'
])
param securityType string = 'TrustedLaunch'

@description('Number of VM instances (100 or less).')
@minValue(1)
@maxValue(100)
param instanceCount int = 3

@description('When true this limits the scale set to a single placement
group, of max size 100 virtual machines. NOTE: If singlePlacementGroup is
true, it may be modified to false. However, if singlePlacementGroup is
false, it may not be modified to true.')
param singlePlacementGroup bool = true

@description('Admin username on all VMs.')
param adminUsername string = 'vmssadmin'

@description('Admin password on all VMs.')
@secure()
param adminPassword string

@description('The base URI where artifacts required by this template are
located. For example, if stored on a public GitHub repo, you\'d use the
following URI: https://raw.githubusercontent.com/Azure/azure-quickstart-
templates/master/201-vmss-windows-webapp-dsc-autoscale/.')
param _artifactsLocation string = deployment().properties.templateLink.uri

@description('The sasToken required to access _artifactsLocation.  If your
artifacts are stored on a public repo or public storage account you can
leave this blank.')
@secure()
param _artifactsLocationSasToken string = ''

@description('Location of the PowerShell DSC zip file relative to the URI
specified in the _artifactsLocation, i.e. DSC/IISInstall.ps1.zip')
param powershelldscZip string = 'DSC/InstallIIS.zip'

@description('Location of the  of the WebDeploy package zip file relative to
the URI specified in _artifactsLocation, i.e.
WebDeploy/DefaultASPWebApp.v1.0.zip')
param webDeployPackage string = 'WebDeploy/DefaultASPWebApp.v1.0.zip'

@description('Version number of the DSC deployment. Changing this value on
subsequent deployments will trigger the extension to run.')
param powershelldscUpdateTagVersion string = '1.0'

@description('Location for all resources.')
param location string = resourceGroup().location
```

```
@description('Fault Domain count for each placement group.')
param platformFaultDomainCount int = 1

var vmScaleSetName =
toLower(substring('vmssName${uniqueString(resourceGroup().id)}', 0, 9))
var longvmScaleSet = toLower(vmssName)
var addressPrefix = '10.0.0.0/16'
var subnetPrefix = '10.0.0.0/24'
var vNetName = '${vmScaleSetName}vnet'
var publicIPAddressName = '${vmScaleSetName}pip'
var subnetName = '${vmScaleSetName}subnet'
var loadBalancerName = '${vmScaleSetName}lb'
var publicIPAddressID = publicIPAddress.id
var lbProbeID = resourceId('Microsoft.Network/loadBalancers/probes',
loadBalancerName, 'tcpProbe')
var natPoolName = '${vmScaleSetName}natpool'
var bePoolName = '${vmScaleSetName}bepool'
var lbPoolID =
resourceId('Microsoft.Network/loadBalancers/backendAddressPools',
loadBalancerName, bePoolName)
var natStartPort = 50000
var natEndPort = 50119
var natBackendPort = 3389
var nicName = '${vmScaleSetName}nic'
var ipConfigName = '${vmScaleSetName}ipconfig'
var frontEndIPConfigID =
resourceId('Microsoft.Network/loadBalancers/frontendIPConfigurations',
loadBalancerName, 'loadBalancerFrontEnd')
var osType = {
  publisher: 'MicrosoftWindowsServer'
  offer: 'WindowsServer'
  sku: windowsOSVersion
  version: 'latest'
}
var securityProfileJson = {
  uefiSettings: {
    secureBootEnabled: true
    vTpmEnabled: true
  }
  securityType: securityType
}
var imageReference = osType
var webDeployPackageFullPath = uri(_artifactsLocation,
'${webDeployPackage}${_artifactsLocationSasToken}')
var powershelldscZipFullPath = uri(_artifactsLocation,
'${powershelldscZip}${_artifactsLocationSasToken}')

resource loadBalancer 'Microsoft.Network/loadBalancers@2023-04-01' = {
  name: loadBalancerName
  location: location
  properties: {
    frontendIPConfigurations: [
      {
        name: 'LoadBalancerFrontEnd'
```

```
        properties: {
          publicIPAddress: {
            id: publicIPAddressID
          }
        }
      }
    ]
    backendAddressPools: [
      {
        name: bePoolName
      }
    ]
    inboundNatPools: [
      {
        name: natPoolName
        properties: {
          frontendIPConfiguration: {
            id: frontEndIPConfigID
          }
          protocol: 'Tcp'
          frontendPortRangeStart: natStartPort
          frontendPortRangeEnd: natEndPort
          backendPort: natBackendPort
        }
      }
    ]
    loadBalancingRules: [
      {
        name: 'LBRule'
        properties: {
          frontendIPConfiguration: {
            id: frontEndIPConfigID
          }
          backendAddressPool: {
            id: lbPoolID
          }
          protocol: 'Tcp'
          frontendPort: 80
          backendPort: 80
          enableFloatingIP: false
          idleTimeoutInMinutes: 5
          probe: {
            id: lbProbeID
          }
        }
      }
    ]
    probes: [
      {
        name: 'tcpProbe'
        properties: {
          protocol: 'Tcp'
          port: 80
          intervalInSeconds: 5
          numberOfProbes: 2
```

```
        }
      }
    ]
  }
}

resource vmScaleSet 'Microsoft.Compute/virtualMachineScaleSets@2023-09-01' =
{
  name: vmScaleSetName
  location: location
  sku: {
    name: vmSku
    tier: 'Standard'
    capacity: instanceCount
  }
  properties: {
    overprovision: true
    upgradePolicy: {
      mode: 'Automatic'
    }
    singlePlacementGroup: singlePlacementGroup
    platformFaultDomainCount: platformFaultDomainCount
    virtualMachineProfile: {
      storageProfile: {
        osDisk: {
          caching: 'ReadWrite'
          createOption: 'FromImage'
        }
        imageReference: imageReference
      }
      osProfile: {
        computerNamePrefix: vmScaleSetName
        adminUsername: adminUsername
        adminPassword: adminPassword
      }
      securityProfile: ((securityType == 'TrustedLaunch') ?
securityProfileJson : null)
      networkProfile: {
        networkInterfaceConfigurations: [
          {
            name: nicName
            properties: {
              primary: true
              ipConfigurations: [
                {
                  name: ipConfigName
                  properties: {
                    subnet: {
                      id: vNet.properties.subnets[0].id
                    }
                    loadBalancerBackendAddressPools: [
                      {
                        id: lbPoolID
                      }
                    ]
```

```
                }
              }
            ]
          }
        }
      ]
    }
    extensionProfile: {
      extensions: [
        {
          name: 'Microsoft.Powershell.DSC'
          properties: {
            publisher: 'Microsoft.Powershell'
            type: 'DSC'
            typeHandlerVersion: '2.9'
            autoUpgradeMinorVersion: true
            forceUpdateTag: powershelldscUpdateTagVersion
            settings: {
              configuration: {
                url: powershelldscZipFullPath
                script: 'InstallIIS.ps1'
                function: 'InstallIIS'
              }
              configurationArguments: {
                nodeName: 'localhost'
                WebDeployPackagePath: webDeployPackageFullPath
              }
            }
          }
        }
      ]
    }
  }
}

resource publicIPAddress 'Microsoft.Network/publicIPAddresses@2023-04-01' =
{
  name: publicIPAddressName
  location: location
  properties: {
    publicIPAllocationMethod: 'Static'
    dnsSettings: {
      domainNameLabel: longvmScaleSet
    }
  }
}

resource vNet 'Microsoft.Network/virtualNetworks@2023-04-01' = {
  name: vNetName
  location: location
  properties: {
    addressSpace: {
      addressPrefixes: [
        addressPrefix
```

```
      ]
    }
    subnets: [
      {
        name: subnetName
        properties: {
          addressPrefix: subnetPrefix
        }
      }
    ]
  }
}

resource autoscalehost 'Microsoft.Insights/autoscalesettings@2022-10-01' = {
  name: 'autoscalehost'
  location: location
  properties: {
    name: 'autoscalehost'
    targetResourceUri: vmScaleSet.id
    enabled: true
    profiles: [
      {
        name: 'Profile1'
        capacity: {
          minimum: '1'
          maximum: '10'
          default: '1'
        }
        rules: [
          {
            metricTrigger: {
              metricName: 'Percentage CPU'
              metricResourceUri: vmScaleSet.id
              timeGrain: 'PT1M'
              statistic: 'Average'
              timeWindow: 'PT5M'
              timeAggregation: 'Average'
              operator: 'GreaterThan'
              threshold: 50
            }
            scaleAction: {
              direction: 'Increase'
              type: 'ChangeCount'
              value: '1'
              cooldown: 'PT5M'
            }
          }
          {
            metricTrigger: {
              metricName: 'Percentage CPU'
              metricResourceUri: vmScaleSet.id
              timeGrain: 'PT1M'
              statistic: 'Average'
              timeWindow: 'PT5M'
              timeAggregation: 'Average'
```

```
              operator: 'LessThan'
              threshold: 30
            }
            scaleAction: {
              direction: 'Decrease'
              type: 'ChangeCount'
              value: '1'
              cooldown: 'PT5M'
            }
          }
        ]
      }
    ]
  }
}

output applicationUrl string =
uri('http://${publicIPAddress.properties.dnsSettings.fqdn}', '/MyApp')
```

The following resources are defined in the Bicep file:

- Microsoft.Network/virtualNetworks
- Microsoft.Network/publicIPAddresses
- Microsoft.Network/loadBalancers
- Microsoft.Compute/virtualMachineScaleSets
- Microsoft.Insights/autoscaleSettings

Define a scale set

To create a Virtual Machine Scale Set with a Bicep file, you define the appropriate resources. The core parts of the Virtual Machine Scale Set resource type are:

Expand table

Property	Description of property	Example template value
type	Azure resource type to create	Microsoft.Compute/virtualMachineScaleSets
name	The scale set name	myScaleSet
location	The location to create the scale set	East US
sku.name	The VM size for each scale set	Standard_A1

Property	Description of property	Example template value
	instance	
sku.capacity	The number of VM instances to initially create	2
upgradePolicy.mode	VM instance upgrade mode when changes occur	Automatic
imageReference	The platform or custom image to use for the VM instances	Microsoft Windows Server 2016 Datacenter
osProfile.computerNamePrefix	The name prefix for each VM instance	myvmss
osProfile.adminUsername	The username for each VM instance	azureuser
osProfile.adminPassword	The password for each VM instance	P@ssw0rd!

To customize a Virtual Machine Scale Set Bicep file, you can change the VM size or initial capacity. Another option is to use a different platform or a custom image.

Add a sample application

To test your Virtual Machine Scale Set, install a basic web application. When you deploy a Virtual Machine Scale Set, VM extensions can provide post-deployment configuration and automation tasks, such as installing an app. Scripts can be downloaded from GitHub or provided to the Azure portal at extension run-time. To apply an extension to your Virtual Machine Scale Set, add the `extensionProfile` section to the resource example above. The extension profile typically defines the following properties:

- Extension type
- Extension publisher
- Extension version
- Location of configuration or install scripts
- Commands to execute on the VM instances

The Bicep file uses the PowerShell DSC extension to install an ASP.NET MVC app that runs in IIS.

An install script is downloaded from GitHub, as defined in `url`. The extension then runs `InstallIIS` from the `IISInstall.ps1` script, as defined in `function` and `Script`. The ASP.NET app itself is provided as a Web Deploy package, which is also downloaded from GitHub, as defined in `WebDeployPackagePath`:

Deploy the Bicep file

1. Save the Bicep file as `main.bicep` to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell.

CLI

Azure CLI

```
az group create --name exampleRG --location eastus
az deployment group create --resource-group exampleRG --template-
file main.bicep --parameters vmssName=<vmss-name>
```

Replace <*vmss-name*> with the name of the Virtual Machine Scale Set. It must be 3-61 characters in length and globally unique across Azure. You'll be prompted to enter `adminPassword`.

> **Note**
>
> When the deployment finishes, you should see a message indicating the deployment succeeded. It can take 10-15 minutes for the Virtual Machine Scale Set to be created and apply the extension to configure the app.

Validate the deployment

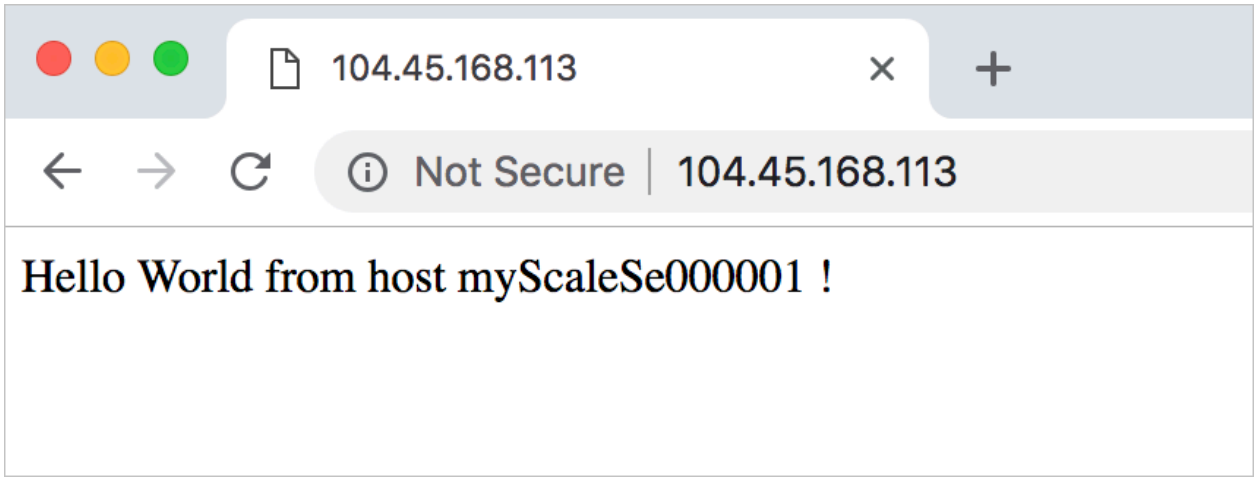
To see your Virtual Machine Scale Set in action, access the sample web application in a web browser. Obtain the public IP address of your load balancer using Azure CLI or Azure PowerShell.

CLI

Azure CLI

```
az network public-ip show --resource-group exampleRG
```

Enter the public IP address of the load balancer in to a web browser in the format *http://publicIpAddress/MyApp*. The load balancer distributes traffic to one of your VM instances, as shown in the following example:



Clean up resources

When no longer needed, use the Azure portal, Azure CLI, or Azure PowerShell to remove the resource group and its resources.

CLI

Azure CLI

```
az group delete --name exampleRG
```

Next steps

In this quickstart, you created a Windows Virtual Machine Scale Set with a Bicep file and used the PowerShell DSC extension to install a basic ASP.NET app on the VM instances. To learn more, continue to the tutorial for how to create and manage Azure Virtual Machine Scale Sets.

Create and manage Azure Virtual Machine Scale Sets

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machines in a scale set using an ARM template

Article • 08/22/2024

This article steps through using an ARM template to create a Virtual Machine Scale Set.

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template will open in the Azure portal.



Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

ARM template

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

ARM templates let you deploy groups of related resources. In a single template, you can create the Virtual Machine Scale Set, install applications, and configure autoscale rules. With the use of variables and parameters, this template can be reused to update existing, or create additional, scale sets. You can deploy templates through the Azure portal, Azure CLI, or Azure PowerShell, or from continuous integration / continuous delivery (CI/CD) pipelines.

Review the template

```
armasm
```

```
{
    "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
      "location": {
        "type": "string",
        "defaultValue": "[resourceGroup().location]",
        "metadata": {
          "description": "Location for all resources"
        }
      },
      "vmSku": {
        "type": "string",
        "defaultValue": "Standard_D2s_v3",
        "metadata": {
          "description": "Size of VMs in the VM Scale Set."
        }
      },
      "vmssName": {
        "type": "string",
        "metadata": {
          "description": "String used as a base for naming resources (9
characters or less). A hash is prepended to this string for some resources,
and resource-specific information is appended."
        }
      },
      "instanceCount": {
        "type": "int",
        "defaultValue": 1,
        "minValue": 1,
        "maxValue": 100,
        "metadata": {
          "description": "Number of VM instances (100 or less)."
        }
      },
      "adminUsername": {
        "type": "string",
        "metadata": {
          "description": "Admin username on all VMs."
        }
      },
      "authenticationType": {
        "type": "string",
        "defaultValue": "sshPublicKey",
        "allowedValues": [
          "sshPublicKey",
          "password"
        ],
        "metadata": {
          "description": "Type of authentication to use on the Virtual
Machine. SSH key is recommended."
        }
      },
```

```
      "adminPasswordOrKey": {
        "type": "securestring",
        "metadata": {
          "description": "SSH Key or password for the Virtual Machine. SSH
key is recommended."
        }
      },
      "securityType": {
          "type": "string",
          "defaultValue": "TrustedLaunch",
          "allowedValues": [
            "Standard",
            "TrustedLaunch"
          ],
          "metadata": {
            "description": "Security Type of the Virtual Machine."
          }
      },
      "_artifactsLocation": {
        "type": "string",
        "defaultValue": "[deployment().properties.templatelink.uri]",
        "metadata": {
          "description": "The base URI where artifacts required by this
template are located"
        }
      },
      "_artifactsLocationSasToken": {
        "type": "securestring",
        "defaultValue": "",
        "metadata": {
          "description": "The sasToken required to access
_artifactsLocation.  When the template is deployed using the accompanying
scripts, a sasToken will be automatically generated"
        }
      }
    },
    "variables": {
      "addressPrefix": "10.0.0.0/16",
      "subnetPrefix": "10.0.0.0/24",
      "networkApiVersion": "2020-11-01",
      "virtualNetworkName": "[concat(parameters('vmssName'), 'vnet')]",
      "publicIPAddressName": "[concat(parameters('vmssName'), 'pip')]",
      "subnetName": "[concat(parameters('vmssName'), 'subnet')]",
      "loadBalancerName": "[concat(parameters('vmssName'), 'lb')]",
      "publicIPAddressID": "
[resourceId('Microsoft.Network/publicIPAddresses',variables('publicIPAddress
Name'))]",
      "networkSecurityGroupName": "[concat(parameters('vmssName'), 'nsg')]",
      "bePoolName": "[concat(parameters('vmssName'), 'bepool')]",
      "lbRuleName": "[concat(parameters('vmssName'), 'lbrule')]",
      "lbProbeName": "[concat(parameters('vmssName'), 'lbprobe')]",
      "bePoolConfigID": "
[resourceId('Microsoft.Network/loadBalancers/backendAddressPools',
variables('loadBalancerName'),variables('bePoolName'))]",
      "lbProbeID": "[resourceId('Microsoft.Network/loadBalancers/probes',
```

```
variables('loadBalancerName'),variables('lbProbeName'))]",
      "nicName": "[concat(parameters('vmssName'), 'nic')]",
      "ipConfigName": "[concat(parameters('vmssName'), 'ipconfig')]",
      "frontEndIPConfigID": "
[resourceId('Microsoft.Network/loadBalancers/frontendIPConfigurations',
variables('loadBalancerName'),'loadBalancerFrontEnd')]",
      "osType": {
          "publisher": "Canonical",
          "offer": "0001-com-ubuntu-server-focal",
          "sku": "20_04-lts-gen2",
          "version": "latest"
      },
      "imageReference": "[variables('osType')]",
      "securityProfileJson": {
          "uefiSettings": {
            "secureBootEnabled": true,
            "vTpmEnabled": true
          },
          "securityType": "[parameters('securityType')]"
      },
      "linuxConfiguration": {
        "disablePasswordAuthentication": true,
        "ssh": {
          "publicKeys": [
            {
              "path": "[concat('/home/', parameters('adminUsername'),
'/.ssh/authorized_keys')]",
              "keyData": "[parameters('adminPasswordOrKey')]"
            }
          ]
        }
      }
    },
    "resources": [
              {
            "type": "Microsoft.Network/networkSecurityGroups",
            "apiVersion": "2023-04-01",
            "name": "[variables('networkSecurityGroupName')]",
            "location": "[parameters('location')]",
            "properties": {
                "securityRules": [
                    {
                        "name": "AllowPort9000",
                        "properties": {
                            "protocol": "*",
                            "sourcePortRange": "*",
                            "destinationPortRange": "9000",
                            "sourceAddressPrefix": "Internet",
                            "destinationAddressPrefix": "*",
                            "access": "Allow",
                            "priority": 100,
                            "direction": "Inbound"
                        }
                    }
                ]
```

```
            }
          },
        {
          "type": "Microsoft.Network/virtualNetworks",
          "apiVersion": "2023-04-01",
          "name": "[variables('virtualNetworkName')]",
          "location": "[parameters('location')]",
          "properties": {
            "addressSpace": {
              "addressPrefixes": [
                "[variables('addressPrefix')]"
              ]
            },
            "subnets": [
              {
                "name": "[variables('subnetName')]",
                "properties": {
                  "addressPrefix": "[variables('subnetPrefix')]",
                  "networkSecurityGroup": {
                    "id": "
[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
                  }
                }
              }
            ]
          }
        },
        {
          "type": "Microsoft.Network/publicIPAddresses",
          "apiVersion": "2023-04-01",
          "name": "[variables('publicIPAddressName')]",
          "location": "[parameters('location')]",
          "sku": {
            "name": "Standard"
          },
          "properties": {

            "publicIPAllocationMethod": "Static",
            "dnsSettings": {
              "domainNameLabel": "[parameters('vmssName')]"
            }
          }
        },
        {
          "type": "Microsoft.Network/loadBalancers",
          "apiVersion": "2023-04-01",
          "name": "[variables('loadBalancerName')]",
          "location": "[parameters('location')]",
          "sku": {
            "name": "Standard"
          },
          "dependsOn": [
            "[resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName'))]"
```

```
      ],
      "properties": {
        "frontendIPConfigurations": [
          {
            "name": "LoadBalancerFrontEnd",
            "properties": {
              "publicIPAddress": {
                "id": "[variables('publicIPAddressID')]"
              }
            }
          }
        ],
        "backendAddressPools": [
          {
            "name": "[variables('bePoolName')]"
          }
        ],
        "probes": [
          {
            "name": "[variables('lbProbeName')]",
            "properties": {
              "port": 9000,
              "protocol": "Tcp",
              "numberOfProbes": 2,
              "intervalInSeconds": 5
            }
          }
        ],
        "loadBalancingRules": [
          {
            "name": "[variables('lbRuleName')]",
            "properties": {
              "frontendIPConfiguration": {
                "id": "[variables('frontEndIPConfigID')]"
              },
              "backendAddressPool": {
                "id": "[variables('bePoolConfigID')]"
              },
              "probe": {
                "id": "[variables('lbProbeID')]"
              },
              "loadDistribution": "Default",
              "backendPort": 9000,
              "frontendPort": 9000,
              "protocol": "Tcp",
              "idleTimeoutInMinutes": 4,
              "enableFloatingIP": false,
              "enableTcpReset": false,
              "disableOutboundSnat": false
            }
          }
        ]
      }
    },
    {
```

```
        "type": "Microsoft.Compute/virtualMachineScaleSets",
        "apiVersion": "2023-09-01",
        "name": "[parameters('vmssName')]",
        "location": "[parameters('location')]",
        "sku": {
          "name": "[parameters('vmSku')]",
          "tier": "Standard",
          "capacity": "[parameters('instanceCount')]"
        },
        "dependsOn": [
          "[resourceId('Microsoft.Network/loadBalancers',
variables('loadBalancerName'))]",
          "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]"
        ],
        "properties": {
          "orchestrationMode": "Flexible",
          "platformFaultDomainCount": 1,
          "singlePlacementGroup": false,
          "virtualMachineProfile": {
            "storageProfile": {
              "osDisk": {
                "createOption": "FromImage",
                "caching": "ReadWrite"
              },
              "imageReference": "[variables('imageReference')]"
            },
            "osProfile": {
              "computerNamePrefix": "[parameters('vmssName')]",
              "adminUsername": "[parameters('adminUsername')]",
              "adminPassword": "[parameters('adminPasswordOrKey')]",
              "linuxConfiguration": "
[if(equals(parameters('authenticationType'), 'password'), null(),
variables('linuxConfiguration'))]"
            },
            "securityProfile": "[if(equals(parameters('securityType'),
'TrustedLaunch'), variables('securityProfileJson'), null())]",
            "networkProfile": {
              "networkApiVersion": "[variables('networkApiVersion')]",
              "networkInterfaceConfigurations": [
                {
                  "name": "[variables('nicName')]",
                  "properties": {
                    "primary": true,
                    "ipConfigurations": [
                      {
                        "name": "[variables('ipConfigName')]",
                        "properties": {
                          "primary": true,
                          "subnet": {
                            "id": "
[resourceId('Microsoft.Network/virtualNetworks/subnets',
variables('virtualNetworkName'), variables('subnetName'))]"
                          },
                          "loadBalancerBackendAddressPools": [
```

```
                            {
                              "id": "
[resourceId('Microsoft.Network/loadBalancers/backendAddressPools',
variables('loadBalancerName'), variables('bePoolName'))]"
                            }
                          ]
                        }
                      }
                    ]
                  }
                }
              ]
            },
            "extensionProfile": {
              "extensions": [
                {
                  "name": "lapextension",
                  "properties": {
                    "publisher": "Microsoft.Azure.Extensions",
                    "type": "CustomScript",
                    "typeHandlerVersion": "2.0",
                    "autoUpgradeMinorVersion": true,
                    "enableAutomaticUpgrade": true,
                    "settings": {
                      "fileUris": [
                        "[uri(parameters('_artifactsLocation'),
concat('installserver.sh', parameters('_artifactsLocationSasToken')))]",
                        "[uri(parameters('_artifactsLocation'),
concat('workserver.py', parameters('_artifactsLocationSasToken')))]"
                      ],
                      "commandToExecute": "bash installserver.sh"
                    }
                  }
                }
              ]
            }
          }
        },
        {
          "type": "Microsoft.Insights/autoscaleSettings",
          "apiVersion": "2022-10-01",
          "name": "[concat(parameters('vmssName'), '-autoscalehost')]",
          "location": "[parameters('location')]",
          "dependsOn": [
            "[resourceId('Microsoft.Compute/virtualMachineScaleSets',
parameters('vmSSName'))]"
          ],
          "properties": {
            "name": "[concat(parameters('vmssName'), '-autoscalehost')]",
            "targetResourceUri": "
[resourceId('Microsoft.Compute/virtualMachineScaleSets',
parameters('vmSSName'))]",
            "enabled": true,
            "profiles": [
```

```
            {
              "name": "Profile1",
              "capacity": {
                "minimum": "1",
                "maximum": "10",
                "default": "1"
              },
              "rules": [
                {
                  "metricTrigger": {
                    "metricName": "Percentage CPU",
                    "metricResourceUri": "
[resourceId('Microsoft.Compute/virtualMachineScaleSets',
parameters('vmSSName'))]",
                    "timeGrain": "PT1M",
                    "statistic": "Average",
                    "timeWindow": "PT5M",
                    "timeAggregation": "Average",
                    "operator": "GreaterThan",
                    "threshold": 60
                  },
                  "scaleAction": {
                    "direction": "Increase",
                    "type": "ChangeCount",
                    "value": "1",
                    "cooldown": "PT1M"
                  }
                },
                {
                  "metricTrigger": {
                    "metricName": "Percentage CPU",
                    "metricResourceUri": "
[resourceId('Microsoft.Compute/virtualMachineScaleSets',
parameters('vmSSName'))]",
                    "timeGrain": "PT1M",
                    "statistic": "Average",
                    "timeWindow": "PT5M",
                    "timeAggregation": "Average",
                    "operator": "LessThan",
                    "threshold": 30
                  },
                  "scaleAction": {
                    "direction": "Decrease",
                    "type": "ChangeCount",
                    "value": "1",
                    "cooldown": "PT1M"
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
```

These resources are defined in the template:

- Microsoft.Network/virtualNetworks
- Microsoft.Network/publicIPAddresses
- Microsoft.Network/loadBalancers
- Microsoft.Compute/virtualMachineScaleSets
- Microsoft.Insights/autoscaleSettings
- Microsoft.Network/networkSecurityGroups

Define a scale set

To create a scale with a template, you define the appropriate resources. The core parts of the Virtual Machine Scale Set resource type are:

Expand table

Property	Description of property	Example template value
type	Azure resource type to create	Microsoft.Compute/virtualMachineScaleSets
name	The scale set name	myScaleSet
location	The location to create the scale set	East US
sku.name	The VM size for each scale set instance	Standard_A1
sku.capacity	The number of VM instances to initially create	2
imageReference	The platform or custom image to use for the VM instances	Canonical Ubuntu Server 16.04-LTS
osProfile.computerNamePrefix	The name prefix for each VM instance	myvmss

Property	Description of property	Example template value
osProfile.adminUsername	The username for each VM instance	azureuser
osProfile.adminPassword	The password for each VM instance	P@ssw0rd!

To customize a scale set template, you can change the VM size or initial capacity. Another option is to use a different platform or a custom image.

Add a sample application

To test your scale set, install a basic web application. When you deploy a scale set, VM extensions can provide post-deployment configuration and automation tasks, such as installing an app. Scripts can be downloaded from Azure storage or GitHub, or provided to the Azure portal at extension run-time. To apply an extension to your scale set, you add the *extensionProfile* section to the preceding resource example. The extension profile typically defines the following properties:

- Extension type
- Extension publisher
- Extension version
- Location of configuration or install scripts
- Commands to execute on the VM instances

The template uses the Custom Script Extension to install Bottle, a Python web framework, and a simple HTTP server.

Two scripts are defined in **fileUris** - *installserver.sh*, and *workserver.py*. These files are downloaded from GitHub, then *commandToExecute* runs `bash installserver.sh` to install and configure the app.

Deploy the template

You can also deploy a Resource Manager template by using Azure CLI:

Azure CLI

```
# Create a resource group
az group create --name myResourceGroup --location EastUS

# Deploy template into resource group
```

```
az deployment group create -g myResourceGroup -f azuredeploy.json --
parameters _artifactsLocation=https://raw.githubusercontent.com/Azure/azure-
quickstart-templates/master/application-workloads/python/vmss-bottle-
autoscale/azuredeploy.json
```

Answer the prompts to provide a scale set name, instance count, and admin credentials for the VM instances. It takes a few minutes for the scale set and supporting resources to be created.

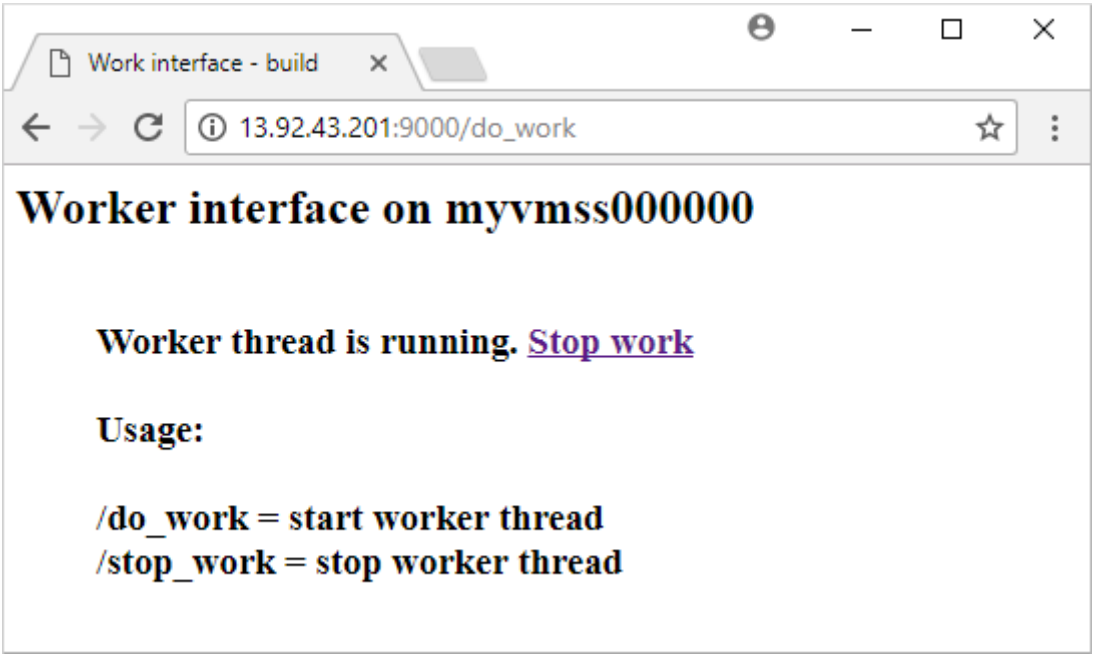
Validate the deployment

To see your scale set in action, access the sample web application in a web browser. Obtain the public IP address of the load balancer with az network public-ip list as follows:

Azure CLI

```
az network public-ip list \
    --resource-group myResourceGroup \
    --query [*].ipAddress -o tsv
```

Enter the public IP address of the load balancer in to a web browser in the format *http://publicIpAddress:9000/do_work*. The load balancer distributes traffic to one of your VM instances, as shown in the following example:



Clean up resources

When no longer needed, you can use az group delete to remove the resource group, scale set, and all related resources as follows. The `--no-wait` parameter returns control to the prompt without waiting for the operation to complete. The `--yes` parameter confirms that you wish to delete the resources without an additional prompt to do so.

Azure CLI

```
az group delete --name myResourceGroup --yes --no-wait
```

Next steps

Learn how to create a Flexible scale set in the Azure portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Create and Manage Linux VMs with the Azure CLI

Article • 04/06/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure virtual machines provide a fully configurable and flexible computing environment. This tutorial covers basic Azure virtual machine deployment items such as selecting a VM size, selecting a VM image, and deploying a VM. You learn how to:

- ✔ Create and connect to a VM
- ✔ Select and use VM images
- ✔ View and use specific VM sizes
- ✔ Resize a VM
- ✔ View and understand VM state

This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version.

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Create resource group

Below, we declare environment variables. A random suffix is appended to resource names that need to be unique for each deployment.

Bash

```
export RANDOM_SUFFIX=$(openssl rand -hex 3)
export REGION="eastus2"
export MY_RESOURCE_GROUP_NAME="myResourceGroupVM$RANDOM_SUFFIX"
az group create --name $MY_RESOURCE_GROUP_NAME --location $REGION
```

Results:

JSON

```
{
  "id": "/subscriptions/xxxxx-xxxxx-xxxxx-
xxxxx/resourceGroups/myResourceGroupVMxxx",
  "location": "eastus2",
  "name": "myResourceGroupVMxxx",
```

```
  "properties": {
    "provisioningState": "Succeeded"
  }
}
```

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine. In this example, a resource group named *myResourceGroupVM* is created in the *eastus2* region.

The resource group is specified when creating or modifying a VM, which can be seen throughout this tutorial.

Create virtual machine

When you create a virtual machine, several options are available such as operating system image, disk sizing, and administrative credentials. The following example creates a VM named *myVM* that runs SUSE Linux Enterprise Server (SLES). A user account named *azureuser* is created on the VM, and SSH keys are generated if they do not exist in the default key location (*~/.ssh*).

Bash

```
export MY_VM_NAME="myVM$RANDOM_SUFFIX"
az vm create \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --name $MY_VM_NAME \
    --image SuseSles15SP5 \
    --public-ip-sku Standard \
    --admin-username azureuser \
    --generate-ssh-keys
```

It may take a few minutes to create the VM. Once the VM has been created, the Azure CLI outputs information about the VM. Take note of the `publicIpAddress`; this address can be used to access the virtual machine.

JSON

```
{
  "fqdns": "",
  "id": "/subscriptions/xxxxx-xxxxx-xxxxx-
xxxxx/resourceGroups/myResourceGroupVMxxx/providers/Microsoft.Compute/virtualMachi
nes/myVMxxx",
  "location": "eastus2",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
```

```
  "publicIpAddress": "52.174.34.95",
  "resourceGroup": "myResourceGroupVMxxx"
}
```

Connect to VM

You can now connect to the VM with SSH in the Azure Cloud Shell or from your local computer. Replace the example IP address with the `publicIpAddress` noted in the previous step.

To connect to the VM, first retrieve the public IP address using the Azure CLI. Execute the following command to store the IP address in a variable: `export IP_ADDRESS=$(az vm show --show-details --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME --query publicIps --output tsv)`

Once you have the IP address, use SSH to connect to the VM. The following command connects to the VM using the `azureuser` account and the retrieved IP address: `ssh -o StrictHostKeyChecking=no azureuser@$IP_ADDRESS`

Understand VM images

The Azure Marketplace includes many images that can be used to create VMs. In the previous steps, a virtual machine was created using a SUSE image. In this step, the Azure CLI is used to search the marketplace for an Ubuntu image, which is then used to deploy a second virtual machine.

To see a list of the most commonly used images, use the az vm image list command.

Bash

```
az vm image list --output table
```

The command output returns the most popular VM images on Azure.

Output

```
Architecture    Offer                         Publisher              Sku
Urn
UrnAlias                Version
--------------  ----------------------------  ---------------------  ------------
----------------------  ------------------------------------------------------------
--------------------  ----------------------  ---------
x64             debian-10                     Debian                 10
```

```
Debian:debian-10:10:latest
Debian                  latest
x64             flatcar-container-linux-free  kinvolk                 stable
kinvolk:flatcar-container-linux-free:stable:latest
Flatcar                 latest
x64             opensuse-leap-15-3            SUSE                    gen2
SUSE:opensuse-leap-15-3:gen2:latest
openSUSE-Leap           latest
x64             RHEL                          RedHat                  7-LVM
RedHat:RHEL:7-LVM:latest
RHEL                    latest
x64             sles-15-sp3                   SUSE                    gen2
SUSE:sles-15-sp3:gen2:latest
SLES                    latest
x64             0001-com-ubuntu-server-jammy  Canonical               18.04-LTS
Canonical:UbuntuServer:18.04-LTS:latest
UbuntuLTS               latest
x64             WindowsServer                 MicrosoftWindowsServer  2022-
Datacenter                      MicrosoftWindowsServer:WindowsServer:2022-
Datacenter:latest                      Win2022Datacenter        latest
x64             WindowsServer                 MicrosoftWindowsServer  2022-
datacenter-azure-edition-core  MicrosoftWindowsServer:WindowsServer:2022-
datacenter-azure-edition-core:latest  Win2022AzureEditionCore  latest
x64             WindowsServer                 MicrosoftWindowsServer  2019-
Datacenter                      MicrosoftWindowsServer:WindowsServer:2019-
Datacenter:latest                      Win2019Datacenter        latest
x64             WindowsServer                 MicrosoftWindowsServer  2016-
Datacenter                      MicrosoftWindowsServer:WindowsServer:2016-
Datacenter:latest                      Win2016Datacenter        latest
x64             WindowsServer                 MicrosoftWindowsServer  2012-R2-
Datacenter                   MicrosoftWindowsServer:WindowsServer:2012-R2-
Datacenter:latest                   Win2012R2Datacenter      latest
x64             WindowsServer                 MicrosoftWindowsServer  2012-
Datacenter                      MicrosoftWindowsServer:WindowsServer:2012-
Datacenter:latest                      Win2012Datacenter        latest
x64             WindowsServer                 MicrosoftWindowsServer  2008-R2-SP1
MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest
Win2008R2SP1            latest
```

A full list can be seen by adding the `--all` parameter. The image list can also be filtered by `--publisher` or `--offer`. In this example, the list is filtered for all images, published by OpenLogic, with an offer that matches *0001-com-ubuntu-server-jammy*.

Bash

```
az vm image list --offer 0001-com-ubuntu-server-jammy --publisher Canonical --all
--output table
```

Example partial output:

Output

```
Architecture    Offer                              Publisher    Sku
Urn                                                                    Version
--------------  ---------------------------------  -----------  ---------------  -
-----------------------------------------------------------------------  ---------
------
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202204200
22.04.202204200
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202205060
22.04.202205060
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202205280
22.04.202205280
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202206040
22.04.202206040
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202206090
22.04.202206090
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202206160
22.04.202206160
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202206220
22.04.202206220
x64             0001-com-ubuntu-server-jammy       Canonical    22_04-lts
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:22.04.202207060
22.04.202207060
```

> **Note**
>
> Canonical has changed the **Offer** names they use for the most recent versions. Before Ubuntu 20.04, the **Offer** name is UbuntuServer. For Ubuntu 20.04 the **Offer** name is `0001-com-ubuntu-server-focal` and for Ubuntu 22.04 it's `0001-com-ubuntu-server-jammy`.

To deploy a VM using a specific image, take note of the value in the *Urn* column, which consists of the publisher, offer, SKU, and optionally a version number to identify the image. When specifying the image, the image version number can be replaced with `latest`, which selects the latest version of the distribution. In this example, the `--image` parameter is used to specify the latest version of a Ubuntu 22.04.

Bash

```bash
export MY_VM2_NAME="myVM2$RANDOM_SUFFIX"
az vm create --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM2_NAME --image
Canonical:0001-com-ubuntu-server-jammy:22_04-lts:latest --generate-ssh-keys
```

Understand VM sizes

A virtual machine size determines the amount of compute resources such as CPU, GPU, and memory that are made available to the virtual machine. Virtual machines need to be sized appropriately for the expected work load. If workload increases, an existing virtual machine can be resized.

VM Sizes

The following table categorizes sizes into use cases.

Expand table

Type	Description
General purpose	Balanced CPU-to-memory. Ideal for dev / test and small to medium applications and data solutions.
Compute optimized	High CPU-to-memory. Good for medium traffic applications, network appliances, and batch processes.
Memory optimized	High memory-to-core. Great for relational databases, medium to large caches, and in-memory analytics.
Storage optimized	High disk throughput and IO. Ideal for Big Data, SQL, and NoSQL databases.
GPU	Specialized VMs targeted for heavy graphic rendering and video editing.
High performance	Our most powerful CPU VMs with optional high-throughput network interfaces (RDMA).

Find available VM sizes

To see a list of VM sizes available in a particular region, use the az vm list-sizes command.

Bash

```
az vm list-sizes --location $REGION --output table
```

Example partial output:

Output

```
  MaxDataDiskCount    MemoryInMb  Name                        NumberOfCores
OsDiskSizeInMb    ResourceDiskSizeInMb
------------------  ------------  --------------------  --------------  -------
```

```
---------  ----------------------
4                   8192          Standard_D2ds_v4           2
1047552           76800
8                   16384         Standard_D4ds_v4           4
1047552           153600
16                  32768         Standard_D8ds_v4           8
1047552           307200
32                  65536         Standard_D16ds_v4          16
1047552           614400
32                  131072        Standard_D32ds_v4          32
1047552           1228800
32                  196608        Standard_D48ds_v4          48
1047552           1843200
32                  262144        Standard_D64ds_v4          64
1047552           2457600
4                   8192          Standard_D2ds_v5           2
1047552           76800
8                   16384         Standard_D4ds_v5           4
1047552           153600
16                  32768         Standard_D8ds_v5           8
1047552           307200
32                  65536         Standard_D16ds_v5          16
1047552           614400
32                  131072        Standard_D32ds_v5          32
1047552           1228800
32                  196608        Standard_D48ds_v5          48
1047552           1843200
32                  262144        Standard_D64ds_v5          64
1047552           2457600
32                  393216        Standard_D96ds_v5          96
1047552           3686400
```

Create VM with specific size

In the previous VM creation example, a size was not provided, which results in a default size. A VM size can be selected at creation time using az vm create and the `--size` parameter.

Bash

```
export MY_VM3_NAME="myVM3$RANDOM_SUFFIX"
az vm create \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --name $MY_VM3_NAME \
    --image SuseSles15SP5 \
    --size Standard_D2ds_v4  \
    --generate-ssh-keys
```

Resize a VM

After a VM has been deployed, it can be resized to increase or decrease resource allocation. You can view the current size of a VM with az vm show:

```
Bash

az vm show --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME --query
hardwareProfile.vmSize
```

Before resizing a VM, check if the desired size is available on the current Azure cluster. The az vm list-vm-resize-options command returns the list of sizes.

```
Bash

az vm list-vm-resize-options --resource-group $MY_RESOURCE_GROUP_NAME --name
$MY_VM_NAME --query [].name
```

If the desired size is available, the VM can be resized from a powered-on state, although it will be rebooted during the operation. Use the az vm resize command to perform the resize.

```
Bash

az vm resize --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME --size
Standard_D4s_v3
```

If the desired size is not available on the current cluster, the VM needs to be deallocated before the resize operation can occur. Use the az vm deallocate command to stop and deallocate the VM. Note that when the VM is powered back on, any data on the temporary disk may be removed. The public IP address also changes unless a static IP address is being used. Once deallocated, the resize can occur.

After the resize, the VM can be started.

```
Bash

az vm start --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME
```

VM power states

An Azure VM can have one of many power states. This state represents the current state of the VM from the standpoint of the hypervisor.

Power states

Expand table

Power State	Description
Starting	Indicates the virtual machine is being started.
Running	Indicates that the virtual machine is running.
Stopping	Indicates that the virtual machine is being stopped.
Stopped	Indicates that the virtual machine is stopped. Virtual machines in the stopped state still incur compute charges.
Deallocating	Indicates that the virtual machine is being deallocated.
Deallocated	Indicates that the virtual machine is removed from the hypervisor but still available in the control plane. Virtual machines in the Deallocated state do not incur compute charges.
-	Indicates that the power state of the virtual machine is unknown.

Find the power state

To retrieve the state of a particular VM, use the az vm get-instance-view command. Be sure to specify a valid name for a virtual machine and resource group.

Bash

```
az vm get-instance-view \
    --name $MY_VM_NAME \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --query instanceView.statuses[1] --output table
```

Output:

Output

```
Code                Level    DisplayStatus
------------------  -------  ---------------
PowerState/running  Info     VM running
```

To retrieve the power state of all the VMs in your subscription, use the Virtual Machines - List All API with parameter **statusOnly** set to *true*.

Management tasks

During the life-cycle of a virtual machine, you may want to run management tasks such as starting, stopping, or deleting a virtual machine. Additionally, you may want to create scripts to automate repetitive or complex tasks. Using the Azure CLI, many common management tasks can be run from the command line or in scripts.

Get IP address

This command returns the private and public IP addresses of a virtual machine.

```Bash
az vm list-ip-addresses --resource-group $MY_RESOURCE_GROUP_NAME --name
$MY_VM_NAME --output table
```

Stop virtual machine

```Bash
az vm stop --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME
```

Start virtual machine

```Bash
az vm start --resource-group $MY_RESOURCE_GROUP_NAME --name $MY_VM_NAME
```

Deleting VM resources

Depending on how you delete a VM, it may only delete the VM resource, not the networking and disk resources. You can change the default behavior to delete other resources when you delete the VM. For more information, see Delete a VM and attached resources.

Deleting a resource group also deletes all resources in the resource group, like the VM, virtual network, and disk. The `--no-wait` parameter returns control to the prompt without waiting for the operation to complete. The `--yes` parameter confirms that you wish to delete the resources without an additional prompt to do so.

Next steps

In this tutorial, you learned about basic VM creation and management such as how to:

- ✓ Create and connect to a VM
- ✓ Select and use VM images
- ✓ View and use specific VM sizes
- ✓ Resize a VM
- ✓ View and understand VM state

Advance to the next tutorial to learn about VM disks.

Create and Manage VM disks

Tutorial - Manage Azure disks with the Azure CLI

Article • 12/06/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure virtual machines (VMs) use disks to store the operating system, applications, and data. When you create a VM, it is important to choose a disk size and configuration appropriate to the expected workload. This tutorial shows you how to deploy and manage VM disks. You learn about:

- ✓ OS disks and temporary disks
- ✓ Data disks
- ✓ Standard and Premium disks
- ✓ Disk performance
- ✓ Attaching and preparing data disks
- ✓ Disk snapshots

Default Azure disks

When an Azure virtual machine is created, two disks are automatically attached to the virtual machine.

Operating system disk - Operating system disks can be sized up to 2 TB, and hosts the VMs operating system. The OS disk is labeled */dev/sda* by default. The disk caching configuration of the OS disk is optimized for OS performance. Because of this configuration, the OS disk **should not** be used for applications or data. For applications and data, use data disks, which are detailed later in this tutorial.

Temporary disk - Temporary disks use a solid-state drive that is located on the same Azure host as the VM. Temp disks are highly performant and may be used for operations such as temporary data processing. However, if the VM is moved to a new host, any data stored on a temporary disk is removed. The size of the temporary disk is determined by the VM size. Temporary disks are labeled */dev/sdb* and have a mountpoint of */mnt*.

Azure data disks

To install applications and store data, additional data disks can be added. Data disks should be used in any situation where durable and responsive data storage is desired. The size of the virtual machine determines how many data disks can be attached to a VM.

VM disk types

Azure provides two types of disks.

Standard disks - backed by HDDs, and delivers cost-effective storage while still being performant. Standard disks are ideal for a cost effective dev and test workload.

Premium disks - backed by SSD-based, high-performance, low-latency disk. Perfect for VMs running production workload. VM sizes with an **S** in the size name, typically support Premium Storage. For example, DS-series, DSv2-series, GS-series, and FS-series VMs support premium storage. When you select a disk size, the value is rounded up to the next type. For example, if the disk size is more than 64 GB, but less than 128 GB, the disk type is P10.

Expand table

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base provisioned	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
IOPS per disk														
**Expanded provisioned IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8,000	16,000	20,000	20,000	20,000	20,000
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year

*Applies only to disks with on-demand bursting enabled.
** Only applies to disks with performance plus (preview) enabled.

When you provision a premium storage disk, unlike standard storage, you are guaranteed the capacity, IOPS, and throughput of that disk. For example, if you create a P50 disk, Azure provisions 4,095-GB storage capacity, 7,500 IOPS, and 250-MB/s throughput for that disk. Your application can use all or part of the capacity and performance. Premium SSD disks are designed to provide low single-digit millisecond latencies and target IOPS and throughput described in the preceding table 99.9% of the time.

While the above table identifies max IOPS per disk, a higher level of performance can be achieved by striping multiple data disks. For instance, 64 data disks can be attached to Standard_GS5 VM. If each of these disks is sized as a P30, a maximum of 80,000 IOPS can be achieved. For detailed information on max IOPS per VM, see VM types and sizes.

Launch Azure Cloud Shell

Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open Cloud Shell, select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create and attach disks

Data disks can be created and attached at VM creation time or to an existing VM.

Attach disk at VM creation

Create a resource group with the az group create command.

Azure CLI

```
az group create --name myResourceGroupDisk --location eastus
```

Create a VM using the az vm create command. The following example creates a VM named *myVM*, adds a user account named *azureuser*, and generates SSH keys if they do not exist. The `--datadisk-sizes-gb` argument is used to specify that an additional disk should be created and attached to the virtual machine. To create and attach more than one disk, use a space-delimited list of disk size values. In the following example, a VM is created with two data disks, both 128 GB. Because the disk sizes are 128 GB, these disks are both configured as P10s, which provide maximum 500 IOPS per disk.

Azure CLI

```
az vm create \
  --resource-group myResourceGroupDisk \
  --name myVM \
  --image Ubuntu2204 \
  --size Standard_DS2_v2 \
  --admin-username azureuser \
  --generate-ssh-keys \
  --data-disk-sizes-gb 128 128
```

Attach disk to existing VM

To create and attach a new disk to an existing virtual machine, use the az vm disk attach command. The following example creates a premium disk, 128 gigabytes in size, and attaches it to the VM created in the last step.

Azure CLI

```
az vm disk attach \
    --resource-group myResourceGroupDisk \
    --vm-name myVM \
    --name myDataDisk \
    --size-gb 128 \
    --sku Premium_LRS \
    --new
```

Prepare data disks

Once a disk has been attached to the virtual machine, the operating system needs to be configured to use the disk. The following example shows how to manually configure a disk. This process can also be automated using cloud-init, which is covered in a later tutorial.

Create an SSH connection with the virtual machine. Replace the example IP address with the public IP of the virtual machine.

Console

```
ssh azureuser@10.101.10.10
```

Partition the disk with `parted`.

Bash

```
sudo parted /dev/sdc --script mklabel gpt mkpart xfspart xfs 0% 100%
```

Write a file system to the partition by using the `mkfs` command. Use `partprobe` to make the OS aware of the change.

Bash

```
sudo mkfs.xfs /dev/sdc1
sudo partprobe /dev/sdc1
```

Mount the new disk so that it is accessible in the operating system.

```
```

```Bash
sudo mkdir /datadrive && sudo mount /dev/sdc1 /datadrive
```

The disk can now be accessed through the `/datadrive` mountpoint, which can be verified by running the `df -h` command.

```Bash
df -h | grep -i "sd"
```

The output shows the new drive mounted on `/datadrive`.

```Bash
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        29G  2.0G   27G   7% /
/dev/sda15      105M  3.6M  101M   4% /boot/efi
/dev/sdb1        14G   41M   13G   1% /mnt
/dev/sdc1        50G   52M   47G   1% /datadrive
```

To ensure that the drive is remounted after a reboot, it must be added to the */etc/fstab* file. To do so, get the UUID of the disk with the `blkid` utility.

```Bash
sudo -i blkid
```

The output displays the UUID of the drive, `/dev/sdc1` in this case.

```Bash
/dev/sdc1: UUID="33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e" TYPE="xfs"
```

> ⓘ **Note**
>
> Improperly editing the **/etc/fstab** file could result in an unbootable system. If unsure, refer to the distribution's documentation for information on how to properly edit this file. It is also recommended that a backup of the /etc/fstab file is created before editing.

Open the `/etc/fstab` file in a text editor as follows:

```Bash
sudo nano /etc/fstab
```

Add a line similar to the following to the */etc/fstab* file, replacing the UUID value with your own.

```Bash
UUID=33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e   /datadrive  xfs    defaults,nofail   1  2
```

When you are done editing the file, use `Ctrl+O` to write the file and `Ctrl+X` to exit the editor.

Now that the disk has been configured, close the SSH session.

```Bash
exit
```

Take a disk snapshot

When you take a disk snapshot, Azure creates a read only, point-in-time copy of the disk. Azure VM snapshots are useful to quickly save the state of a VM before you make configuration changes. In the event of an issue or error, VM can be restored using a snapshot. When a VM has more than one disk, a snapshot is taken of each disk independently of the others. To take application consistent backups, consider stopping the VM before you take disk snapshots. Alternatively, use the Azure Backup service, which enables you to perform automated backups while the VM is running.

Create snapshot

Before you create a snapshot, you need the ID or name of the disk. Use az vm show to show the disk ID. In this example, the disk ID is stored in a variable so that it can be used in a later step.

Azure CLI

```
osdiskid=$(az vm show \
   -g myResourceGroupDisk \
   -n myVM \
   --query "storageProfile.osDisk.managedDisk.id" \
   -o tsv)
```

Now that you have the ID, use az snapshot create to create a snapshot of the disk.

Azure CLI

```
az snapshot create \
    --resource-group myResourceGroupDisk \
    --source "$osdiskid" \
    --name osDisk-backup
```

Create disk from snapshot

This snapshot can then be converted into a disk using az disk create, which can be used to recreate the virtual machine.

Azure CLI

```
az disk create \
   --resource-group myResourceGroupDisk \
   --name mySnapshotDisk \
   --source osDisk-backup
```

Restore virtual machine from snapshot

To demonstrate virtual machine recovery, delete the existing virtual machine using az vm delete.

Azure CLI

```
az vm delete \
--resource-group myResourceGroupDisk \
--name myVM
```

Create a new virtual machine from the snapshot disk.

Azure CLI

```
az vm create \
    --resource-group myResourceGroupDisk \
    --name myVM \
    --attach-os-disk mySnapshotDisk \
    --os-type linux
```

Reattach data disk

All data disks need to be reattached to the virtual machine.

Find the data disk name using the az disk list command. This example places the name of the disk in a variable named `datadisk`, which is used in the next step.

Azure CLI

```
datadisk=$(az disk list \
   -g myResourceGroupDisk \
   --query "[?contains(name,'myVM')].[id]" \
   -o tsv)
```

Use the az vm disk attach command to attach the disk.

Azure CLI

```
az vm disk attach \
   –g myResourceGroupDisk \
   --vm-name myVM \
   --name $datadisk
```

Next steps

In this tutorial, you learned about VM disks topics such as:

- ✓ OS disks and temporary disks
- ✓ Data disks
- ✓ Standard and Premium disks
- ✓ Disk performance
- ✓ Attaching and preparing data disks
- ✓ Disk snapshots

Advance to the next tutorial to learn about automating VM configuration.

Automate VM configuration

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Create a custom image of an Azure VM with the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap configurations such as preloading applications, application configurations, and other OS configurations. In this tutorial, you create your own custom image of an Azure virtual machine. You learn how to:

- ✔ Create an Azure Compute Gallery (formerly known as Shared Image Gallery)
- ✔ Create an image definition
- ✔ Create an image version
- ✔ Create a VM from an image
- ✔ Share a gallery

This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

If you choose to install and use the CLI locally, this tutorial requires that you're running the Azure CLI version 2.35.0 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Overview

An Azure Compute Gallery simplifies custom image sharing across your organization. Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap configurations such as preloading applications, application configurations, and other OS configurations.

The Azure Compute Gallery lets you share your custom VM images with others. Choose which images you want to share, which regions you want them to be available in, and who you want to share them with.

The Azure Compute Gallery feature has multiple resource types:

Expand table

Resource	Description
Image source	This is a resource that can be used to create an **image version** in a gallery. An image source can be an existing Azure VM that is either generalized or specialized, a managed image, a snapshot, or an image version in another gallery.
Gallery	Like the Azure Marketplace, a **gallery** is a repository for managing and sharing images and VM applications, but you control who has access.
Image definition	Image definitions are created within a gallery and carry information about the image and requirements for using it internally. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements. It is a definition of a type of image.
Image version	An **image version** is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an **image version** to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times.

Before you begin

The following steps show how to take an existing VM and turn it into a reusable custom image that you can use to create new VM instances.

To complete the example in this tutorial, you must have an existing virtual machine. If needed, you can see the CLI quickstart to create a VM to use for this tutorial. When working through the tutorial, replace the resource names where needed.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create a gallery

A gallery is the primary resource used for enabling image sharing.

Allowed characters for gallery name are uppercase or lowercase letters, digits, dots, and periods. The gallery name can't contain dashes. Gallery names must be unique within

your subscription.

Create a gallery using az sig create. The following example creates a resource group named gallery named *myGalleryRG* in *East US*, and a gallery named *myGallery*.

Azure CLI

```
az group create --name myGalleryRG --location eastus
az sig create --resource-group myGalleryRG --gallery-name myGallery
```

Get information about the VM

You can see a list of VMs that are available using az vm list.

Azure CLI

```
az vm list --output table
```

Once you know the VM name and what resource group it is in, get the ID of the VM using az vm get-instance-view.

Azure CLI

```
az vm get-instance-view -g MyResourceGroup -n MyVm --query id
```

Copy the ID of your VM to use later.

Create an image definition

Image definitions create a logical grouping for images. They're used to manage information about the image versions that are created within them.

Image definition names can be made up of uppercase or lowercase letters, digits, dots, dashes, and periods.

For more information about the values you can specify for an image definition, see Image definitions.

Create an image definition in the gallery using az sig image-definition create.

In this example, the image definition is named *myImageDefinition*, and is for a specialized Linux OS image.

Azure CLI

```
az sig image-definition create \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   --publisher myPublisher \
   --offer myOffer \
   --sku mySKU \
   --os-type Linux \
   --os-state specialized
```

Copy the ID of the image definition from the output to use later.

Create the image version

Create an image version from the VM using az sig image-version create.

Allowed characters for image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion.MinorVersion.Patch*.

In this example, the version of our image is *1.0.0* and we're going to create two replicas in the *West Central US* region, one replica in the *South Central US* region and one replica in the *East US 2* region using zone-redundant storage. The replication regions must include the region the source VM is located.

Replace the value of `--managed-image` in this example with the ID of your VM from the previous step.

Azure CLI

```
az sig image-version create \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   --gallery-image-version 1.0.0 \
   --target-regions "westcentralus" "southcentralus=1"
"eastus=1=standard_zrs" \
   --replica-count 2 \
   --managed-image "/subscriptions/<Subscription
ID>/resourceGroups/MyResourceGroup/providers/Microsoft.Compute/virtualMachin
es/myVM"
```

ⓘ Note

You need to wait for the image version to completely finish being built and replicated before you can use the same managed image to create another image version.

You can also store your image in Premium storage by a adding `--storage-account-type premium_lrs`, or Zone Redundant Storage by adding `--storage-account-type standard_zrs` when you create the image version.

Create the VM

Create the VM using az vm create using the `--specialized` parameter to indicate the image is a specialized image.

Use the image definition ID for `--image` to create the VM from the latest version of the image that is available. You can also create the VM from a specific version by supplying the image version ID for `--image`.

In this example, we're creating a VM from the latest version of the *myImageDefinition* image.

Azure CLI

```
az group create --name myResourceGroup --location eastus
az vm create --resource-group myResourceGroup \
    --name myVM2 \
    --image "/subscriptions/<Subscription
ID>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myGalle
ry/images/myImageDefinition" \
    --specialized
```

Share the gallery

You can share images across subscriptions using Azure role-based access control (Azure RBAC). You can share images at the gallery, image definition or image version level. Any user that has read permissions to an image version, even across subscriptions, will be able to deploy a VM using the image version.

We recommend that you share with other users at the gallery level. To get the object ID of your gallery, use az sig show.

Azure CLI

```
az sig show \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --query id
```

Use the object ID as a scope, along with an email address and az role assignment create to give a user access to the Azure Compute Gallery. Replace `<email-address>` and `<gallery iD>` with your own information.

Azure CLI

```
az role assignment create \
   --role "Reader" \
   --assignee <email address> \
   --scope <gallery ID>
```

For more information about how to share resources using Azure RBAC, see Add or remove Azure role assignments using Azure CLI.

Azure Image Builder

Azure also offers a service, built on Packer, Azure VM Image Builder. Describe your customizations in a template, and it will handle the image creation.

Next steps

In this tutorial, you created a custom VM image. You learned how to:

- ✓ Create an Azure Compute Gallery
- ✓ Create an image definition
- ✓ Create an image version
- ✓ Create a VM from an image
- ✓ Share a gallery

Advance to the next tutorial to learn about Virtual Machine Scale Sets.

Create a virtual machine scale set

Feedback

Was this page helpful? Yes No

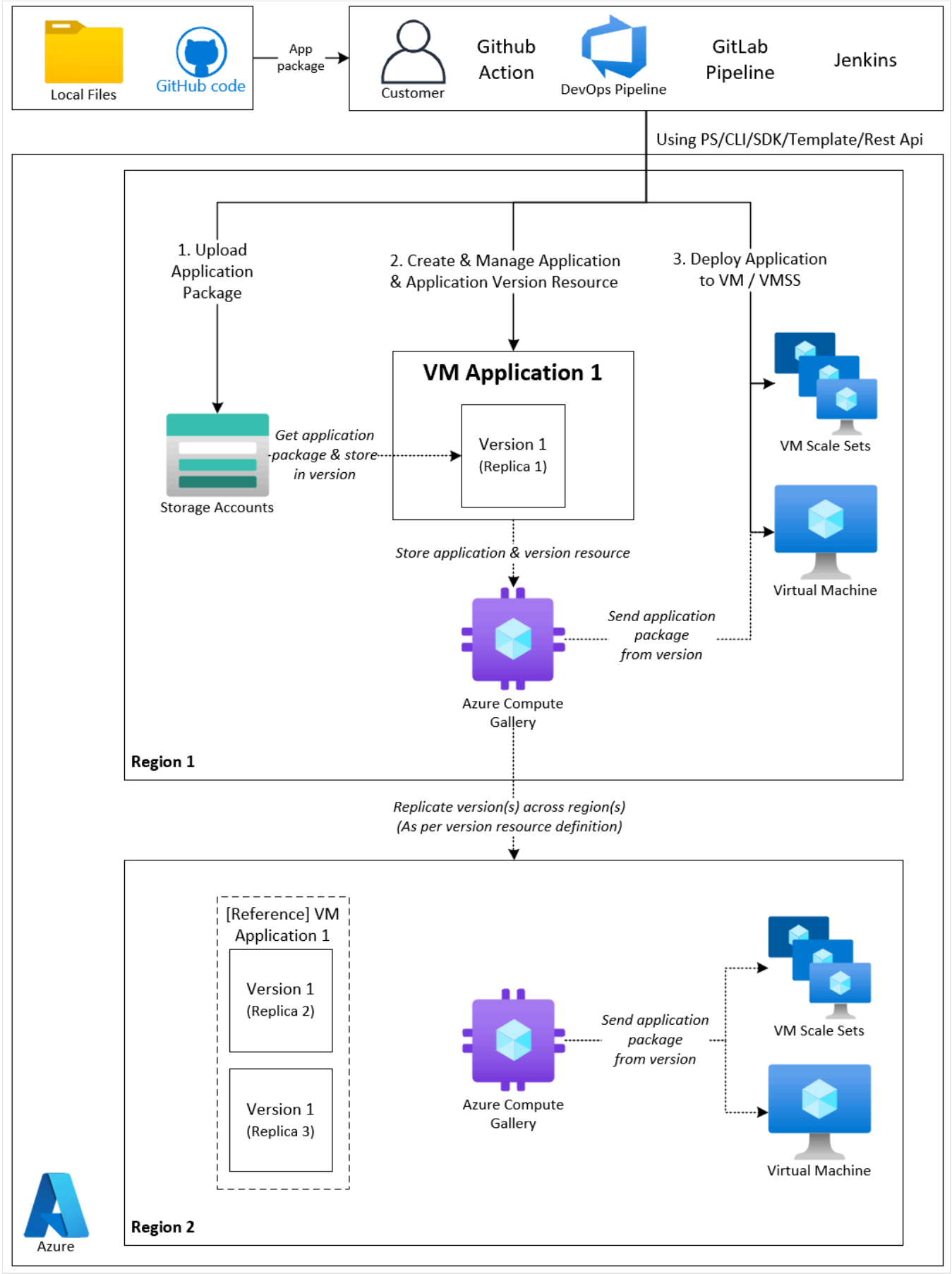
Provide product feedback | Get help at Microsoft Q&A

Create and deploy VM Application

Summarize this article for me

VM Application is a resource type in Azure Compute Gallery that simplifies publishing, deployment, management, sharing, and global distribution of applications and scripts for your virtual machines. VM Applications support a wide range of scenarios, including high-scale deployments, low-latency requirements, failure resiliency, secure trusted rollouts, fleet-wide consistency, microservice architectures, and post-deployment management. Learn more about VM Applications.

To create and deploy applications on Azure VM, first package and upload your application to Azure Storage Account as a storage blob. Then create `Azure VM application` resource and `VM application version` resource referencing these storage blobs. Azure stores and replicates these blobs in regional managed storage accounts. Finally, deploy the application on any VM or Virtual Machine Scale Sets by passing application reference in `applicationProfile`.



Prerequisites

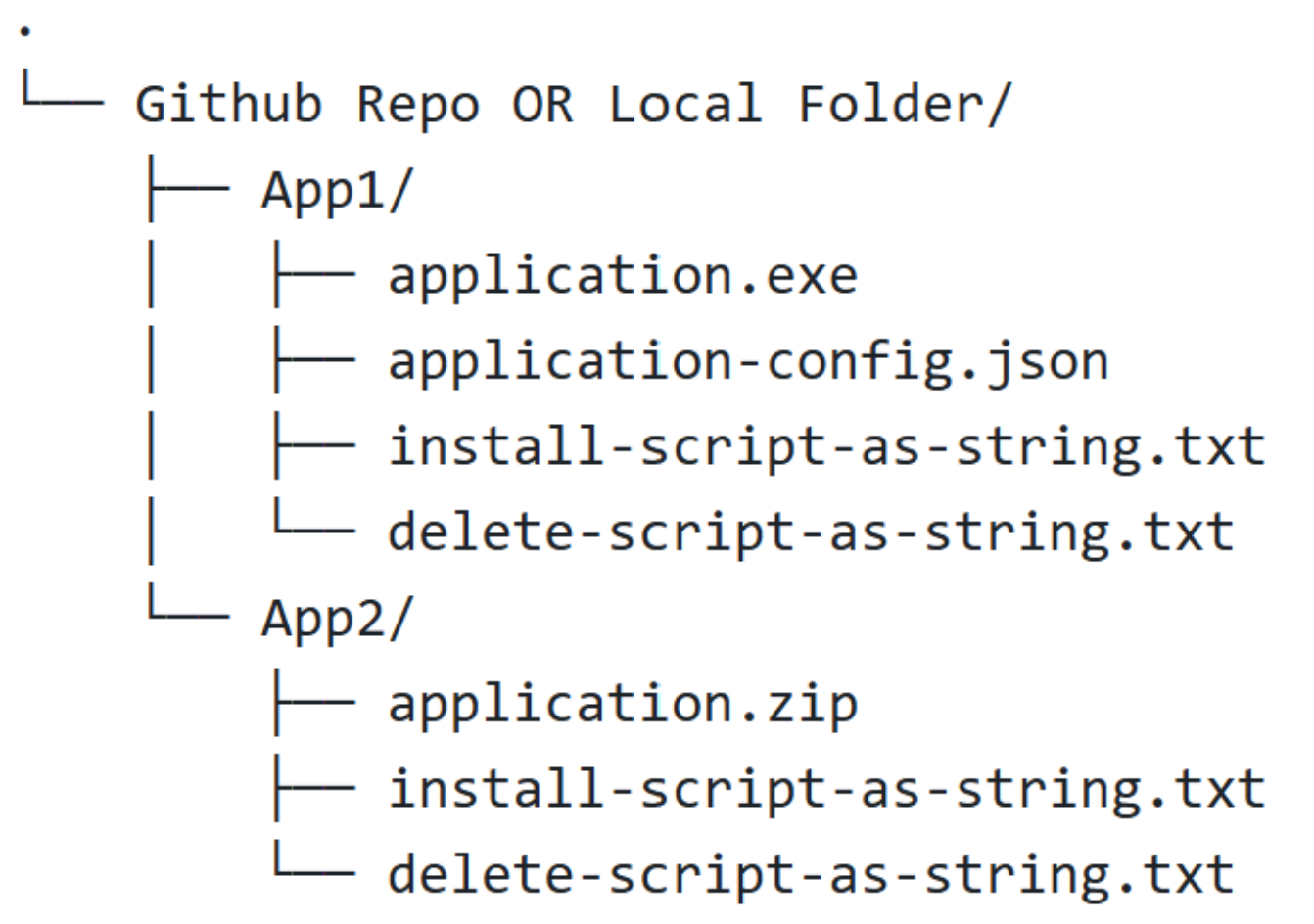
1. Create Azure storage account and storage container. This container is used to upload your application files.

2. Create Azure Compute Gallery for storing and sharing application resources.
3. Create user-assigned managed identity and assign to Azure Compute Gallery

> ⓘ Note
>
> Create your storage account with anonymous access disabled for improved security. Use a managed identity assigned to Azure Compute Gallery to publish your VM Applications securely without requiring SAS tokens.

Package the application

Before publishing the VM Application resource in Azure Compute Gallery, package your application files for upload to Azure Storage Account. The following diagram shows possible folder structures for organizing your application files:



1. Package the application files

VM Application requires a single file as the application package.

- If your application installation requires a single executable file (.exe, .msi, .sh, .ps, etc.) then you can use it as the application package.
- If your application requires multiple files (executables, dependencies, scripts, metadata, etc.), archive them into a single file (`.zip`, `.tar`, `.tar.gz`) and use this file as the application package.
- For microservice application, you can package and publish each microservice as a separate Azure VM Application. This modularization facilitates application reusability, cross-team development, and sequential installation of microservices using `order` property in the applicationProfile.

Learn more about creating application packages for Azure VM Applications.

2. (Optional) Package the application configuration file

- You can provide the configuration file separately using `defaultConfigurationLink` in the `publishingProfile` of the VM Application. This reduces the overhead of archiving and unarchiving application packages.
- Configuration files can also be passed during app deployment using `configurationReference` property in the `applicationProfile` of the VM enabling customized installation per VM.
- If both properties are set, `configurationReference` overrides the configuration passed using `defaultConfigurationLink`.

3. Create the install script

After the application and configuration blob is downloaded on the VM, Azure executes the provided install script to install the application. **The install script is provided as a string** and has a maximum character limit of 4,096 chars. The install commands should be written assuming the application package and the configuration file are in the current directory.

There may be few operations required to be performed in the install script

- **(Optional) Use the right command interpreter** The default command interpreter used by Azure are `/bin/bash` on Linux OS and `cmd.exe` on Windows OS. It's possible to use a different interpreter like Chocolatey or PowerShell, if its installed on the machine. Call the executable and pass the commands to it. For example, `powershell.exe -command '<powershell command>'`. If you're using PowerShell, you need to be using version 3.11.0 of the Az.Storage module.

- **(Optional) Rename application blob and configuration blob** Azure can't retain the original blob name and the file extensions. Azure downloads the application file with the

name of the VM application resource published on Azure Compute Gallery. So if name of VM App is "VMApp1", the file downloaded on VM has the name "VMApp1" without an extension. The configuration file has a default name as "VMApp1-config" without a file extension.

You can rename the file with the file extension using the install script or you can also pass the names in `packageFileName` and `configFileName` properties of the publishingProfile of VM Application version resource. Azure uses these names instead of default names while downloading the files.

- **Unarchive application blob** For archived application packages, it needs to be unarchived before installing the application. Use .zip or .tar since most OS has built-in support for unarchiving these formats. For other formats, make sure the Guest OS provides support.

- **Convert the script to string** The install script is passed as a string for the `install` property in the `publishingProfile` of Azure VM Application version resource.

- **Install quietly and synchronously** Install applications quietly or silently without any user interface prompts. Since the installation runs on a VM where customer input isn't available, you must use silent or unattended installation flags. Additionally, if the application installer spawns background or child processes or runs asynchronously, the parent script might return before the installation completes. To ensure the full installation finishes before Azure marks the operation as successful, use wait flags or blocking commands. This approach helps when installation involves multiple steps that run asynchronously or you need to verify installation completes before the subsequent application in the `order` sequence begins.

  Common examples include:
  - **Windows .msi**: `start /wait msiexec /i app.msi /quiet /qn`
  - **Windows .exe**: `start /wait app.exe /silent` or `start /wait app.exe /quiet` or `start /wait app.exe /S`
  - **PowerShell**: `Start-Process -FilePath '.\app.exe' -ArgumentList '/quiet' -Wait`
  - **Linux .sh**: `./install.sh --silent` (shell scripts run synchronously by default)
  - **Linux .deb**: `sudo DEBIAN_FRONTEND=noninteractive dpkg -i app.deb` (runs synchronously by default)
  - **Linux .rpm**: `sudo rpm -ivh app.rpm` (runs synchronously by default)

- **(Optional) Set right execution policy and permissions** After unarchiving, file permissions could be reset. It's a good practice to set the right permissions before executing the files.

- **(Optional) Set verbose execution and logging** VM Applications automatically captures any output from the install script and stores it in a stdout file on the VM. Use this

capability to log installation progress, debug failures, and verify successful deployments.
    - **Enable verbose mode** to capture application installer logs. Bash: `set -x`, PowerShell: `$VerbosePreference='Continue'`, Cmd: `@echo on`.
    - **Add custom messages** to track script progress. Bash: `echo "Install started"`, PowerShell: `Write-Output "Install started"`, Cmd: `echo Install started`

    You can retrieve these logs using Run Command or by connecting to the VM. The `stdout` and `stderr` files are located at the following path
    - Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>/`
    - Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.9\Downloads\<application name>\<application version>/`

- **(Optional) Move application and configuration blob to appropriate location** Azure downloads the application blob and configuration blob to following locations. The install script must move the files to appropriate locations when necessary.

    Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>`

    Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.16\Downloads\<application name>\<application version>`

Here are few sample install scripts based on the file extension of the application blob. If `packageFileName` & `configFileName` properties are provided in the `publishingProfile` with file name and extension, remove the code for renaming files in following scripts.

.TAR

Replace `VMApp1` with your VM App name in application package (VMApp1) and configuration file (VMApp1-config). Replace `app.tar` with desired app name. Replace `config.yaml` with your desired config name.

Install using bash without config file:

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && bash ./install.sh
```

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && sudo
DEBIAN_FRONTEND=noninteractive dpkg -i ./app.deb
```

Install using bash with config file:

Bash

```
mv VMApp1 app.tar && mv VMApp1-config config.yaml && tar -xf app.tar && chmod -
R +xr . && bash ./install.sh --config config.yaml
```

If config file is packaged within the tar file

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && sudo debconf-set-
selections < config.cfg && sudo DEBIAN_FRONTEND=noninteractive dpkg -i
./app.deb
```

4. Create the remove script

The `remove` script enables you to define the operations for removing the application. The `remove` script is provided as a string and has a maximum character limit of 4,096 characters. Write the commands assuming the application package and the configuration file are in the current directory. During uninstall operation, Azure runs the uninstall script and then deletes all files from the repository.

There are few operations that the `remove` script must perform.

- **Uninstall application:** Properly uninstall the application from the VM. For example:

CMD

CMD on Windows:

cli

```
start /wait uninstall.exe /quiet
```

cli

```
start /wait python.exe /uninstall /quiet
```

cli

```
start /wait msiexec /x app.msi /quiet /qn
```

- **Remove residual files:** If the application installation creates files in other parts of the file system, remove those files.

Upload the application files to Azure storage account

> ⓘ **Important**
>
> Before uploading files or generating SAS URLs, assign roles to the user, service principal, or managed identity that performs these operations on the target Storage account or container:
>
> - Storage Blob Data Contributor or Storage Blob Data Owner: required to upload, modify, or delete blobs.
> - For generating SAS with Azure AD (for example, using `--auth-mode` login in CLI/PowerShell), also assign Storage Blob Delegator at the storage account scope.

1. Upload your application and configuration files to a container in your Azure storage account.

Upload your packaged application and optional configuration file as blobs. Block blobs work for most scenarios.

Optionally, if you need to use page blobs, byte-align your files before uploading using one of the following scripts:

CLI

Azure CLI

```
inputFile="<the file you want to pad>"

# Get the file size
fileSize=$(stat -c %s "$inputFile")

# Calculate the remainder when divided by 512
```

```
remainder=$((fileSize % 512))

if [ "$remainder" -ne 0 ]; then
  # Calculate how many bytes to pad
  difference=$((512 - remainder))

  # Create padding (empty bytes)
  dd if=/dev/zero bs=1 count=$difference >> "$inputFile"
fi
```

2. Generate SAS or Blob URL for the application package and the configuration file

Once the application and configuration files are uploaded to the storage account, you need to generate a Blob URL or SAS URL with read privilege for these blobs. These URLs are then provided as reference while creating the VM Application version resource.

> ⓘ **Note**
>
> Use of blob URL with managed identity is strongly recommended for improved security and privacy.

Use blob URL:

- If managed identity is assigned to Azure Compute Gallery.
- If storage account is enabled for anonymous access. Use this method only for testing since its insecure.

Use SAS URL:

- If storage account is disabled for anonymous access and managed identity isn't assigned to Azure Compute Gallery.

Blob URL = https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${BLOB_NAME}
SAS URL = https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${BLOB_NAME}?${SAS_TOKEN}

CLI using Blob URL

Use the following script if managed identity is assigned to Azure Compute Gallery. This approach uses blob URLs without SAS tokens.

Azure CLI

```
#!/bin/bash
set -euo pipefail

# === CONFIGURATION ===
STORAGE_ACCOUNT="your-storage-account-name"
CONTAINER_NAME="your-container-name"
APP_FILE="./path/to/your-app-file"           # Path to your application payload
file
CONFIG_FILE="./path/to/your-config-file"     # Path to your configuration file
(optional)

# === LOGIN (if not already logged in) ===
az login --only-show-errors

# === CREATE CONTAINER IF NOT EXISTS ===
az storage container create \
  --name "$CONTAINER_NAME" \
  --account-name "$STORAGE_ACCOUNT" \
  --auth-mode login \
  --only-show-errors

# === UPLOAD APPLICATION FILE ===
APP_BLOB_NAME=$(basename "$APP_FILE")
az storage blob upload \
  --account-name "$STORAGE_ACCOUNT" \
  --container-name "$CONTAINER_NAME" \
  --name "$APP_BLOB_NAME" \
  --file "$APP_FILE" \
  --auth-mode login \
  --overwrite \
  --only-show-errors

# === UPLOAD CONFIG FILE (optional) ===
if [ -n "$CONFIG_FILE" ] && [ -f "$CONFIG_FILE" ]; then
  CONFIG_BLOB_NAME=$(basename "$CONFIG_FILE")
  az storage blob upload \
    --account-name "$STORAGE_ACCOUNT" \
    --container-name "$CONTAINER_NAME" \
    --name "$CONFIG_BLOB_NAME" \
    --file "$CONFIG_FILE" \
    --auth-mode login \
    --overwrite \
    --only-show-errors
fi

# === GENERATE BLOB URLs ===
echo "Generating Blob URLs..."

APP_BLOB_URL="https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME
}/${APP_BLOB_NAME}"
echo "Application file: $APP_BLOB_URL"
```

```
if [ -n "$CONFIG_FILE" ] && [ -f "$CONFIG_FILE" ]; then

CONFIG_BLOB_URL="https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${CONFIG_BLOB_NAME}"
  echo "Configuration file: $CONFIG_BLOB_URL"
fi
```

Create the VM Application

To create the VM Application, first create the **VM Application** resource, which describes the application. Then create **VM Application Version** resource within it, which contains the VM application payload and scripts to install, update, and delete the application. Payload is supplied using Blob URL or SAS URL to the blob container in Azure Storage Account.

Refer schema for VM Application and VM Application version resource to learn more about each property.

Template

Use the following ARM template to create an Azure Compute Gallery, VM Application, and VM Application version. This template demonstrates the key properties and configuration options for publishing your application package.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2020-06-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "galleryName": {
      "type": "string"
    },
    "applicationName": {
      "type": "string"
    },
    "versionName": {
      "type": "string",
      "metadata": {
        "description": "Must follow the format: major.minor.patch (Example: 1.0.0)"
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "West US"
    },
```

```
    "supportedOSType": {
      "type": "string",
      "allowedValues": ["Windows", "Linux"]
    },
    "endOfLifeDate": {
      "type": "string",
      "metadata": {
        "description": "Optional. This property is for information only and
doesn't block app deployment."
      }
    },
    "description": {
      "type": "string",
      "defaultValue": "Description of the application"
    },
    "eula": {
      "type": "string",
      "defaultValue": ""
    },
    "privacyStatementUri": {
      "type": "string",
      "defaultValue": ""
    },
    "releaseNoteUri": {
      "type": "string",
      "defaultValue": ""
    },
    "mediaLink": {
      "type": "string"
    },
    "configLink": {
      "type": "string"
    },
    "appConfigFileName": {
      "type": "string"
    },
    "appPackageFileName": {
      "type": "string"
    },
    "replicaRegion1": {
      "type": "string",
      "defaultValue": "East US"
    },
    "replicaRegion2": {
      "type": "string",
      "defaultValue": "South Central US"
    },
    "installScript": {
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to install the application.
Example: echo 'Installing application...'"
      }
    },
    "updateScript": {
```

```
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to update the application.
Example: echo 'Updating application...'"
      }
    },
    "removeScript": {
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to delete the application.
Example: echo 'Deleting application...'"
      }
    },
    "storageAccountType": {
      "type": "string",
      "allowedValues": ["PremiumV2_LRS", "Premium_LRS", "Standard_LRS",
"Standard_ZRS"],
      "defaultValue": "Standard_LRS"
    }
  },
  "resources": [
    {
      "type": "Microsoft.Compute/galleries",
      "apiVersion": "2024-03-03",
      "name": "[parameters('galleryName')]",
      "location": "[parameters('location')]",
      "properties": {
        "identifier": {}
      }
    },
    {
      "type": "Microsoft.Compute/galleries/applications",
      "apiVersion": "2024-03-03",
      "name": "[format('{0}/{1}', parameters('galleryName'),
parameters('applicationName'))]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Compute/galleries',
parameters('galleryName'))]"
      ],
      "properties": {
        "supportedOSType": "[parameters('supportedOSType')]",
        "endOfLifeDate": "[parameters('endOfLifeDate')]",
        "description": "[parameters('description')]",
        "eula": "[if(equals(parameters('eula'), ''), json('null'),
parameters('eula'))]",
        "privacyStatementUri": "[if(equals(parameters('privacyStatementUri'),
''), json('null'), parameters('privacyStatementUri'))]",
        "releaseNoteUri": "[if(equals(parameters('releaseNoteUri'), ''),
json('null'), parameters('releaseNoteUri'))]"
      }
    },
    {
      "type": "Microsoft.Compute/galleries/applications/versions",
      "apiVersion": "2024-03-03",
```

```
      "name": "[format('{0}/{1}/{2}', parameters('galleryName'),
parameters('applicationName'), parameters('versionName'))]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Compute/galleries/applications',
parameters('galleryName'), parameters('applicationName'))]"
      ],
      "properties": {
        "publishingProfile": {
          "source": {
            "mediaLink": "[parameters('mediaLink')]",
            "defaultConfigurationLink": "[parameters('configLink')]"
          },
          "manageActions": {
            "install": "[parameters('installScript')]",
            "remove": "[parameters('removeScript')]",
            "update": "[parameters('updateScript')]"
          },
          "settings": {
            "scriptBehaviorAfterReboot": "Rerun",
            "configFileName": "[parameters('appConfigFileName')]",
            "packageFileName": "[parameters('appPackageFileName')]"
          },
          "targetRegions": [
            {
              "name": "[parameters('location')]",
              "regionalReplicaCount": 3,
              "storageAccountType": "[parameters('storageAccountType')]"
            },
            {
              "name": "[parameters('replicaRegion1')]",
              "regionalReplicaCount": 1,
              "storageAccountType": "[parameters('storageAccountType')]"
            },
            {
              "name": "[parameters('replicaRegion2')]"
            },
          ],
          "excludeFromLatest": false,
          "replicaCount": 2,
          "storageAccountType": "[parameters('storageAccountType')]"
        },
        "safetyProfile": {
          "allowDeletionOfReplicatedLocations": true
        },
        "endOfLifeDate": "[parameters('endOfLifeDate')]"
      }
    }
  ]
}
```

Deploy the VM Apps

One or more VM Applications can now be referenced in the `applicationProfile` of Azure VM or Azure Virtual Machine Scale Sets. Azure then pulls the payload of the VM Application and installs it on each VM using the provided install script. The `order` property defines the sequential order in which the VM Applications are installed on the VM.

Refer schema of applicationProfile of the VM / Virtual Machine Scale Sets to learn more about each property.

Template

Use the following ARM template to deploy VM Application on Azure VM or Azure Virtual Machine Scale Sets. This template demonstrates the key properties and configuration options for deploying your VM application.

Deploy VM Application on Virtual Machine Scale Sets

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmssName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "subscriptionId": {
      "type": "string"
    },
    "resourceGroupName": {
      "type": "string"
    },
    "galleryName": {
      "type": "string"
    },
    "applicationName1": {
      "type": "string"
    },
    "applicationVersion1": {
      "type": "string",
      "defaultValue": "latest"
    },
    "configurationReference1": {
      "type": "string",
```

```
      "metadata": {
        "description": "Optional path to configuration file from Storage
Account. Overrides default configuration file."
      }
    },
    "applicationName2": {
      "type": "string"
    },
    "applicationVersion2": {
      "type": "string",
      "defaultValue": "latest"
    }
  },
  "variables": {
    "packageReferenceId1": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName1'), parameters('applicationVersion1'))]",
    "packageReferenceId2": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName2'), parameters('applicationVersion2'))]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachineScaleSets",
      "apiVersion": "2024-03-03",
      "name": "[parameters('vmssName')]",
      "location": "[parameters('location')]",
      "properties": {
        "virtualMachineProfile": {
          "applicationProfile": {
            "galleryApplications": [
              {
                "order": 1,
                "packageReferenceId": "[variables('packageReferenceId1')]",
                "configurationReference": "
[parameters('configurationReference1')]",
                "treatFailureAsDeploymentFailure": true
              },
              {
                "order": 2,
                "packageReferenceId": "[variables('packageReferenceId2')]",
                "treatFailureAsDeploymentFailure": false
              }
            ]
          }
        }
      }
    }
  ]
}
```

Deploy VM Application on Azure VM

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "subscriptionId": {
      "type": "string"
    },
    "resourceGroupName": {
      "type": "string"
    },
    "galleryName": {
      "type": "string"
    },
    "applicationName1": {
      "type": "string"
    },
    "applicationVersion1": {
      "type": "string",
      "defaultValue": "latest"
    },
    "configurationReference1": {
      "type": "string",
      "metadata": {
        "description": "Optional path to configuration blob for application 1"
      }
    },
    "applicationName2": {
      "type": "string"
    },
    "applicationVersion2": {
      "type": "string",
      "defaultValue": "latest"
    }
  },
  "variables": {
    "packageReferenceId1": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName1'), parameters('applicationVersion1'))]",
    "packageReferenceId2": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
```

```
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName2'), parameters('applicationVersion2'))]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2024-07-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "applicationProfile": {
          "galleryApplications": [
            {
              "order": 1,
              "packageReferenceId": "[variables('packageReferenceId1')]",
              "configurationReference": "
[parameters('configurationReference1')]",
              "treatFailureAsDeploymentFailure": true
            },
            {
              "order": 2,
              "packageReferenceId": "[variables('packageReferenceId2')]",
              "treatFailureAsDeploymentFailure": false
            }
          ]
        }
      }
    }
  ]
}
```

Next steps

- Learn to manage, update, or delete Azure VM Applications.
- Learn more about Azure VM Applications.

Last updated on 03/06/2026

Tutorial - How to use cloud-init to customize a Linux virtual machine in Azure on first boot

Article • 05/02/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

In a previous tutorial, you learned how to SSH to a virtual machine (VM) and manually install NGINX. To create VMs in a quick and consistent manner, some form of automation is typically desired. A common approach to customize a VM on first boot is to use cloud-init. In this tutorial you learn how to:

- ✔ Create a cloud-init config file
- ✔ Create a VM that uses a cloud-init file
- ✔ View a running Node.js app after the VM is created
- ✔ Use Key Vault to securely store certificates
- ✔ Automate secure deployments of NGINX with cloud-init

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Cloud-init overview

Cloud-init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. As cloud-init runs during the initial boot process, there are no additional steps or required agents to apply your configuration.

Cloud-init also works across distributions. For example, you don't use **apt-get install** or **yum install** to install a package. Instead you can define a list of packages to install. Cloud-init automatically uses the native package management tool for the distro you select.

We are working with our partners to get cloud-init included and working in the images that they provide to Azure. For detailed information cloud-init support for each distribution, see Cloud-init support for VMs in Azure.

Create cloud-init config file

To see cloud-init in action, create a VM that installs NGINX and runs a simple 'Hello World' Node.js app. The following cloud-init configuration installs the required packages, creates a Node.js app, then initializes and starts the app.

At your bash prompt or in the Cloud Shell, create a file named *cloud-init.txt* and paste the following configuration. For example, type `sensible-editor cloud-init.txt` to create the file and see a list of available editors. Make sure that the whole cloud-init file is copied correctly, especially the first line:

YAML

```
#cloud-config
package_upgrade: true
packages:
  - nginx
  - nodejs
  - npm
write_files:
  - owner: www-data:www-data
    path: /etc/nginx/sites-available/default
    defer: true
    content: |
      server {
        listen 80;
        location / {
          proxy_pass http://localhost:3000;
          proxy_http_version 1.1;
          proxy_set_header Upgrade $http_upgrade;
          proxy_set_header Connection keep-alive;
          proxy_set_header Host $host;
          proxy_cache_bypass $http_upgrade;
        }
      }
  - owner: azureuser:azureuser
    path: /home/azureuser/myapp/index.js
    defer: true
    content: |
      var express = require('express')
      var app = express()
      var os = require('os');
      app.get('/', function (req, res) {
        res.send('Hello World from host ' + os.hostname() + '!')
      })
      app.listen(3000, function () {
        console.log('Hello world app listening on port 3000!')
      })
runcmd:
  - service nginx restart
  - cd "/home/azureuser/myapp"
  - npm init
```

```
  - npm install express -y
  - nodejs index.js
```

For more information about cloud-init configuration options, see cloud-init config examples.

Create virtual machine

Before you can create a VM, create a resource group with az group create. The following example creates a resource group. In these commands, a random suffix is appended to the resource group and VM names to prevent name collisions during repeated deployments.

Bash

```
export RANDOM_SUFFIX=$(openssl rand -hex 3)
export RESOURCE_GROUP="myResourceGroupAutomate$RANDOM_SUFFIX"
export REGION="eastus2"
az group create --name $RESOURCE_GROUP --location $REGION
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxx-xxxxx-xxxxx-
xxxxx/resourceGroups/myResourceGroupAutomatexxx",
  "location": "eastus",
  "managedBy": null,
  "name": "myResourceGroupAutomatexxx",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```


Now create a VM with az vm create. Use the `--custom-data` parameter to pass in your cloud-init config file. Provide the full path to the *cloud-init.txt* config if you saved the file outside of your present working directory. The following example creates a VM; note that the VM name is also appended with the random suffix.

Bash

```
export VM_NAME="myAutomatedVM$RANDOM_SUFFIX"
az vm create \
    --resource-group $RESOURCE_GROUP \
    --name $VM_NAME \
```

```
    --image Ubuntu2204 \
    --admin-username azureuser \
    --generate-ssh-keys \
    --custom-data cloud-init.txt
```

Results:

JSON


```
{
  "fqdns": "",
  "id":
"/subscriptions/xxxxx/resourceGroups/myResourceGroupAutomatexxx/providers/Microsof
t.Compute/virtualMachines/myAutomatedVMxxx",
  "location": "eastus",
  "name": "myAutomatedVMxxx",
  "powerState": "VM running",
  "publicIpAddress": "x.x.x.x",
  "resourceGroup": "myResourceGroupAutomatexxx",
  "zones": ""
}
```


It takes a few minutes for the VM to be created, the packages to install, and the app to start. There are background tasks that continue to run after the Azure CLI returns you to the prompt. It may be another couple of minutes before you can access the app. When the VM has been created, take note of the `publicIpAddress` displayed by the Azure CLI. This address is used to access the Node.js app via a web browser.

To allow web traffic to reach your VM, open port 80 from the Internet with az vm open-port:

Bash

```
az vm open-port --port 80 --resource-group $RESOURCE_GROUP --name $VM_NAME
```

Results:

JSON


```
{
  "endpoints": [
    {
      "name": "80",
      "protocol": "tcp",
      "publicPort": 80,
      "privatePort": 80
    }
  ],
  "id":
```

```
"/subscriptions/xxxxx/resourceGroups/myResourceGroupAutomatexxx/providers/Microsoft.Compute/virtualMachines/myAutomatedVMxxx",
  "location": "eastus",
  "name": "myAutomatedVMxxx"
}
```


Test web app

Now you can open a web browser and enter the public IP address in the address bar. Provide your own public IP address from the VM create process. Your Node.js app is displayed as shown in the following example:



Next steps

In this tutorial, you configured VMs on first boot with cloud-init. You learned how to:

- ✓ Create a cloud-init config file
- ✓ Create a VM that uses a cloud-init file
- ✓ View a running Node.js app after the VM is created
- ✓ Use Key Vault to securely store certificates
- ✓ Automate secure deployments of NGINX with cloud-init

Advance to the next tutorial to learn how to create custom VM images.

Create custom VM images

Tutorial: Create and manage Azure virtual networks for Linux virtual machines with the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure virtual machines use Azure networking for internal and external network communication. This tutorial walks through deploying two virtual machines and configuring Azure networking for these VMs. The examples in this tutorial assume that the VMs are hosting a web application with a database back-end, however an application is not deployed in the tutorial. In this tutorial, you learn how to:

- ✔ Create a virtual network and subnet
- ✔ Create a public IP address
- ✔ Create a front-end VM
- ✔ Secure network traffic
- ✔ Create a back-end VM

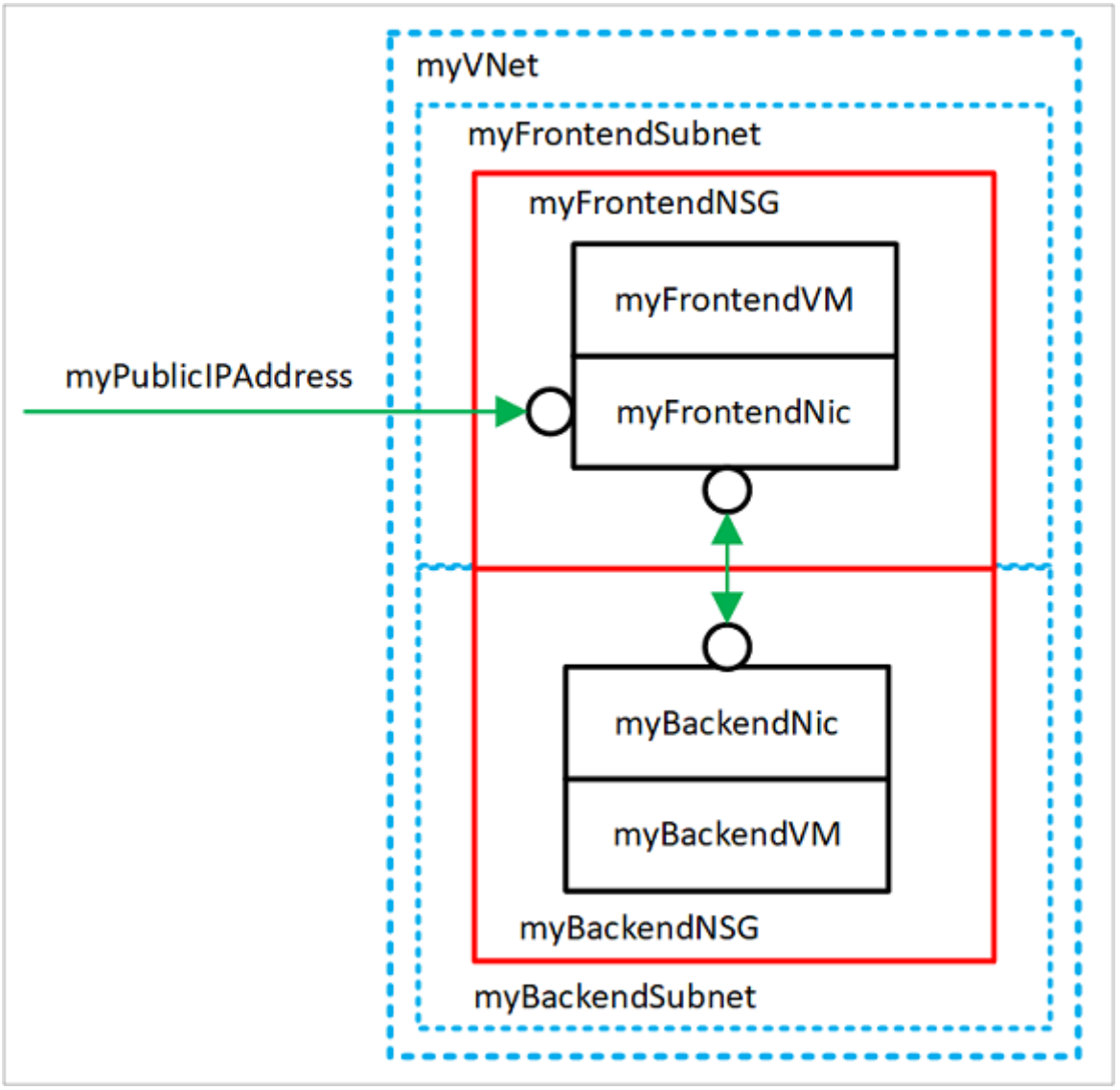
This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

VM networking overview

Azure virtual networks enable secure network connections between virtual machines, the internet, and other Azure services such as Azure SQL Database. Virtual networks are broken down into logical segments called subnets. Subnets are used to control network flow, and as a security boundary. When deploying a VM, it generally includes a virtual network interface, which is attached to a subnet.

As you complete the tutorial, the following virtual network resources are created:



- *myVNet* - The virtual network that the VMs use to communicate with each other and the internet.
- *myFrontendSubnet* - The subnet in *myVNet* used by the front-end resources.
- *myPublicIPAddress* - The public IP address used to access *myFrontendVM* from the internet.
- *myFrontentNic* - The network interface used by *myFrontendVM* to communicate with *myBackendVM*.
- *myFrontendVM* - The VM used to communicate between the internet and *myBackendVM*.
- *myBackendNSG* - The network security group that controls communication between the *myFrontendVM* and *myBackendVM*.
- *myBackendSubnet* - The subnet associated with *myBackendNSG* and used by the back-end resources.
- *myBackendNic* - The network interface used by *myBackendVM* to communicate with *myFrontendVM*.
- *myBackendVM* - The VM that uses port 22 and 3306 to communicate with *myFrontendVM*.

Create a virtual network and subnet

For this tutorial, a single virtual network is created with two subnets. A front-end subnet for hosting a web application, and a back-end subnet for hosting a database server.

Before you can create a virtual network, create a resource group with az group create. The following example creates a resource group named *myRGNetwork* in the eastus location.

Azure CLI

```
az group create --name myRGNetwork --location eastus
```

Create virtual network

Use the az network vnet create command to create a virtual network. In this example, the network is named *mvVNet* and is given an address prefix of *10.0.0.0/16*. A subnet is also created with a name of *myFrontendSubnet* and a prefix of *10.0.1.0/24*. Later in this tutorial a front-end VM is connected to this subnet.

Azure CLI

```
az network vnet create \
  --resource-group myRGNetwork \
  --name myVNet \
  --address-prefix 10.0.0.0/16 \
  --subnet-name myFrontendSubnet \
  --subnet-prefix 10.0.1.0/24
```

Create subnet

A new subnet is added to the virtual network using the az network vnet subnet create command. In this example, the subnet is named *myBackendSubnet* and is given an address prefix of *10.0.2.0/24*. This subnet is used with all back-end services.

Azure CLI

```
az network vnet subnet create \
  --resource-group myRGNetwork \
  --vnet-name myVNet \
  --name myBackendSubnet \
  --address-prefix 10.0.2.0/24
```

At this point, a network has been created and segmented into two subnets, one for front-end services, and another for back-end services. In the next section, virtual machines are created and connected to these subnets.

Create a public IP address

A public IP address allows Azure resources to be accessible on the internet. The allocation method of the public IP address can be configured as dynamic or static. By default, a public IP address is dynamically allocated. Dynamic IP addresses are released when a VM is deallocated. This behavior causes the IP address to change during any operation that includes a VM deallocation.

The allocation method can be set to static, which ensures that the IP address remains assigned to a VM, even during a deallocated state. When using a statically allocated IP address, the IP address itself cannot be specified. Instead, it is allocated from a pool of available addresses.

Azure CLI

```
az network public-ip create --resource-group myRGNetwork --name
myPublicIPAddress
```

When creating a VM with the az vm create command, the default public IP address allocation method is dynamic. When creating a virtual machine using the az vm create command, include the `--public-ip-address-allocation static` argument to assign a static public IP address. This operation is not demonstrated in this tutorial, however in the next section a dynamically allocated IP address is changed to a statically allocated address.

Change allocation method

The IP address allocation method can be changed using the az network public-ip update command. In this example, the IP address allocation method of the front-end VM is changed to static.

First, deallocate the VM.

Azure CLI

```
az vm deallocate --resource-group myRGNetwork --name myFrontendVM
```

Use the az network public-ip update command to update the allocation method. In this case, the `--allocation-method` is being set to *static*.

Azure CLI

```
az network public-ip update --resource-group myRGNetwork --name
myPublicIPAddress --allocation-method static
```

Start the VM.

Azure CLI

```
az vm start --resource-group myRGNetwork --name myFrontendVM --no-wait
```

No public IP address

Often, a VM does not need to be accessible over the internet. To create a VM without a public IP address, use the `--public-ip-address ""` argument with an empty set of double quotes. This configuration is demonstrated later in this tutorial.

Create a front-end VM

Use the az vm create command to create the VM named *myFrontendVM* using *myPublicIPAddress*.

Azure CLI

```
az vm create \
  --resource-group myRGNetwork \
  --name myFrontendVM \
  --vnet-name myVNet \
  --subnet myFrontendSubnet \
  --nsg myFrontendNSG \
  --public-ip-address myPublicIPAddress \
  --image Ubuntu2204 \
  --generate-ssh-keys
```

Secure network traffic

A network security group (NSG) contains a list of security rules that allow or deny network traffic to resources connected to Azure Virtual Networks (VNet). NSGs can be associated to subnets or individual network interfaces. When an NSG is associated with

a network interface, it applies only the associated VM. When an NSG is associated to a subnet, the rules apply to all resources connected to the subnet.

Network security group rules

NSG rules define networking ports over which traffic is allowed or denied. The rules can include source and destination IP address ranges so that traffic is controlled between specific systems or subnets. NSG rules also include a priority (between 1—and 4096). Rules are evaluated in the order of priority. A rule with a priority of 100 is evaluated before a rule with priority 200.

All NSGs contain a set of default rules. The default rules cannot be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create.

The default rules for NSGs are:

- **Virtual network** - Traffic originating and ending in a virtual network is allowed both in inbound and outbound directions.
- **Internet** - Outbound traffic is allowed, but inbound traffic is blocked.
- **Load balancer** - Allow Azure's load balancer to probe the health of your VMs and role instances. If you are not using a load balanced set, you can override this rule.

Create network security groups

A network security group can be created at the same time as a VM using the az vm create command. When doing so, the NSG is associated with the VMs network interface and an NSG rule is auto created to allow traffic on port *22* from any source. Earlier in this tutorial, the front-end NSG was auto-created with the front-end VM. An NSG rule was also auto created for port 22.

In some cases, it may be helpful to pre-create an NSG, such as when default SSH rules should not be created, or when the NSG should be attached to a subnet.

Use the az network nsg create command to create a network security group.

Azure CLI

```
az network nsg create --resource-group myRGNetwork --name myBackendNSG
```

Instead of associating the NSG to a network interface, it is associated with a subnet. In this configuration, any VM that is attached to the subnet inherits the NSG rules.

Update the existing subnet named *myBackendSubnet* with the new NSG.

Azure CLI

```
az network vnet subnet update \
  --resource-group myRGNetwork \
  --vnet-name myVNet \
  --name myBackendSubnet \
  --network-security-group myBackendNSG
```

Secure incoming traffic

When the front-end VM was created, an NSG rule was created to allow incoming traffic on port 22. This rule allows SSH connections to the VM. For this example, traffic should also be allowed on port *80*. This configuration allows a web application to be accessed on the VM.

Use the az network nsg rule create command to create a rule for port *80*.

Azure CLI

```
az network nsg rule create \
  --resource-group myRGNetwork \
  --nsg-name myFrontendNSG \
  --name http \
  --access allow \
  --protocol Tcp \
  --direction Inbound \
  --priority 200 \
  --source-address-prefix "*" \
  --source-port-range "*" \
  --destination-address-prefix "*" \
  --destination-port-range 80
```

The front-end VM is only accessible on port *22* and port *80*. All other incoming traffic is blocked at the network security group. It may be helpful to visualize the NSG rule configurations. Return the NSG rule configuration with the az network rule list command.

Azure CLI

```
az network nsg rule list --resource-group myRGNetwork --nsg-name
myFrontendNSG --output table
```

Secure VM to VM traffic

Network security group rules can also apply between VMs. For this example, the front-end VM needs to communicate with the back-end VM on port *22* and *3306*. This configuration allows SSH connections from the front-end VM, and also allow an application on the front-end VM to communicate with a back-end MySQL database. All other traffic should be blocked between the front-end and back-end virtual machines.

Use the az network nsg rule create command to create a rule for port 22. Notice that the `--source-address-prefix` argument specifies a value of *10.0.1.0/24*. This configuration ensures that only traffic from the front-end subnet is allowed through the NSG.

Azure CLI

```
az network nsg rule create \
  --resource-group myRGNetwork \
  --nsg-name myBackendNSG \
  --name SSH \
  --access Allow \
  --protocol Tcp \
  --direction Inbound \
  --priority 100 \
  --source-address-prefix 10.0.1.0/24 \
  --source-port-range "*" \
  --destination-address-prefix "*" \
  --destination-port-range "22"
```

Now add a rule for MySQL traffic on port 3306.

Azure CLI

```
az network nsg rule create \
  --resource-group myRGNetwork \
  --nsg-name myBackendNSG \
  --name MySQL \
  --access Allow \
  --protocol Tcp \
  --direction Inbound \
  --priority 200 \
  --source-address-prefix 10.0.1.0/24 \
  --source-port-range "*" \
  --destination-address-prefix "*" \
  --destination-port-range "3306"
```

Finally, because NSGs have a default rule allowing all traffic between VMs in the same VNet, a rule can be created for the back-end NSGs to block all traffic. Notice here that the `--priority` is given a value of *300*, which is lower that both the NSG and MySQL rules. This configuration ensures that SSH and MySQL traffic is still allowed through the NSG.

Azure CLI

```
az network nsg rule create \
  --resource-group myRGNetwork \
  --nsg-name myBackendNSG \
  --name denyAll \
  --access Deny \
  --protocol Tcp \
  --direction Inbound \
  --priority 300 \
  --source-address-prefix "*" \
  --source-port-range "*" \
  --destination-address-prefix "*" \
  --destination-port-range "*"
```

Create back-end VM

Now create a virtual machine, which is attached to the *myBackendSubnet*. Notice that the `--nsg` argument has a value of empty double quotes. An NSG does not need to be created with the VM. The VM is attached to the back-end subnet, which is protected with the pre-created back-end NSG. This NSG applies to the VM. Also, notice here that the `--public-ip-address` argument has a value of empty double quotes. This configuration creates a VM without a public IP address.

Azure CLI

```
az vm create \
  --resource-group myRGNetwork \
  --name myBackendVM \
  --vnet-name myVNet \
  --subnet myBackendSubnet \
  --public-ip-address "" \
  --nsg "" \
  --image Ubuntu2204 \
  --generate-ssh-keys
```

The back-end VM is only accessible on port *22* and port *3306* from the front-end subnet. All other incoming traffic is blocked at the network security group. It may be helpful to visualize the NSG rule configurations. Return the NSG rule configuration with the az network rule list command.

Azure CLI

```
az network nsg rule list --resource-group myRGNetwork --nsg-name
myBackendNSG --output table
```

Next steps

In this tutorial, you created and secured Azure networks as related to virtual machines. You learned how to:

- ✓ Create a virtual network and subnet
- ✓ Create a public IP address
- ✓ Create a front-end VM
- ✓ Secure network traffic
- ✓ Create back-end VM

To learn about protecting your VM disks, see Backup and disaster recovery for disks.

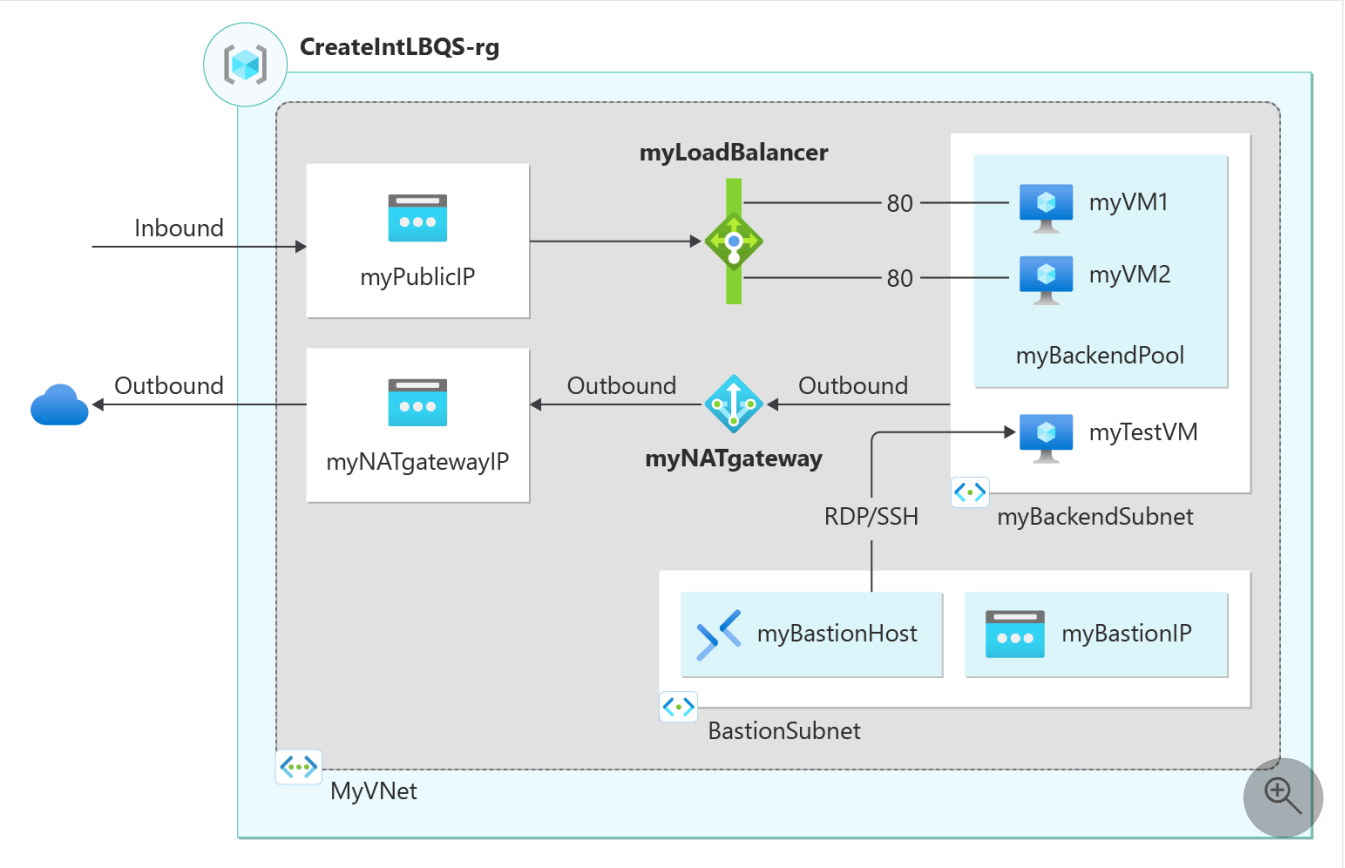
Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a public load balancer to load balance VMs using the Azure CLI

Get started with Azure Load Balancer by using the Azure CLI to create a public load balancer and two virtual machines. Along with these resources, you deploy Azure Bastion, NAT Gateway, a virtual network, and the required subnets.



If you don't have an Azure account, create a free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Get started with Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Authenticate to Azure using Azure CLI.

    - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use and manage extensions with the Azure CLI.
    - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.
- This quickstart requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with az group create:

Azure CLI

```
az group create \
  --name CreatePubLBQS-rg \
  --location eastus
```

Create a virtual network

Before you deploy VMs and test your load balancer, create the supporting virtual network and subnet.

Create a virtual network using az network vnet create. The virtual network and subnet contain the resources deployed later in this article.

Azure CLI

```
az network vnet create \
  --resource-group CreatePubLBQS-rg \
  --location eastus \
  --name myVNet \
  --address-prefixes 10.1.0.0/16 \
  --subnet-name myBackendSubnet \
  --subnet-prefixes 10.1.0.0/24
```

Create a public IP address

To access your web app on the Internet, you need a public IP address for the load balancer.

Use az network public-ip create to create the public IP for the load balancer frontend.

Azure CLI

```
az network public-ip create \
  --resource-group CreatePubLBQS-rg \
  --name myPublicIP \
  --sku Standard \
  --zone 1 2 3
```

To create a zonal public IP address in Zone 1 instead, use the following command:

Azure CLI

```
az network public-ip create \
  --resource-group CreatePubLBQS-rg \
  --name myPublicIP \
  --sku Standard \
  --zone 1
```

Create a load balancer

This section details how you can create and configure the following components of the load balancer:

- A frontend IP pool that receives the incoming network traffic on the load balancer
- A backend IP pool where the frontend pool sends the load balanced network traffic
- A health probe that determines health of the backend VM instances
- A load balancer rule that defines how traffic is distributed to the VMs

Create the load balancer resource

Create a public load balancer with az network lb create:

Azure CLI

```
az network lb create \
  --resource-group CreatePubLBQS-rg \
  --name myLoadBalancer \
  --sku Standard \
  --public-ip-address myPublicIP \
  --frontend-ip-name myFrontEnd \
  --backend-pool-name myBackEndPool
```

If the public IP created is zonal, the specified zone needs to be defined when creating the public load balancer.

Azure CLI

```
az network lb create \
  --resource-group CreatePubLBQS-rg \
  --name myLoadBalancer \
  --sku Standard \
  --public-ip-address myPublicIP \
  --frontend-ip-name myFrontEnd \
  --public-ip-zone 1 \
  --backend-pool-name myBackEndPool
```

Create the health probe

A health probe checks all virtual machine instances to ensure they can send network traffic.

A virtual machine with a failed probe check is removed from the load balancer. The virtual machine is added back into the load balancer when the failure is resolved.

Create a health probe with az network lb probe create:

Azure CLI

```
az network lb probe create \
  --resource-group CreatePubLBQS-rg \
  --lb-name myLoadBalancer \
  --name myHealthProbe \
  --protocol tcp \
  --port 80
```

Create the load balancer rule

A load balancer rule defines:

- Frontend IP configuration for the incoming traffic
- The backend IP pool to receive the traffic
- The required source and destination port

Create a load balancer rule with az network lb rule create:

Azure CLI

```
az network lb rule create \
  --resource-group CreatePubLBQS-rg \
  --lb-name myLoadBalancer \
  --name myHTTPRule \
  --protocol tcp \
  --frontend-port 80 \
  --backend-port 80 \
  --frontend-ip-name myFrontEnd \
  --backend-pool-name myBackEndPool \
  --probe-name myHealthProbe \
  --disable-outbound-snat true \
  --idle-timeout 15 \
  --enable-tcp-reset true
```

Create a network security group

For a standard load balancer, the VMs in the backend pool are required to have network interfaces that belong to a network security group.

Use az network nsg create to create the network security group:

Azure CLI

```
az network nsg create \
  --resource-group CreatePubLBQS-rg \
  --name myNSG
```

Create a network security group rule

Create a network security group rule using az network nsg rule create:

Azure CLI

```
az network nsg rule create \
  --resource-group CreatePubLBQS-rg \
  --nsg-name myNSG \
  --name myNSGRuleHTTP \
  --protocol '*' \
  --direction inbound \
  --source-address-prefix '*' \
  --source-port-range '*' \
  --destination-address-prefix '*' \
  --destination-port-range 80 \
  --access allow \
  --priority 200
```

Create a bastion host

In this section, you create the resources for Azure Bastion. Azure Bastion is used to securely manage the virtual machines in the backend pool of the load balancer.

> **Important**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

Create a public IP address

Use az network public-ip create to create a public ip address for the bastion host. The public IP is used by the bastion host for secure access to the virtual machine resources.

Azure CLI

```
az network public-ip create \
  --resource-group CreatePubLBQS-rg \
  --name myBastionIP \
  --sku Standard \
  --zone 1 2 3
```

Create a bastion subnet

Use az network vnet subnet create to create a bastion subnet. The bastion subnet is used by the bastion host to access the virtual network.

Azure CLI

```
az network vnet subnet create \
  --resource-group CreatePubLBQS-rg \
  --name AzureBastionSubnet \
  --vnet-name myVNet \
  --address-prefixes 10.1.1.0/27
```

Create bastion host

Use az network bastion create to create a bastion host. The bastion host is used to connect securely to the virtual machine resources created later in this article.

Azure CLI

```
az network bastion create \
  --resource-group CreatePubLBQS-rg \
  --name myBastionHost \
  --public-ip-address myBastionIP \
  --vnet-name myVNet \
  --location eastus
```

It can take a few minutes for the Azure Bastion host to deploy.

Create backend servers

In this section, you create:

- Two network interfaces for the virtual machines
- Two virtual machines to be used as backend servers for the load balancer

Create network interfaces for the virtual machines

Create two network interfaces with az network nic create:

Azure CLI

```
array=(myNicVM1 myNicVM2)
for vmnic in "${array[@]}"
do
  az network nic create \
      --resource-group CreatePubLBQS-rg \
      --name $vmnic \
      --vnet-name myVNet \
      --subnet myBackEndSubnet \
      --network-security-group myNSG
done
```

Create virtual machines

Create the virtual machines with az vm create:

Azure CLI

```
az vm create \
  --resource-group CreatePubLBQS-rg \
  --name myVM1 \
  --nics myNicVM1 \
  --image win2019datacenter \
```

```
    --admin-username azureuser \
    --zone 1 \
    --no-wait
```

Azure CLI

```
az vm create \
  --resource-group CreatePubLBQS-rg \
  --name myVM2 \
  --nics myNicVM2 \
  --image win2019datacenter \
  --admin-username azureuser \
  --zone 2 \
  --no-wait
```

It can take a few minutes for the VMs to deploy. You can continue to the next steps while the VMs are creating.

> **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Add virtual machines to load balancer backend pool

Add the virtual machines to the backend pool with az network nic ip-config address-pool add:

Azure CLI

```
array=(myNicVM1 myNicVM2)
for vmnic in "${array[@]}"
do
  az network nic ip-config address-pool add \
   --address-pool myBackendPool \
   --ip-config-name ipconfig1 \
   --nic-name $vmnic \
   --resource-group CreatePubLBQS-rg \
   --lb-name myLoadBalancer
done
```

Create NAT gateway

To provide outbound internet access for resources in the backend pool, create a NAT gateway.

Create public IP

Use az network public-ip create to create a single IP for the outbound connectivity.

Azure CLI

```
az network public-ip create \
  --resource-group CreatePubLBQS-rg \
  --name myNATgatewayIP \
  --sku Standard \
  --zone 1 2 3
```

To create a zonal redundant public IP address in Zone 1 instead, use the following command:

Azure CLI

```
az network public-ip create \
  --resource-group CreatePubLBQS-rg \
  --name myNATgatewayIP \
  --sku Standard \
  --zone 1
```

Create NAT gateway resource

Use az network nat gateway create to create the NAT gateway resource. The public IP created in the previous step is associated with the NAT gateway.

Azure CLI

```
az network nat gateway create \
  --resource-group CreatePubLBQS-rg \
```

```
    --name myNATgateway \
    --public-ip-addresses myNATgatewayIP \
    --idle-timeout 10
```

Associate NAT gateway with subnet

Configure the source subnet in virtual network to use a specific NAT gateway resource with az network vnet subnet update.

Azure CLI

```
  az network vnet subnet update \
    --resource-group CreatePubLBQS-rg \
    --vnet-name myVNet \
    --name myBackendSubnet \
    --nat-gateway myNATgateway
```

Install IIS

Use az vm extension set to install IIS on the virtual machines and set the default website to the computer name.

Azure CLI

```
  array=(myVM1 myVM2)
    for vm in "${array[@]}"
    do
     az vm extension set \
       --publisher Microsoft.Compute \
       --version 1.8 \
       --name CustomScriptExtension \
       --vm-name $vm \
       --resource-group CreatePubLBQS-rg \
       --settings '{"commandToExecute":"powershell Add-WindowsFeature Web-Server;
powershell Add-Content -Path \"C:\\inetpub\\wwwroot\\Default.htm\" -Value
$($env:computername)"}'
  done
```

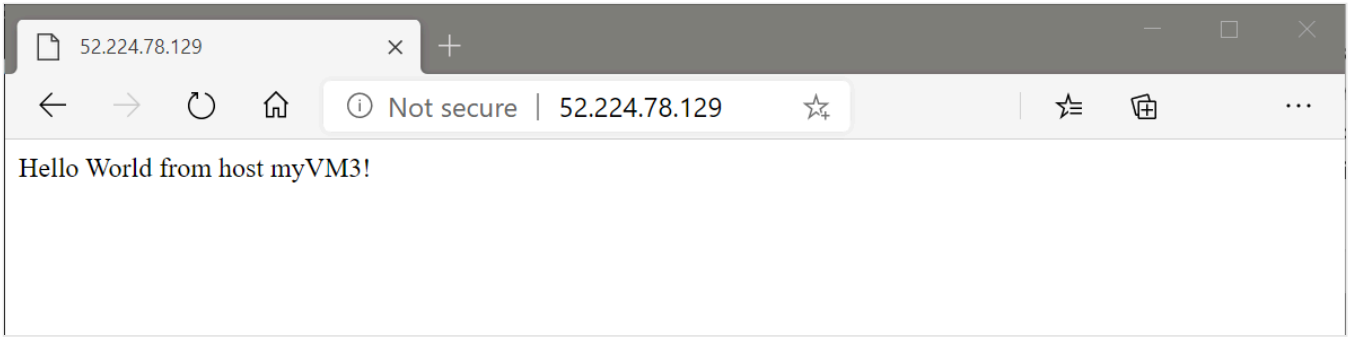
Test the load balancer

To get the public IP address of the load balancer, use az network public-ip show.

Copy the public IP address, and then paste it into the address bar of your browser.

Azure CLI

```
az network public-ip show \
  --resource-group CreatePubLBQS-rg \
  --name myPublicIP \
  --query ipAddress \
  --output tsv
```



Clean up resources

When no longer needed, use the az group delete command to remove the resource group, load balancer, and all related resources.

Azure CLI

```
az group delete \
  --name CreatePubLBQS-rg
```

Next steps

In this quickstart:

- You created a standard public load balancer
- Attached two virtual machines
- Configured the load balancer traffic rule and health probe
- Tested the load balancer

To learn more about Azure Load Balancer, continue to:

What is Azure Load Balancer?

Last updated on 01/28/2026

Tutorial: Create and manage a Virtual Machine Scale Set with the Azure CLI

Article • 08/22/2024

A Virtual Machine Scale Set allows you to deploy and manage a set of virtual machines. Throughout the lifecycle of a Virtual Machine Scale Set, you may need to run one or more management tasks. In this tutorial you learn how to:

- ✓ Create a resource group
- ✓ Create a Virtual Machine Scale Set
- ✓ Scale out and in
- ✓ Stop, Start and restart VM instances

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

This article requires version 2.0.29 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a Virtual Machine Scale Set. Create a resource group with the az group create command. In this example, a resource group named *myResourceGroup* is created in the *eastus* region.

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

The resource group name is specified when you create or modify a scale set throughout this tutorial.

Create a scale set

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

You create a Virtual Machine Scale Set with the az vmss create command. The following example creates a scale set named *myScaleSet*, and generates SSH keys if they don't exist:

Azure CLI

```
az vmss create \
  --resource-group myResourceGroup \
  --name myScaleSet \
  --orchestration-mode flexible \
  --image <SKU image> \
  --admin-username azureuser \
  --generate-ssh-keys
```

It takes a few minutes to create and configure all the scale set resources and VM instances. To distribute traffic to the individual VM instances, a load balancer is also created.

View information about the VM instances in your scale set

To view a list of VM instances in a scale set, use az vm list as follows:

Azure CLI

```
az vm list --resource-group myResourceGroup --output table
```

The following example output shows two VM instances in the scale set:

Output

```
Name                 ResourceGroup    Location    Zones
-------------------  ---------------  ----------  -------
myScaleSet_instance1  myResourceGroup  eastus
myScaleSet_instance2  myResourceGroup  eastus
```

To see additional information about a specific VM instance, use az vm show and specify the VM name.

Azure CLI

```
az vm show --resource-group myResourceGroup --name myScaleSet_instance1
```

Output

```
{
  "hardwareProfile": {
    "vmSize": "Standard_DS1_v2",
  },
  "id":
"/subscriptions/resourceGroups/myResourceGroup/providers/Microsoft.Compute/v
irtualMachines/myScaleSet_instance1",
  "location": "eastus",
  "name": "myScaleSet_instance1",
  "networkProfile": {
    "networkInterfaces": [
      {
        "deleteOption": "Delete",
        "id":
"/subscriptions/resourceGroups/myResourceGroup/providers/Microsoft.Network/n
etworkInterfaces/mysca2215Nic-0396c71c",
        "primary": true,
        "resourceGroup": "myResourceGroup"
      }
    ]
```

```
  },
  "osProfile": {
    "adminUsername": "azureuser",
    "allowExtensionOperations": true,
    "computerName": "myScaleSN30BP1",
    "linuxConfiguration": {
      "disablePasswordAuthentication": true,
      "enableVmAgentPlatformUpdates": false,
      "patchSettings": {
        "assessmentMode": "ImageDefault",
        "patchMode": "ImageDefault"
      },
      "provisionVmAgent": true,
      "ssh": {
        "publicKeys": [
          {
            "keyData": "ssh-rsa",
            "path": "/home/azureuser/.ssh/authorized_keys"
          }
        ]
      }
    },
    "requireGuestProvisionSignal": true,
    "secrets": [],
  },
  "provisioningState": "Succeeded",
  "resourceGroup": "myResourceGroup",
  "storageProfile": {
    "dataDisks": [],
    "imageReference": {
      "exactVersion": "XXXXX",
      "offer": "myOffer",
      "publisher": "myPublisher",
      "sku": "mySKU",
      "version": "latest"
    },
    "osDisk": {
      "caching": "ReadWrite",
      "createOption": "FromImage",
      "deleteOption": "Delete",
      "diskSizeGb": 30,
      "managedDisk": {
        "id":
"/subscriptions/resourceGroups/myResourceGroup/providers/Microsoft.Compute/d
isks/myScaleSet_instance1_disk1",
        "resourceGroup": "myResourceGroup",
        "storageAccountType": "Premium_LRS"
      },
      "name": "myScaleSet_instance1_disk1",
      "osType": "Linux",
    }
  },
  "tags": {},
  "timeCreated": "2022-11-16T20:32:15.024581+00:00",
  "type": "Microsoft.Compute/virtualMachines",
```

```
  "virtualMachineScaleSet": {
    "id":
"/subscriptions/resourceGroups/myResourceGroup/providers/Microsoft.Compute/v
irtualMachineScaleSets/myScaleSet",
    "resourceGroup": "myResourceGroup"
  },
}
```

Create a scale set with a specific VM instance size

When you created a scale set at the start of the tutorial, a default VM SKU of *Standard_D1_v2* was provided for the VM instances. You can specify a different VM instance size based on the output from az vm list-sizes. The following example would create a scale set with the `--vm-sku` parameter to specify a VM instance size of *Standard_F1*. As it takes a few minutes to create and configure all the scale set resources and VM instances, you don't have to deploy the following scale set:

Azure CLI

```azurecli
az vmss create \
  --resource-group myResourceGroup \
  --name myScaleSet \
  --orchestration-mode flexible \
  --image <SKU image> \
  --vm-sku Standard_F1 \
  --admin-user azureuser \
  --generate-ssh-keys
```

Change the capacity of a scale set

When you created a scale set at the start of the tutorial, two VM instances were deployed by default. You can specify the `--instance-count` parameter with az vmss create to change the number of instances created with a scale set. To increase or decrease the number of VM instances in your existing scale set, you can manually change the capacity. The scale set creates or removes the required number of VM instances, then configures the load balancer to distribute traffic.

To manually increase or decrease the number of VM instances in the scale set, use az vmss scale. The following example sets the number of VM instances in your scale set to *3*:

Azure CLI

```
az vmss scale \
  --resource-group myResourceGroup \
  --name myScaleSet \
  --new-capacity 3
```

It takes a few minutes to update the capacity of your scale set. To see the number of instances you now have in the scale set, use az vm list and query on the associated resource group.

Azure CLI

```
az vm list --resource-group myResourceGroup --output table
```

Output

```
Name                 ResourceGroup    Location    Zones
-------------------  ---------------  ----------  -------
myScaleSet_instance1  myResourceGroup  eastus
myScaleSet_instance2  myResourceGroup  eastus
myScaleSet_instance3  myResourceGroup  eastus
```

Stop and deallocate VM instances in a scale set

To stop all the VM instances in a scale set, use az vmss stop.

Azure CLI

```
az vmss stop \
  --resource-group myResourceGroup \
  --name myScaleSet
```

To stop individual VM instances in a scale set, use az vm stop and specify the instance name.

Azure CLI

```
az vm stop \
  --resource-group myResourceGroup \
  --name myScaleSet_instance1
```

Stopped VM instances remain allocated and continue to incur compute charges. If you instead wish the VM instances to be deallocated and only incur storage charges, use az vm deallocate and specify the instance names you want deallocated.

Azure CLI

```
az vm deallocate \
  --resource-group myResourceGroup \
  --name myScaleSet_instance1
```

Start VM instances in a scale set

To start all the VM instances in a scale set, use az vmss start.

Azure CLI

```
az vmss start \
  --resource-group myResourceGroup \
  --name myScaleSet
```

To start individual VM instances in a scale set, use az vm start and specify the instance name.

Azure CLI

```
az vm start \
  --resource-group myResourceGroup \
  --name myScaleSet_instance1
```

Restart VM instances in a scale set

To restart all the VM instances in a scale set, use az vmss restart.

Azure CLI

```
az vmss restart \
  --resource-group myResourceGroup \
  --name myScaleSet
```

To restart individual VM instances in a scale set, use az vm restart and specify the instance name.

Azure CLI

```
az vm restart \
  --resource-group myResourceGroup \
  --name myScaleSet_instance1
```

Clean up resources

When you delete a resource group, all resources contained within, such as the VM instances, virtual network, and disks, are also deleted. The `--no-wait` parameter returns control to the prompt without waiting for the operation to complete. The `--yes` parameter confirms that you wish to delete the resources without an extra prompt to do so.

Azure CLI

```
az group delete --name myResourceGroup --no-wait --yes
```

Next steps

In this tutorial, you learned how to perform some basic scale set creation and management tasks with the Azure CLI:

- ✓ Create a resource group
- ✓ Create a scale set
- ✓ View and use specific VM sizes
- ✓ Manually scale a scale set
- ✓ Perform common scale set management tasks such as stopping, starting and restarting your scale set

Advance to the next tutorial to learn how to connect to your scale set instances.

Use data disks with scale sets

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Create and Manage Windows VMs with Azure PowerShell

Article • 10/28/2024

Applies to: ✔ Windows VMs

Azure virtual machines provide a fully configurable and flexible computing environment. This tutorial covers basic Azure virtual machine (VM) deployment tasks like selecting a VM size, selecting a VM image, and deploying a VM. You learn how to:

- ✔ Create and connect to a VM
- ✔ Select and use VM images
- ✔ View and use specific VM sizes
- ✔ Resize a VM
- ✔ View and understand VM state

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create resource group

Create a resource group with the New-AzResourceGroup command.

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine. In the following example, a resource group named *myResourceGroupVM* is created in the *EastUS* region:

Azure PowerShell

```
New-AzResourceGroup `
   -ResourceGroupName "myResourceGroupVM" `
   -Location "EastUS"
```

The resource group is specified when creating or modifying a VM, which can be seen throughout this tutorial.

Create a VM

When creating a VM, several options are available like operating system image, network configuration, and administrative credentials. This example creates a VM named *myVM*, running the default version of Windows Server 2016 Datacenter.

Set the username and password needed for the administrator account on the VM with Get-Credential:

Azure PowerShell

```
$cred = Get-Credential
```

Create the VM with New-AzVM.

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName "myResourceGroupVM" `
    -Name "myVM" `
    -Location "EastUS" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress" `
    -Credential $cred
```

Connect to VM

After the deployment has completed, create a remote desktop connection with the VM.

Run the following commands to return the public IP address of the VM. Take note of this IP Address so you can connect to it with your browser to test web connectivity in a future step.

Azure PowerShell

```
Get-AzPublicIpAddress `
   -ResourceGroupName "myResourceGroupVM"  | Select IpAddress
```

Use the following command, on your local machine, to create a remote desktop session with the VM. Replace the IP address with the *publicIPAddress* of your VM. When prompted, enter the credentials used when creating the VM.

PowerShell

```
mstsc /v:<publicIpAddress>
```

In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username and password you created for the VM and then click **OK**.

Understand marketplace images

The Azure marketplace includes many images that can be used to create a new VM. In the previous steps, a VM was created using the Windows Server 2016 Datacenter image. In this step, the PowerShell module is used to search the marketplace for other Windows images, which can also be used as a base for new VMs. This process consists of finding the publisher, offer, SKU, and optionally a version number to identify the image.

Use the Get-AzVMImagePublisher command to return a list of image publishers:

Azure PowerShell

```
Get-AzVMImagePublisher -Location "EastUS"
```

Use the Get-AzVMImageOffer to return a list of image offers. With this command, the returned list is filtered on the specified publisher named `MicrosoftWindowsServer`:

Azure PowerShell

```
Get-AzVMImageOffer `
   -Location "EastUS" `
   -PublisherName "MicrosoftWindowsServer"
```

The results will look something like this example:

PowerShell

```
Offer             PublisherName          Location
-----             -------------          --------
Windows-HUB       MicrosoftWindowsServer EastUS
WindowsServer     MicrosoftWindowsServer EastUS
WindowsServer-HUB MicrosoftWindowsServer EastUS
```

The Get-AzVMImageSku command will then filter on the publisher and offer name to return a list of image names.

Azure PowerShell

```
Get-AzVMImageSku `
   -Location "EastUS" `
   -PublisherName "MicrosoftWindowsServer" `
   -Offer "WindowsServer"
```

The results will look something like this example:

PowerShell

```
Skus                                      Offer         PublisherName
Location
----                                      -----         -------------
--------
2008-R2-SP1                               WindowsServer
MicrosoftWindowsServer EastUS
2008-R2-SP1-smalldisk                     WindowsServer
MicrosoftWindowsServer EastUS
2012-Datacenter                           WindowsServer
MicrosoftWindowsServer EastUS
2012-Datacenter-smalldisk                 WindowsServer
MicrosoftWindowsServer EastUS
2012-R2-Datacenter                        WindowsServer
MicrosoftWindowsServer EastUS
2012-R2-Datacenter-smalldisk              WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter                           WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter-Server-Core               WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter-Server-Core-smalldisk     WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter-smalldisk                 WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter-with-Containers           WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter-with-Containers-smalldisk WindowsServer
MicrosoftWindowsServer EastUS
2016-Datacenter-with-RDSH                 WindowsServer
MicrosoftWindowsServer EastUS
2016-Nano-Server                          WindowsServer
MicrosoftWindowsServer EastUS
```

This information can be used to deploy a VM with a specific image. This example deploys a VM using the latest version of a Windows Server 2016 with Containers image.

```
```

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName "myResourceGroupVM" `
    -Name "myVM2" `
    -Location "EastUS" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress2" `
    -ImageName "MicrosoftWindowsServer:WindowsServer:2016-Datacenter-with-
Containers:latest" `
    -Credential $cred
```

Understand VM sizes

The VM size determines the amount of compute resources like CPU, GPU, and memory that are made available to the VM. Virtual machines should be created using a VM size appropriate for the workload. If a workload increases, an existing virtual machine can also be resized.

VM Sizes

The following table categorizes sizes into use cases.

Expand table

Type	Common sizes	Description
General purpose	B, Dsv3, Dv3, DSv2, Dv2, Av2, DC	Balanced CPU-to-memory. Ideal for dev / test and small to medium applications and data solutions.
Compute optimized	Fsv2	High CPU-to-memory. Good for medium traffic applications, network appliances, and batch processes.
Memory optimized	Esv3, Ev3, M, DSv2, Dv2	High memory-to-core. Great for relational databases, medium to large caches, and in-memory analytics.
Storage optimized	Lsv2, Ls	High disk throughput and IO. Ideal for Big Data, SQL, and NoSQL databases.
GPU	NV, NVv2, NC, NCv2, NCv3, ND	Specialized VMs targeted for heavy graphic rendering and video editing.
High performance	H	Our most powerful CPU VMs with optional high-throughput network interfaces (RDMA).

Find available VM sizes

To see a list of VM sizes available in a particular region, use the Get-AzVMSize command.

Azure PowerShell

```
Get-AzVMSize -Location "EastUS"
```

Resize a VM

After a VM has been deployed, it can be resized to increase or decrease resource allocation.

Before resizing a VM, check if the size you want is available on the current VM cluster. The Get-AzVMSize command returns a list of sizes.

Azure PowerShell

```
Get-AzVMSize -ResourceGroupName "myResourceGroupVM" -VMName "myVM"
```

If the size is available, the VM can be resized from a powered-on state, however it is rebooted during the operation.

Azure PowerShell

```
$vm = Get-AzVM `
   -ResourceGroupName "myResourceGroupVM"  `
   -VMName "myVM"
$vm.HardwareProfile.VmSize = "Standard_DS3_v2"
Update-AzVM `
   -VM $vm `
   -ResourceGroupName "myResourceGroupVM"
```

If the size you want isn't available on the current cluster, the VM needs to be deallocated before the resize operation can occur. Deallocating a VM will remove any data on the temp disk, and the public IP address will change unless a static IP address is being used.

Azure PowerShell

```
Stop-AzVM `
   -ResourceGroupName "myResourceGroupVM" `
   -Name "myVM" -Force
```

```
$vm = Get-AzVM `
   -ResourceGroupName "myResourceGroupVM"  `
   -VMName "myVM"
$vm.HardwareProfile.VmSize = "Standard_E2s_v3"
Update-AzVM -VM $vm `
   -ResourceGroupName "myResourceGroupVM"
Start-AzVM `
   -ResourceGroupName "myResourceGroupVM"  `
   -Name $vm.name
```

VM power states

An Azure VM can have one of many power states.

Expand table

Power State	Description
Starting	The virtual machine is being started.
Running	The virtual machine is running.
Stopping	The virtual machine is being stopped.
Stopped	The VM is stopped. Virtual machines in the stopped state still incur compute charges.
Deallocating	The VM is being deallocated.
Deallocated	Indicates that the VM is removed from the hypervisor but is still available in the control plane. Virtual machines in the `Deallocated` state do not incur compute charges.
-	The power state of the VM is unknown.

To get the state of a particular VM, use the Get-AzVM command. Be sure to specify a valid name for a VM and resource group.

Azure PowerShell

```
Get-AzVM `
    -ResourceGroupName "myResourceGroupVM" `
    -Name "myVM" `
    -Status | Select @{n="Status"; e={$_.Statuses[1].Code}}
```

The output will look something like this example:

PowerShell

```
Status
------
PowerState/running
```

To retrieve the power state of all the VMs in your subscription, use the Virtual Machines - List All API with parameter **statusOnly** set to *true*.

Management tasks

During the lifecycle of a VM, you may want to run management tasks like starting, stopping, or deleting a VM. Additionally, you may want to create scripts to automate repetitive or complex tasks. Using Azure PowerShell, many common management tasks can be run from the command line or in scripts.

Stop a VM

Stop and deallocate a VM with Stop-AzVM:

Azure PowerShell

```
Stop-AzVM `
   -ResourceGroupName "myResourceGroupVM" `
   -Name "myVM" -Force
```

If you want to keep the VM in a provisioned state, use the -StayProvisioned parameter.

Start a VM

Azure PowerShell

```
Start-AzVM `
   -ResourceGroupName "myResourceGroupVM" `
   -Name "myVM"
```

Deleting VM resources

You can delete a VM, but by default this only deletes the VM resource, not the disks and networking resources the VM uses. You can change the default behavior to delete other resources when you delete the VM. For more information, see Delete a VM and attached resources.

Next steps

In this tutorial, you learned about basic VM creation and management such as how to:

- ✓ Create and connect to a VM
- ✓ Select and use VM images
- ✓ View and use specific VM sizes
- ✓ Resize a VM
- ✓ View and understand VM state

Advance to the next tutorial to learn about VM disks.

Create and Manage VM disks

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Manage disks with Azure PowerShell

Article • 12/06/2024

Azure virtual machines (VMs) use disks to store operating systems (OS), applications, and data. When you create a VM, it's important to choose an appropriate disk size and configuration for the expected workload.

This tutorial covers deployment and management of VM disks. In this tutorial, you learn how to:

- ✓ Create, attach, and initialize a data disk
- ✓ Verify a disk's status
- ✓ Initialize a disk
- ✓ Expand and upgrade a disk
- ✓ Detach and delete a disk

Prerequisites

You must have an Azure account with an active subscription. If you don't have an Azure subscription, create a free account before you begin.

Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article, without having to install anything on your local environment.

To start Azure Cloud Shell:

Expand table

Option	Example/Link
Select **Try It** in the upper-right corner of a code or command block. Selecting **Try It** doesn't automatically copy the code or command to Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	Launch Cloud Shell

Option	Example/Link
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To use Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block (or command block) to copy the code or command.
3. Paste the code or command into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux, or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code or command.

Create a VM

The exercises in this tutorial require a VM. Follow the steps in this section to create one.

Before you begin, find the `$azRegion` variable located in the first line of sample code and update the value to reflect your desired region. For example, to specify the **Central US** region, use `$azRegion = "Central US"`. Next, use the code to deploy a VM within a new resource group. You're prompted for username and password values for the VM's local administrator account.

Azure PowerShell

```
$azRegion = "[Your Region]"
$azResourceGroup = "myDemoResourceGroup"
$azVMName = "myDemoVM"
$azDataDiskName = "myDemoDataDisk"

New-AzVm `
    -Location $azRegion `
    -ResourceGroupName $azResourceGroup `
    -Name $azVMName `
    -Size "Standard_D2s_v3" `
    -VirtualNetworkName "myDemoVnet" `
    -SubnetName "myDemoSubnet" `
    -SecurityGroupName "myDemoNetworkSecurityGroup" `
    -PublicIpAddressName "myDemoPublicIpAddress"
```

The output confirms the VM's successful creation.

Output

```
ResourceGroupName        : myDemoResourceGroup
Id                       :
/subscriptions/{GUID}/resourceGroups/myDemoResourceGroup/providers/Microsoft
.Compute/virtualMachines/myDemoTestVM
VmId                     : [{GUID}]
Name                     : myDemoVM
Type                     : Microsoft.Compute/virtualMachines
Location                 : centralus
Tags                     : {}
HardwareProfile          : {VmSize}
NetworkProfile           : {NetworkInterfaces}
OSProfile                : {ComputerName, AdminUsername,
WindowsConfiguration, AllowExtensionOperations, RequireGuestProvisionSignal}
ProvisioningState        : Succeeded
StorageProfile           : {ImageReference, OsDisk, DataDisks}
FullyQualifiedDomainName : mydemovm-abc123.Central US.cloudapp.azure.com
```

The VM is provisioned, and two disks are automatically created and attached.

- An **operating system disk**, which hosts the virtual machine's operating system.
- A **temporary disk**, which is primarily used for operations such as temporary data processing.

Add a data disk

We recommend that you separate application and user data from OS-related data when possible. If you need to store user or application data on your VM, you'll typically create and attach additional data disks.

Follow the steps in this section to create, attach, and initialize a data disk on the VM.

Create the data disk

This section guides you through the creation of a data disk.

1. Before a data disk can be created, you must first create a disk object. The following code sample uses the New-AzDiskConfig cmdlet to configure a disk object.

   Azure PowerShell

   ```
   $diskConfig = New-AzDiskConfig `
       -Location $azRegion `
       -CreateOption Empty `
   ```

```
    -DiskSizeGB 128 `
    -SkuName "Standard_LRS"
```

2. After the disk object is created, use the New-AzDisk cmdlet to provision a data disk.

Azure PowerShell

```
$dataDisk = New-AzDisk `
    -ResourceGroupName $azResourceGroup `
    -DiskName $azDataDiskName `
    -Disk $diskConfig
```

You can use the Get-AzDisk cmdlet to verify that the disk was created.

Azure PowerShell

```
Get-AzDisk `
    -ResourceGroupName $azResourceGroup `
    -DiskName $azDataDiskName
```

In this example, the output confirms that the disk was created. The `DiskState` and `ManagedBy` property values confirm that the disk is not yet attached.

Output

```
ResourceGroupName             : myDemoResourceGroup
ManagedBy                     :
ManagedByExtended             : {}
OsType                        :
DiskSizeGB                    : 128
DiskSizeBytes                 : 137438953472
ProvisioningState             : Succeeded
DiskIOPSReadWrite             : 500
DiskMBpsReadWrite             : 60
DiskState                     : Unattached
Name                          : myDemoDataDisk
```

Attach the data disk

A data disk must be attached to a VM before the VM can access it. Complete the steps in this section to create a reference for the VM, connect the disk, and update the VM's configuration.

1. Get the VM to which you'll attach the data disk. The following sample code uses the Get-AzVM cmdlet to create a reference to the VM.

Azure PowerShell

```
$vm = Get-AzVM `
    -ResourceGroupName $azResourceGroup `
    -Name $azVMName
```

2. Next, attach the data disk to the VM's configuration with the Add-AzVMDataDisk cmdlet.

Azure PowerShell

```
$vm = Add-AzVMDataDisk `
    -VM $vm `
    -Name $azDataDiskName `
    -CreateOption Attach `
    -ManagedDiskId $dataDisk.Id `
    -Lun 1
```

3. Finally, update the VM's configuration with the Update-AzVM cmdlet.

Azure PowerShell

```
Update-AzVM `
    -ResourceGroupName $azResourceGroup `
    -VM $vm
```

After a brief pause, the output confirms a successful attachment.

Output

```
RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

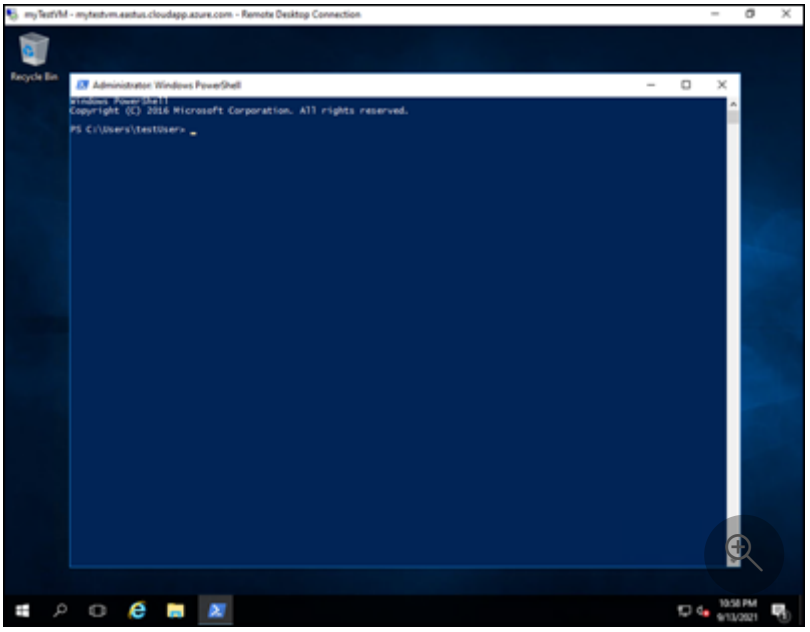
Initialize the data disk

After a data disk is attached to the VM, the OS needs to be configured to use the disk. The following section provides guidance on how to connect to the remote VM and configure the first disk added.

1. Sign in to the Azure portal.

2. Locate the VM to which you've attached the data disk. Create a Remote Desktop Protocol (RDP) connection and sign in as the local administrator.

3. After you establish an RDP connection to the remote VM, select the Windows **Start** menu. Enter **PowerShell** in the search box and select **Windows PowerShell** to open a PowerShell window.



4. In the open PowerShell window, run the following script.

PowerShell

```
Get-Disk | Where PartitionStyle -eq 'raw' |
    Initialize-Disk -PartitionStyle MBR -PassThru |
    New-Partition -AssignDriveLetter -UseMaximumSize |
    Format-Volume -FileSystem NTFS -NewFileSystemLabel "myDemoDataDisk"
-Confirm:$false
```

The output confirms a successful initialization.

Output

```
DriveLetter FileSystemLabel FileSystem DriveType HealthStatus
OperationalStatus SizeRemaining   Size
----------- --------------- ---------- --------- ------------ ---------
-------- -------------   ----
F           myDemoDataDisk  NTFS       Fixed     Healthy      OK
127.89 GB 128 GB
```

Expand a disk

You can expand Azure disks to provide extra storage capacity when your VM is low on available disk space.

Some scenarios require data to be stored on the OS disk. For example, you may be required to support legacy applications that install components on the OS drive. You may also have the need to migrate an on-premises physical PC or VM with a larger OS drive. In such cases, it may become necessary to expand a VM's OS disk.

Shrinking an existing disk isn't supported, and can potentially result in data loss.

Update the disk's size

Follow the steps below to resize either the OS disk or a data disk.

> **ⓘ Important**
>
> If your disk meets the requirements in **Expand without downtime**, you can skip steps 2 and 6.
>
> Shrinking an existing disk isn't supported and may result in data loss.
>
> After expanding the disks, you need to expand the volume in the operating system to take advantage of the larger disk.

1. Select the VM that contains the disk that you'll resize with the `Get-AzVM` cmdlet.

   Azure PowerShell

   ```
   $vm = Get-AzVM `
     -ResourceGroupName $azResourceGroup `
     -Name $azVMName
   ```

2. Before you can resize a VM's disk, you must stop the VM. Use the `Stop-AzVM` cmdlet to stop the VM. You'll be prompted for confirmation.

   > **ⓘ Important**
   >
   > Before you initiate a VM shutdown, always confirm that there are no important resources or data that could be lost.

   Azure PowerShell

   ```
   Stop-AzVM `
       -ResourceGroupName $azResourceGroup `
       -Name $azVMName
   ```

After a short pause, the output confirms that the machine is successfully stopped.

Output

```
OperationId : abcd1234-ab12-cd34-123456abcdef
Status      : Succeeded
StartTime   : 9/13/2021 7:10:23 PM
EndTime     : 9/13/2021 7:11:12 PM
Error       :
```

3. After the VM is stopped, get a reference to either the OS or data disk attached to the VM with the `Get-AzDisk` cmdlet.

   The following example selects the VM's OS disk.

   Azure PowerShell

   ```
   $disk= Get-AzDisk `
       -ResourceGroupName $azResourceGroup `
       -DiskName $vm.StorageProfile.OsDisk.Name
   ```

   The following example selects the VM's first data disk.

   Azure PowerShell

   ```
       $disk= Get-AzDisk `
           -ResourceGroupName $azResourceGroup `
           -DiskName $vm.StorageProfile.DataDisks[0].Name
   ```

4. Now that you have a reference to the disk, set the size of the disk to 250 GiB.

   > **ⓘ Important**
   >
   > The new size should be greater than the existing disk size. The maximum allowed is 4,095 GiB for OS disks.

   Azure PowerShell

   ```
   $disk.DiskSizeGB = 250
   ```

5. Next, update the disk image with the `Update-AzDisk` cmdlet.

   Azure PowerShell

```
Update-AzDisk `
    -ResourceGroupName $azResourceGroup `
    -Disk $disk -DiskName $disk.Name
```

The disk image is updated, and the output confirms the disk's new size.

Output

```
ResourceGroupName             : myDemoResourceGroup
ManagedBy                     :
/subscriptions/{GUID}/resourceGroups/myDemoResourceGroup/providers/Micr
osoft.Compute/virtualMachines/myDemoVM
Sku                           :
Microsoft.Azure.Management.Compute.Models.DiskSku
TimeCreated                   : 9/135/2021 6:41:10 PM
CreationData                  :
Microsoft.Azure.Management.Compute.Models.CreationData
DiskSizeGB                    : 250
DiskSizeBytes                 : 268435456000
UniqueId                      : {GUID}
ProvisioningState             : Succeeded
DiskIOPSReadWrite             : 500
DiskMBpsReadWrite             : 60
DiskState                     : Reserved
Encryption                    :
Microsoft.Azure.Management.Compute.Models.Encryption
Id                            :
/subscriptions/{GUID}/resourceGroups/myDemoResourceGroup/providers/Micr
osoft.Compute/disks/myDemoDataDisk
Name                          : myDemoDataDisk
Type                          : Microsoft.Compute/disks
Location                      : centralus
```

6. Finally, restart the VM with the `Start-AzVM` cmdlet.

Azure PowerShell

```
Start-AzVM `
    -ResourceGroupName $azResourceGroup `
    -Name $azVMName
```

After a short pause, the output confirms that the machine is successfully started.

Output

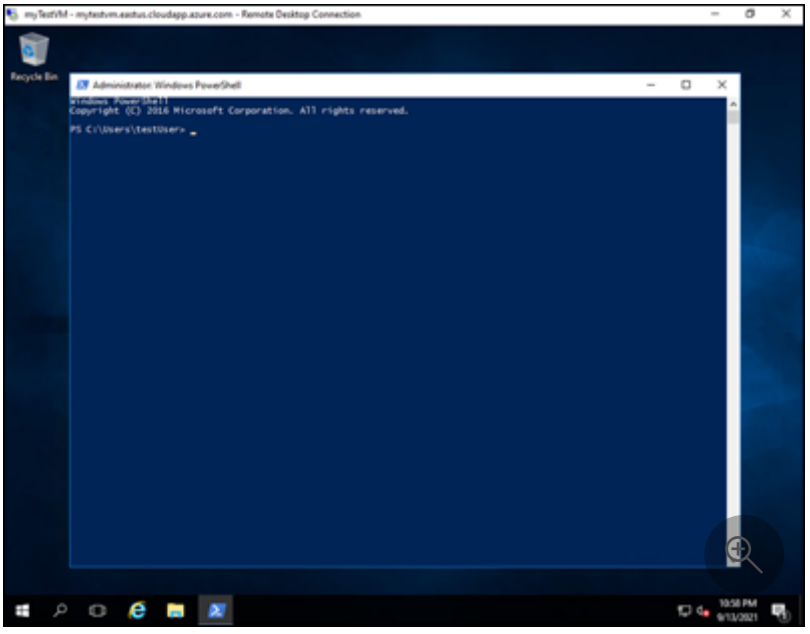
```
OperationId : abcd1234-ab12-cd34-123456abcdef
Status      : Succeeded
StartTime   : 9/13/2021 7:44:54 PM
```

```
EndTime      : 9/13/2021 7:45:15 PM
Error        :
```

Expand the disk volume in the OS

Before you can take advantage of the new disk size, you need to expand the volume within the OS. Follow the steps below to expand the disk volume and take advantage of the new disk size.

1. Sign in to the Azure portal.
2. Locate the VM to which you've attached the data disk. Create a Remote Desktop Protocol (RDP) connection and sign in. If you no longer have access to an administrative account, create a credential object for a specified user name and password with the Get-Credential cmdlet.
3. After you've established an RDP connection to the remote VM, select the Windows **Start** menu. Enter **PowerShell** in the search box and select **Windows PowerShell** to open a PowerShell window.



4. Open PowerShell and run the following script. Change the value of the `-DriveLetter` variable as appropriate. For example, to resize the partition on the **F:** drive, use `$driveLetter = "F"`.

PowerShell

```powershell
$driveLetter = "[Drive Letter]"
$size = (Get-PartitionSupportedSize -DriveLetter $driveLetter)
Resize-Partition `
```

```
    -DriveLetter $driveLetter `
    -Size $size.SizeMax
```

5. Minimize the RDP window and switch back to Azure Cloud Shell. Use the `Get-AzDisk` cmdlet to verify that the disk was resized successfully.

Azure PowerShell

```
Get-AzDisk `
    -ResourceGroupName $azResourceGroup | Out-Host -Paging
```

Upgrade a disk

There are several ways to respond to changes in your organization's workloads. For example, you may choose to upgrade a standard HDD to a premium SSD to handle increased demand.

Follow the steps in this section to upgrade a managed disk from standard to premium.

1. Select the VM that contains the disk that you'll upgrade with the `Get-AzVM` cmdlet.

Azure PowerShell

```
$vm = Get-AzVM `
  -ResourceGroupName $azResourceGroup `
  -Name $azVMName
```

2. Before you can upgrade a VM's disk, you must stop the VM. Use the `Stop-AzVM` cmdlet to stop the VM. You'll be prompted for confirmation.

> **ⓘ Important**
>
> Before you initiate a VM shutdown, always confirm that there are no important resources or data that could be lost.

Azure PowerShell

```
Stop-AzVM `
    -ResourceGroupName $azResourceGroup `
    -Name $azVMName
```

After a short pause, the output confirms that the machine is successfully stopped.

Output

```
OperationId : abcd1234-ab12-cd34-123456abcdef
Status      : Succeeded
StartTime   : 9/13/2021 7:10:23 PM
EndTime     : 9/13/2021 7:11:12 PM
Error       :
```

3. After the VM is stopped, get a reference to either the OS or data disk attached to the VM with the `Get-AzDisk` cmdlet.

   The following example selects the VM's OS disk.

Azure PowerShell

```
$disk= Get-AzDisk `
    -ResourceGroupName $azResourceGroup `
    -DiskName $vm.StorageProfile.OsDisk.Name
```

   The following example selects the VM's first data disk.

Azure PowerShell

```
    $disk= Get-AzDisk `
        -ResourceGroupName $azResourceGroup `
        -DiskName $vm.StorageProfile.DataDisks[0].Name
```

4. Now that you have a reference to the disk, set the disk's SKU to **Premium_LRS**.

Azure PowerShell

```
$disk.Sku =
[Microsoft.Azure.Management.Compute.Models.DiskSku]::new('Premium_LRS')
```

5. Next, update the disk image with the `Update-AzDisk` cmdlet.

Azure PowerShell

```
Update-AzDisk `
    -ResourceGroupName $azResourceGroup `
    -Disk $disk -DiskName $disk.Name
```

   The disk image is updated. Use the following example code to validate that the disk's SKU has been upgraded.

Azure PowerShell

```
$disk.Sku.Name
```

The output confirms the disk's new SKU.

Output

```
Premium_LRS
```

6. Finally, restart the VM with the `Start-AzVM` cmdlet.

Azure PowerShell

```
Start-AzVM `
    -ResourceGroupName $azResourceGroup `
    -Name $azVMName
```

After a short pause, the output confirms that the machine is successfully started.

Output

```
OperationId : abcd1234-ab12-cd34-123456abcdef
Status      : Succeeded
StartTime   : 9/13/2021 7:44:54 PM
EndTime     : 9/13/2021 7:45:15 PM
Error       :
```

Detach a data disk

You can detach a data disk from a VM when you want to attach it to a different VM, or when it's no longer needed. By default, detached disks are not deleted to prevent unintentional data loss. A detached disk will continue to incur storage charges until it's deleted.

1. First, select the VM to which the disk is attached with the `Get-AzVM` cmdlet.

Azure PowerShell

```
$vm = Get-AzVM `
   -ResourceGroupName $azResourceGroup `
   -Name $azVMName
```

2. Next, detach the disk from the VM with the `Remove-AzVMDataDisk` cmdlet.

Azure PowerShell

```
Remove-AzVMDataDisk `
    -VM $vm `
    -Name $azDataDiskName
```

3. Update the state of the VM with the `Update-AzVM` cmdlet to remove the data disk.

Azure PowerShell

```
Update-AzVM `
    -ResourceGroupName $azResourceGroup `
    -VM $vm
```

After a short pause, the output confirms that the VM is successfully updated.

Output

```
RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

Delete a data disk

When you delete a VM, data disks attached to the VM remain provisioned and continue to incur charges until they're deleted. This default behavior helps prevent data loss caused by unintentional deletion.

You can use the following sample PowerShell script to delete unattached disks. The retrieval of disks is limited to the **myDemoResourceGroup** because the `-ResourceGroupName` switch is used with the `Get-AzDisk` cmdlet.

Azure PowerShell

```
# Get all disks in resource group $azResourceGroup
$allDisks = Get-AzDisk -ResourceGroupName $azResourceGroup

# Determine the number of disks in the collection
if($allDisks.Count -ne 0) {

    Write-Host "Found $($allDisks.Count) disks."

    # Iterate through the collection
    foreach ($disk in $allDisks) {

        # Use the disk's "ManagedBy" property to determine if it is
```

```
unattached
        if($disk.ManagedBy -eq $null) {

            # Confirm that the disk can be deleted
            Write-Host "Deleting unattached disk $($disk.Name)."
            $confirm = Read-Host "Continue? (Y/N)"
            if ($confirm.ToUpper() -ne 'Y') { break }
            else {

                # Delete the disk
                $disk | Remove-AzDisk -Force
                Write-Host "Unattached disk $($disk.Name) deleted."
            }
        }
    }
}
```

The unattached data disk is deleted as shown by the output.

Output

```
Name      : abcd1234-ab12-cd34-ef56-abcdef123456
StartTime : 9/13/2021 10:14:05 AM
EndTime   : 9/13/2021 10:14:35 AM
Status    : Succeeded
Error     :
```

Clean up resources

When no longer needed, delete the resource group, VM, and all related resources. You can use the following sample PowerShell script to delete the resource group created earlier in this tutorial.

> **⊗ Caution**
>
> Use caution when deleting a resource group. To avoid the loss of important data, always confirm that there are no important resources or data contained within the resource group before it is deleted.

Azure PowerShell

```
    Remove-AzResourceGroup -Name $azResourceGroup
```

You're prompted for confirmation. After a short pause, the `True` response confirms that the **myDemoResourceGroup** is successfully deleted.

Output

```
Confirm
Are you sure you want to remove resource group 'myDemoResourceGroup'
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
True
```

Next steps

In this tutorial, you learned how to:

- ✓ Create, attach, and initialize a data disk
- ✓ Verify a disk's status
- ✓ Initialize a disk
- ✓ Expand and upgrade a disk
- ✓ Detach and delete a disk

Advance to the next tutorial to learn how to automate VM configuration.

Automate VM configuration

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Create Windows VM images with Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

Images can be used to bootstrap deployments and ensure consistency across multiple VMs. In this tutorial, you create your own specialized image of an Azure virtual machine using PowerShell and store it in an Azure Compute Gallery (formerly known as Shared Image Gallery). You learn how to:

- ✔ Create an Azure Compute Gallery
- ✔ Create an image definition
- ✔ Create an image version
- ✔ Create a VM from an image
- ✔ Share a gallery

Before you begin

The steps below detail how to take an existing VM and turn it into a re-usable custom image that you can use to create new VMs.

To complete the example in this tutorial, you must have an existing virtual machine. If needed, you can see the PowerShell quickstart to create a VM to use for this tutorial. When working through the tutorial, replace the resource names where needed.

Overview

an Azure Compute Gallery simplifies custom image sharing across your organization. Custom images are like marketplace images, but you create them yourself. Custom images can be used to bootstrap configurations such as preloading applications, application configurations, and other OS configurations.

The Azure Compute Gallery lets you share your custom VM images with others. Choose which images you want to share, which regions you want to make them available in, and who you want to share them with.

The Azure Compute Gallery feature has multiple resource types:

Expand table

Resource	Description
Image source	This is a resource that can be used to create an **image version** in a gallery. An image source can be an existing Azure VM that is either generalized or specialized, a managed image, a snapshot, or an image version in another gallery.
Gallery	Like the Azure Marketplace, a **gallery** is a repository for managing and sharing images and VM applications, but you control who has access.
Image definition	Image definitions are created within a gallery and carry information about the image and requirements for using it internally. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements. It is a definition of a type of image.
Image version	An **image version** is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an **image version** to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Get the VM

You can see a list of VMs that are available in a resource group using Get-AzVM. Once you know the VM name and what resource group, you can use `Get-AzVM` again to get the VM object and store it in a variable to use later. This example gets a VM named *sourceVM* from the *myResourceGroup* resource group and assigns it to the variable *$sourceVM*.

Azure PowerShell

```
$sourceVM = Get-AzVM `
   -Name sourceVM `
   -ResourceGroupName myResourceGroup
```

Create a resource group

Create a resource group with the New-AzResourceGroup command.

An Azure resource group is a logical container into which Azure resources are deployed and managed. In the following example, a resource group named *myGalleryRG* is created in the *EastUS* region:

Azure PowerShell

```
$resourceGroup = New-AzResourceGroup `
   -Name 'myGalleryRG' `
   -Location 'EastUS'
```

Create a gallery

A gallery is the primary resource used for enabling image sharing. Allowed characters for gallery name are uppercase or lowercase letters, digits, dots, and periods. The gallery name cannot contain dashes. Gallery names must be unique within your subscription.

Create a gallery using New-AzGallery. The following example creates a gallery named *myGallery* in the *myGalleryRG* resource group.

Azure PowerShell

```
$gallery = New-AzGallery `
   -GalleryName 'myGallery' `
   -ResourceGroupName $resourceGroup.ResourceGroupName `
   -Location $resourceGroup.Location `
   -Description 'Azure Compute Gallery for my organization'
```

Create an image definition

Image definitions create a logical grouping for images. They are used to manage information about the image versions that are created within them. Image definition names can be made up of uppercase or lowercase letters, digits, dots, dashes and periods. For more information about the values you can specify for an image definition, see Image definitions.

Create the image definition using New-AzGalleryImageDefinition. In this example, the gallery image is named *myGalleryImage* and is created for a specialized image.

Azure PowerShell

```
$galleryImage = New-AzGalleryImageDefinition `
   -GalleryName $gallery.Name `
   -ResourceGroupName $resourceGroup.ResourceGroupName `
   -Location $gallery.Location `
   -Name 'myImageDefinition' `
   -OsState specialized `
   -OsType Windows `
   -Publisher 'myPublisher' `
   -Offer 'myOffer' `
   -Sku 'mySKU'
```

Create an image version

Create an image version from a VM using New-AzGalleryImageVersion.

Allowed characters for image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion.MinorVersion.Patch*.

In this example, the image version is *1.0.0* and it's replicated to both *East US* and *South Central US* datacenters. When choosing target regions for replication, you need to include the *source* region as a target for replication.

To create an image version from the VM, use `$vm.Id.ToString()` for the `-Source`.

Azure PowerShell

```
   $region1 = @{Name='South Central US';ReplicaCount=1}
   $region2 = @{Name='East US';ReplicaCount=2}
   $targetRegions = @($region1,$region2)

New-AzGalleryImageVersion `
   -GalleryImageDefinitionName $galleryImage.Name`
   -GalleryImageVersionName '1.0.0' `
   -GalleryName $gallery.Name `
   -ResourceGroupName $resourceGroup.ResourceGroupName `
   -Location $resourceGroup.Location `
   -TargetRegion $targetRegions  `
   -Source $sourceVM.Id.ToString() `
   -PublishingProfileEndOfLifeDate '2030-12-01'
```

It can take a while to replicate the image to all of the target regions.

Create a VM

Once you have a specialized image, you can create one or more new VMs. Using the New-AzVM cmdlet. To use the image, use `Set-AzVMSourceImage` and set the `-Id` to the image definition ID ($galleryImage.Id in this case) to always use the latest image version.

Replace resource names as needed in this example.

Azure PowerShell

```
# Create some variables for the new VM.
$resourceGroup = "myResourceGroup"
$location = "South Central US"
$vmName = "mySpecializedVM"

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create the network resources.
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -
AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup -Location
$location `
  -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location
$location `
  -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -
IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name
myNetworkSecurityGroupRuleRDP  -Protocol Tcp `
  -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange
* -DestinationAddressPrefix * `
  -DestinationPortRange 3389 -Access Deny
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -
Location $location `
  -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface -Name $vmName -ResourceGroupName
$resourceGroup -Location $location `
  -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -
NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration using $imageVersion.Id to specify
the image version.
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1_v2 | `
Set-AzVMSourceImage -Id $galleryImage.Id | `
Add-AzVMNetworkInterface -Id $nic.Id

# Create a virtual machine
New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig
```

Share the gallery

We recommend that you share access at the gallery level. Use an email address and the Get-AzADUser cmdlet to get the object ID for the user, then use New-AzRoleAssignment to give them access to the gallery. Replace the example email, alinne_montes@contoso.com in this example, with your own information.

Azure PowerShell

```
# Get the object ID for the user
$user = Get-AzADUser -StartsWith alinne_montes@contoso.com
# Grant access to the user for our gallery
New-AzRoleAssignment `
   -ObjectId $user.Id `
   -RoleDefinitionName Reader `
   -ResourceName $gallery.Name `
   -ResourceType Microsoft.Compute/galleries `
   -ResourceGroupName $resourceGroup.ResourceGroupName
```

Clean up resources

When no longer needed, you can use the Remove-AzResourceGroup cmdlet to remove the resource group, and all related resources:

Azure PowerShell

```
# Delete the gallery
Remove-AzResourceGroup -Name myGalleryRG

# Delete the VM
Remove-AzResourceGroup -Name myResoureceGroup
```

Azure Image Builder

Azure also offers a service, built on Packer, Azure VM Image Builder. Simply describe your customizations in a template, and it will handle the image creation.

Next steps

In this tutorial, you created a specialized VM image. You learned how to:

- ✓ Create an Azure Compute Gallery
- ✓ Create an image definition
- ✓ Create an image version
- ✓ Create a VM from an image

✓ Share a gallery

Advance to the next tutorial to learn about Virtual Machine Scale Sets.

Create a virtual machine scale set

Feedback

Was this page helpful? Yes No

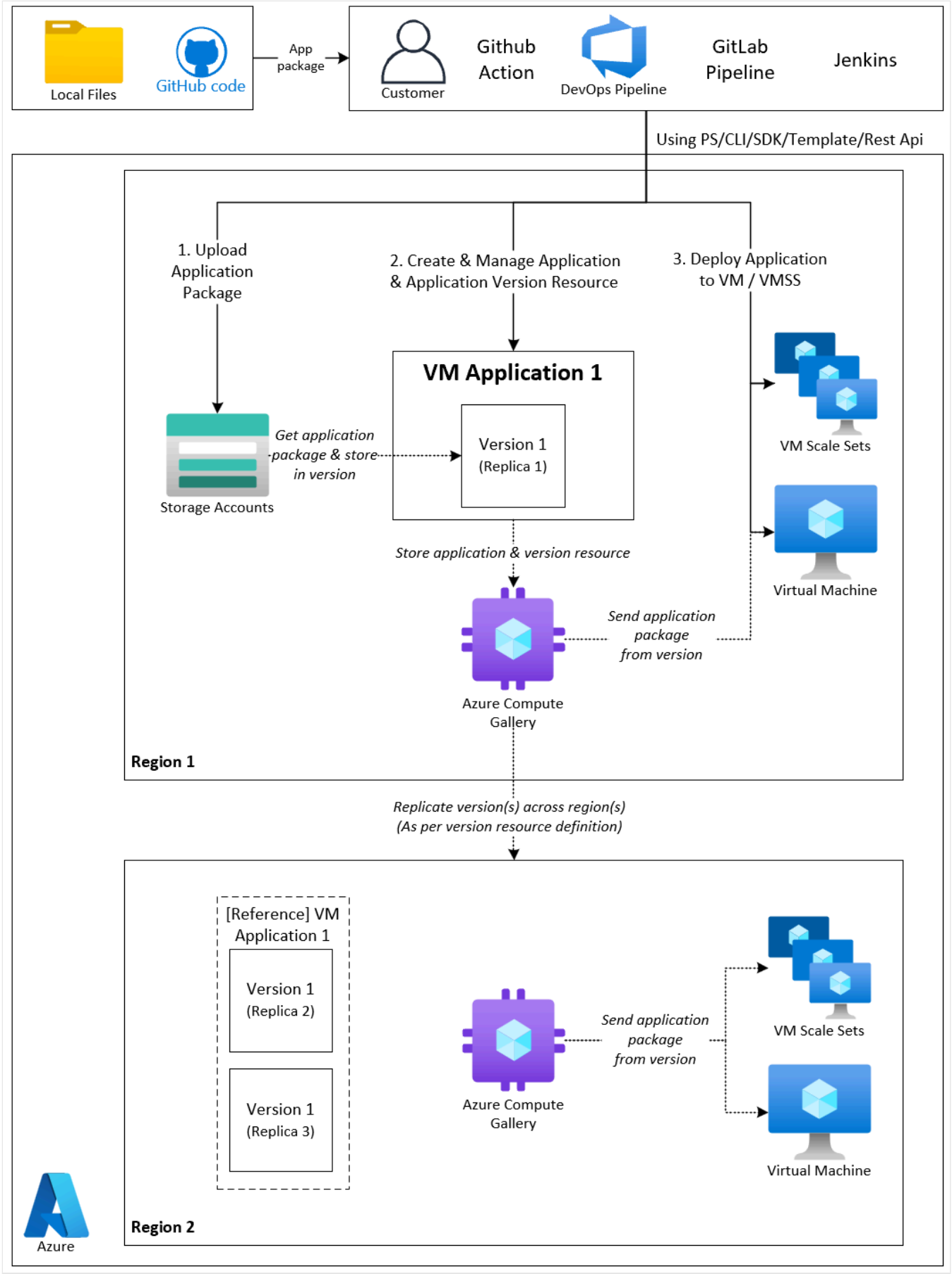
Provide product feedback | Get help at Microsoft Q&A

Create and deploy VM Application

Summarize this article for me

VM Application is a resource type in Azure Compute Gallery that simplifies publishing, deployment, management, sharing, and global distribution of applications and scripts for your virtual machines. VM Applications support a wide range of scenarios, including high-scale deployments, low-latency requirements, failure resiliency, secure trusted rollouts, fleet-wide consistency, microservice architectures, and post-deployment management. Learn more about VM Applications.

To create and deploy applications on Azure VM, first package and upload your application to Azure Storage Account as a storage blob. Then create `Azure VM application` resource and `VM application version` resource referencing these storage blobs. Azure stores and replicates these blobs in regional managed storage accounts. Finally, deploy the application on any VM or Virtual Machine Scale Sets by passing application reference in `applicationProfile`.



Prerequisites

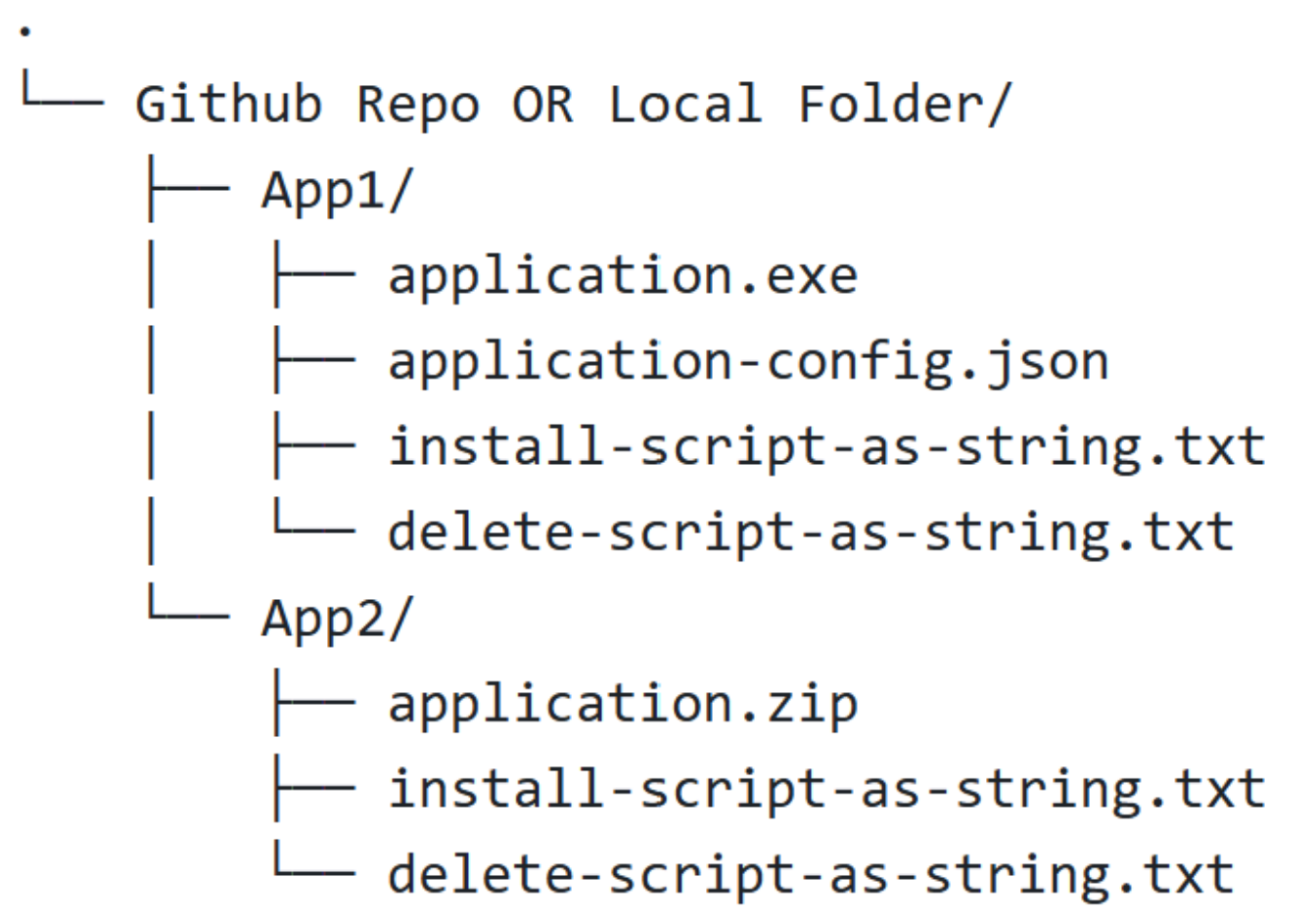
1. Create Azure storage account and storage container. This container is used to upload your application files.

2. Create Azure Compute Gallery for storing and sharing application resources.
3. Create user-assigned managed identity and assign to Azure Compute Gallery

> ⓘ Note
>
> Create your storage account with anonymous access disabled for improved security. Use a managed identity assigned to Azure Compute Gallery to publish your VM Applications securely without requiring SAS tokens.

Package the application

Before publishing the VM Application resource in Azure Compute Gallery, package your application files for upload to Azure Storage Account. The following diagram shows possible folder structures for organizing your application files:



1. Package the application files

VM Application requires a single file as the application package.

- If your application installation requires a single executable file (.exe, .msi, .sh, .ps, etc.) then you can use it as the application package.
- If your application requires multiple files (executables, dependencies, scripts, metadata, etc.), archive them into a single file (`.zip`, `.tar`, `.tar.gz`) and use this file as the application package.
- For microservice application, you can package and publish each microservice as a separate Azure VM Application. This modularization facilitates application reusability, cross-team development, and sequential installation of microservices using `order` property in the applicationProfile.

Learn more about creating application packages for Azure VM Applications.

2. (Optional) Package the application configuration file

- You can provide the configuration file separately using `defaultConfigurationLink` in the `publishingProfile` of the VM Application. This reduces the overhead of archiving and unarchiving application packages.
- Configuration files can also be passed during app deployment using `configurationReference` property in the `applicationProfile` of the VM enabling customized installation per VM.
- If both properties are set, `configurationReference` overrides the configuration passed using `defaultConfigurationLink`.

3. Create the install script

After the application and configuration blob is downloaded on the VM, Azure executes the provided install script to install the application. **The install script is provided as a string** and has a maximum character limit of 4,096 chars. The install commands should be written assuming the application package and the configuration file are in the current directory.

There may be few operations required to be performed in the install script

- **(Optional) Use the right command interpreter** The default command interpreter used by Azure are `/bin/bash` on Linux OS and `cmd.exe` on Windows OS. It's possible to use a different interpreter like Chocolatey or PowerShell, if its installed on the machine. Call the executable and pass the commands to it. For example, `powershell.exe -command '<powershell command>'`. If you're using PowerShell, you need to be using version 3.11.0 of the Az.Storage module.

- **(Optional) Rename application blob and configuration blob** Azure can't retain the original blob name and the file extensions. Azure downloads the application file with the

name of the VM application resource published on Azure Compute Gallery. So if name of VM App is "VMApp1", the file downloaded on VM has the name "VMApp1" without an extension. The configuration file has a default name as "VMApp1-config" without a file extension.

You can rename the file with the file extension using the install script or you can also pass the names in `packageFileName` and `configFileName` properties of the publishingProfile of VM Application version resource. Azure uses these names instead of default names while downloading the files.

- **Unarchive application blob** For archived application packages, it needs to be unarchived before installing the application. Use .zip or .tar since most OS has built-in support for unarchiving these formats. For other formats, make sure the Guest OS provides support.

- **Convert the script to string** The install script is passed as a string for the `install` property in the `publishingProfile` of Azure VM Application version resource.

- **Install quietly and synchronously** Install applications quietly or silently without any user interface prompts. Since the installation runs on a VM where customer input isn't available, you must use silent or unattended installation flags. Additionally, if the application installer spawns background or child processes or runs asynchronously, the parent script might return before the installation completes. To ensure the full installation finishes before Azure marks the operation as successful, use wait flags or blocking commands. This approach helps when installation involves multiple steps that run asynchronously or you need to verify installation completes before the subsequent application in the `order` sequence begins.

  Common examples include:
  - **Windows .msi**: `start /wait msiexec /i app.msi /quiet /qn`
  - **Windows .exe**: `start /wait app.exe /silent` or `start /wait app.exe /quiet` or `start /wait app.exe /S`
  - **PowerShell**: `Start-Process -FilePath '.\app.exe' -ArgumentList '/quiet' -Wait`
  - **Linux .sh**: `./install.sh --silent` (shell scripts run synchronously by default)
  - **Linux .deb**: `sudo DEBIAN_FRONTEND=noninteractive dpkg -i app.deb` (runs synchronously by default)
  - **Linux .rpm**: `sudo rpm -ivh app.rpm` (runs synchronously by default)

- **(Optional) Set right execution policy and permissions** After unarchiving, file permissions could be reset. It's a good practice to set the right permissions before executing the files.

- **(Optional) Set verbose execution and logging** VM Applications automatically captures any output from the install script and stores it in a stdout file on the VM. Use this

capability to log installation progress, debug failures, and verify successful deployments.
    - **Enable verbose mode** to capture application installer logs. Bash: `set -x`, PowerShell: `$VerbosePreference='Continue'`, Cmd: `@echo on`.
    - **Add custom messages** to track script progress. Bash: `echo "Install started"`, PowerShell: `Write-Output "Install started"`, Cmd: `echo Install started`

    You can retrieve these logs using Run Command or by connecting to the VM. The `stdout` and `stderr` files are located at the following path
    - Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>/`
    - Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.9\Downloads\<application name>\<application version>/`

- **(Optional) Move application and configuration blob to appropriate location** Azure downloads the application blob and configuration blob to following locations. The install script must move the files to appropriate locations when necessary.

    Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>`

    Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.16\Downloads\<application name>\<application version>`

Here are few sample install scripts based on the file extension of the application blob. If `packageFileName` & `configFileName` properties are provided in the `publishingProfile` with file name and extension, remove the code for renaming files in following scripts.

.TAR

Replace `VMApp1` with your VM App name in application package (VMApp1) and configuration file (VMApp1-config). Replace `app.tar` with desired app name. Replace `config.yaml` with your desired config name.

Install using bash without config file:

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && bash ./install.sh
```

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && sudo
DEBIAN_FRONTEND=noninteractive dpkg -i ./app.deb
```

Install using bash with config file:

Bash

```
mv VMApp1 app.tar && mv VMApp1-config config.yaml && tar -xf app.tar && chmod -
R +xr . && bash ./install.sh --config config.yaml
```

If config file is packaged within the tar file

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && sudo debconf-set-
selections < config.cfg && sudo DEBIAN_FRONTEND=noninteractive dpkg -i
./app.deb
```

4. Create the remove script

The `remove` script enables you to define the operations for removing the application. The `remove` script is provided as a string and has a maximum character limit of 4,096 characters. Write the commands assuming the application package and the configuration file are in the current directory. During uninstall operation, Azure runs the uninstall script and then deletes all files from the repository.

There are few operations that the `remove` script must perform.

- **Uninstall application:** Properly uninstall the application from the VM. For example:

CMD

CMD on Windows:

cli

```
start /wait uninstall.exe /quiet
```

cli

```
start /wait python.exe /uninstall /quiet
```

cli

```
start /wait msiexec /x app.msi /quiet /qn
```

- **Remove residual files:** If the application installation creates files in other parts of the file system, remove those files.

Upload the application files to Azure storage account

> ⓘ **Important**
>
> Before uploading files or generating SAS URLs, assign roles to the user, service principal, or managed identity that performs these operations on the target Storage account or container:
>
> - Storage Blob Data Contributor or Storage Blob Data Owner: required to upload, modify, or delete blobs.
> - For generating SAS with Azure AD (for example, using `--auth-mode` login in CLI/PowerShell), also assign Storage Blob Delegator at the storage account scope.

1. Upload your application and configuration files to a container in your Azure storage account.

Upload your packaged application and optional configuration file as blobs. Block blobs work for most scenarios.

Optionally, if you need to use page blobs, byte-align your files before uploading using one of the following scripts:

CLI

Azure CLI

```
inputFile="<the file you want to pad>"

# Get the file size
fileSize=$(stat -c %s "$inputFile")

# Calculate the remainder when divided by 512
```

```
remainder=$((fileSize % 512))

if [ "$remainder" -ne 0 ]; then
  # Calculate how many bytes to pad
  difference=$((512 - remainder))

  # Create padding (empty bytes)
  dd if=/dev/zero bs=1 count=$difference >> "$inputFile"
fi
```

2. Generate SAS or Blob URL for the application package and the configuration file

Once the application and configuration files are uploaded to the storage account, you need to generate a Blob URL or SAS URL with read privilege for these blobs. These URLs are then provided as reference while creating the VM Application version resource.

> ⓘ **Note**
>
> Use of blob URL with managed identity is strongly recommended for improved security and privacy.

Use blob URL:

- If managed identity is assigned to Azure Compute Gallery.
- If storage account is enabled for anonymous access. Use this method only for testing since its insecure.

Use SAS URL:

- If storage account is disabled for anonymous access and managed identity isn't assigned to Azure Compute Gallery.

Blob URL = https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${BLOB_NAME}
SAS URL = https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${BLOB_NAME}?${SAS_TOKEN}

CLI using Blob URL

Use the following script if managed identity is assigned to Azure Compute Gallery. This approach uses blob URLs without SAS tokens.

Azure CLI

```
#!/bin/bash
set -euo pipefail

# === CONFIGURATION ===
STORAGE_ACCOUNT="your-storage-account-name"
CONTAINER_NAME="your-container-name"
APP_FILE="./path/to/your-app-file"           # Path to your application payload
file
CONFIG_FILE="./path/to/your-config-file"     # Path to your configuration file
(optional)

# === LOGIN (if not already logged in) ===
az login --only-show-errors

# === CREATE CONTAINER IF NOT EXISTS ===
az storage container create \
  --name "$CONTAINER_NAME" \
  --account-name "$STORAGE_ACCOUNT" \
  --auth-mode login \
  --only-show-errors

# === UPLOAD APPLICATION FILE ===
APP_BLOB_NAME=$(basename "$APP_FILE")
az storage blob upload \
  --account-name "$STORAGE_ACCOUNT" \
  --container-name "$CONTAINER_NAME" \
  --name "$APP_BLOB_NAME" \
  --file "$APP_FILE" \
  --auth-mode login \
  --overwrite \
  --only-show-errors

# === UPLOAD CONFIG FILE (optional) ===
if [ -n "$CONFIG_FILE" ] && [ -f "$CONFIG_FILE" ]; then
  CONFIG_BLOB_NAME=$(basename "$CONFIG_FILE")
  az storage blob upload \
    --account-name "$STORAGE_ACCOUNT" \
    --container-name "$CONTAINER_NAME" \
    --name "$CONFIG_BLOB_NAME" \
    --file "$CONFIG_FILE" \
    --auth-mode login \
    --overwrite \
    --only-show-errors
fi

# === GENERATE BLOB URLs ===
echo "Generating Blob URLs..."

APP_BLOB_URL="https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME
}/${APP_BLOB_NAME}"
echo "Application file: $APP_BLOB_URL"
```

```
if [ -n "$CONFIG_FILE" ] && [ -f "$CONFIG_FILE" ]; then

CONFIG_BLOB_URL="https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${CONFIG_BLOB_NAME}"
  echo "Configuration file: $CONFIG_BLOB_URL"
fi
```

Create the VM Application

To create the VM Application, first create the **VM Application** resource, which describes the application. Then create **VM Application Version** resource within it, which contains the VM application payload and scripts to install, update, and delete the application. Payload is supplied using Blob URL or SAS URL to the blob container in Azure Storage Account.

Refer schema for VM Application and VM Application version resource to learn more about each property.

Template

Use the following ARM template to create an Azure Compute Gallery, VM Application, and VM Application version. This template demonstrates the key properties and configuration options for publishing your application package.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2020-06-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "galleryName": {
      "type": "string"
    },
    "applicationName": {
      "type": "string"
    },
    "versionName": {
      "type": "string",
      "metadata": {
        "description": "Must follow the format: major.minor.patch (Example: 1.0.0)"
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "West US"
    },
```

```
    "supportedOSType": {
      "type": "string",
      "allowedValues": ["Windows", "Linux"]
    },
    "endOfLifeDate": {
      "type": "string",
      "metadata": {
        "description": "Optional. This property is for information only and
doesn't block app deployment."
      }
    },
    "description": {
      "type": "string",
      "defaultValue": "Description of the application"
    },
    "eula": {
      "type": "string",
      "defaultValue": ""
    },
    "privacyStatementUri": {
      "type": "string",
      "defaultValue": ""
    },
    "releaseNoteUri": {
      "type": "string",
      "defaultValue": ""
    },
    "mediaLink": {
      "type": "string"
    },
    "configLink": {
      "type": "string"
    },
    "appConfigFileName": {
      "type": "string"
    },
    "appPackageFileName": {
      "type": "string"
    },
    "replicaRegion1": {
      "type": "string",
      "defaultValue": "East US"
    },
    "replicaRegion2": {
      "type": "string",
      "defaultValue": "South Central US"
    },
    "installScript": {
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to install the application.
Example: echo 'Installing application...'"
      }
    },
    "updateScript": {
```

```
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to update the application.
Example: echo 'Updating application...'"
      }
    },
    "removeScript": {
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to delete the application.
Example: echo 'Deleting application...'"
      }
    },
    "storageAccountType": {
      "type": "string",
      "allowedValues": ["PremiumV2_LRS", "Premium_LRS", "Standard_LRS",
"Standard_ZRS"],
      "defaultValue": "Standard_LRS"
    }
  },
  "resources": [
    {
      "type": "Microsoft.Compute/galleries",
      "apiVersion": "2024-03-03",
      "name": "[parameters('galleryName')]",
      "location": "[parameters('location')]",
      "properties": {
        "identifier": {}
      }
    },
    {
      "type": "Microsoft.Compute/galleries/applications",
      "apiVersion": "2024-03-03",
      "name": "[format('{0}/{1}', parameters('galleryName'),
parameters('applicationName'))]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Compute/galleries',
parameters('galleryName'))]"
      ],
      "properties": {
        "supportedOSType": "[parameters('supportedOSType')]",
        "endOfLifeDate": "[parameters('endOfLifeDate')]",
        "description": "[parameters('description')]",
        "eula": "[if(equals(parameters('eula'), ''), json('null'),
parameters('eula'))]",
        "privacyStatementUri": "[if(equals(parameters('privacyStatementUri'),
''), json('null'), parameters('privacyStatementUri'))]",
        "releaseNoteUri": "[if(equals(parameters('releaseNoteUri'), ''),
json('null'), parameters('releaseNoteUri'))]"
      }
    },
    {
      "type": "Microsoft.Compute/galleries/applications/versions",
      "apiVersion": "2024-03-03",
```

```
      "name": "[format('{0}/{1}/{2}', parameters('galleryName'),
parameters('applicationName'), parameters('versionName'))]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Compute/galleries/applications',
parameters('galleryName'), parameters('applicationName'))]"
      ],
      "properties": {
        "publishingProfile": {
          "source": {
            "mediaLink": "[parameters('mediaLink')]",
            "defaultConfigurationLink": "[parameters('configLink')]"
          },
          "manageActions": {
            "install": "[parameters('installScript')]",
            "remove": "[parameters('removeScript')]",
            "update": "[parameters('updateScript')]"
          },
          "settings": {
            "scriptBehaviorAfterReboot": "Rerun",
            "configFileName": "[parameters('appConfigFileName')]",
            "packageFileName": "[parameters('appPackageFileName')]"
          },
          "targetRegions": [
            {
              "name": "[parameters('location')]",
              "regionalReplicaCount": 3,
              "storageAccountType": "[parameters('storageAccountType')]"
            },
            {
              "name": "[parameters('replicaRegion1')]",
              "regionalReplicaCount": 1,
              "storageAccountType": "[parameters('storageAccountType')]"
            },
            {
              "name": "[parameters('replicaRegion2')]"
            },
          ],
          "excludeFromLatest": false,
          "replicaCount": 2,
          "storageAccountType": "[parameters('storageAccountType')]"
        },
        "safetyProfile": {
          "allowDeletionOfReplicatedLocations": true
        },
        "endOfLifeDate": "[parameters('endOfLifeDate')]"
      }
    }
  ]
}
```

Deploy the VM Apps

One or more VM Applications can now be referenced in the `applicationProfile` of Azure VM or Azure Virtual Machine Scale Sets. Azure then pulls the payload of the VM Application and installs it on each VM using the provided install script. The `order` property defines the sequential order in which the VM Applications are installed on the VM.

Refer schema of applicationProfile of the VM / Virtual Machine Scale Sets to learn more about each property.

Template

Use the following ARM template to deploy VM Application on Azure VM or Azure Virtual Machine Scale Sets. This template demonstrates the key properties and configuration options for deploying your VM application.

Deploy VM Application on Virtual Machine Scale Sets

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmssName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "subscriptionId": {
      "type": "string"
    },
    "resourceGroupName": {
      "type": "string"
    },
    "galleryName": {
      "type": "string"
    },
    "applicationName1": {
      "type": "string"
    },
    "applicationVersion1": {
      "type": "string",
      "defaultValue": "latest"
    },
    "configurationReference1": {
      "type": "string",
```

```
      "metadata": {
        "description": "Optional path to configuration file from Storage
Account. Overrides default configuration file."
      }
    },
    "applicationName2": {
      "type": "string"
    },
    "applicationVersion2": {
      "type": "string",
      "defaultValue": "latest"
    }
  },
  "variables": {
    "packageReferenceId1": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName1'), parameters('applicationVersion1'))]",
    "packageReferenceId2": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName2'), parameters('applicationVersion2'))]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachineScaleSets",
      "apiVersion": "2024-03-03",
      "name": "[parameters('vmssName')]",
      "location": "[parameters('location')]",
      "properties": {
        "virtualMachineProfile": {
          "applicationProfile": {
            "galleryApplications": [
              {
                "order": 1,
                "packageReferenceId": "[variables('packageReferenceId1')]",
                "configurationReference": "
[parameters('configurationReference1')]",
                "treatFailureAsDeploymentFailure": true
              },
              {
                "order": 2,
                "packageReferenceId": "[variables('packageReferenceId2')]",
                "treatFailureAsDeploymentFailure": false
              }
            ]
          }
        }
      }
    }
  ]
}
```

Deploy VM Application on Azure VM

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "subscriptionId": {
      "type": "string"
    },
    "resourceGroupName": {
      "type": "string"
    },
    "galleryName": {
      "type": "string"
    },
    "applicationName1": {
      "type": "string"
    },
    "applicationVersion1": {
      "type": "string",
      "defaultValue": "latest"
    },
    "configurationReference1": {
      "type": "string",
      "metadata": {
        "description": "Optional path to configuration blob for application 1"
      }
    },
    "applicationName2": {
      "type": "string"
    },
    "applicationVersion2": {
      "type": "string",
      "defaultValue": "latest"
    }
  },
  "variables": {
    "packageReferenceId1": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName1'), parameters('applicationVersion1'))]",
    "packageReferenceId2": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
```

```
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName2'), parameters('applicationVersion2'))]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2024-07-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "applicationProfile": {
          "galleryApplications": [
            {
              "order": 1,
              "packageReferenceId": "[variables('packageReferenceId1')]",
              "configurationReference": "
[parameters('configurationReference1')]",
              "treatFailureAsDeploymentFailure": true
            },
            {
              "order": 2,
              "packageReferenceId": "[variables('packageReferenceId2')]",
              "treatFailureAsDeploymentFailure": false
            }
          ]
        }
      }
    }
  ]
}
```

Next steps

- Learn to manage, update, or delete Azure VM Applications.
- Learn more about Azure VM Applications.

Last updated on 03/06/2026

Tutorial - Deploy applications to a Windows virtual machine in Azure with the Custom Script Extension

Article • 08/22/2024

Applies to: ✔ Windows ✔ Flexible scale sets ✔ Uniform scale sets

To configure virtual machines (VMs) in a quick and consistent manner, you can use the Custom Script Extension for Windows. In this tutorial you learn how to:

- ✔ Use the Custom Script Extension to install IIS.
- ✔ Create a VM that uses the Custom Script Extension.
- ✔ View a running IIS site after the extension is applied.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, select **Open Cloudshell** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Custom script extension overview

The Custom Script Extension downloads and executes scripts on Azure VMs. This extension is useful for post-deployment configuration, software installation, or any other configuration or management task. You can download scripts from Azure storage or GitHub, or you can provide scripts to the Azure portal at extension run time.

The Custom Script extension integrates with Azure Resource Manager templates and can be run by using the Azure CLI, PowerShell, Azure portal, or the Azure Virtual Machine REST API.

You can use the Custom Script Extension with both Linux and Windows VMs.

Create virtual machine

Set the administrator username and password for the VM with Get-Credential:

Azure PowerShell

```
$cred = Get-Credential
```

Now you can create the VM with New-AzVM. The following example creates a VM named *myVM* in the *EastUS* location. If they don't already exist, the resource group *myResourceGroupAutomate* and supporting network resources are created. To allow web traffic, the cmdlet also opens port *80*.

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName "myResourceGroupAutomate" `
    -Name "myVM" `
    -Location "East US" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress" `
    -OpenPorts 80 `
    -Credential $cred
```

The resources and VM take a few minutes to be created.

Automate IIS install

Use Set-AzVMExtension to install the Custom Script Extension. The extension runs `powershell Add-WindowsFeature Web-Server` to install the IIS webserver and then updates the *Default.htm* page to show the hostname of the VM:

Azure PowerShell

```
Set-AzVMExtension -ResourceGroupName "myResourceGroupAutomate" `
    -ExtensionName "IIS" `
    -VMName "myVM" `
    -Location "EastUS" `
    -Publisher Microsoft.Compute `
    -ExtensionType CustomScriptExtension `
    -TypeHandlerVersion 1.8 `
    -SettingString '{"commandToExecute":"powershell Add-WindowsFeature Web-
Server; powershell Add-Content -Path \"C:\\inetpub\\wwwroot\\Default.htm\" -
Value $($env:computername)"}'
```

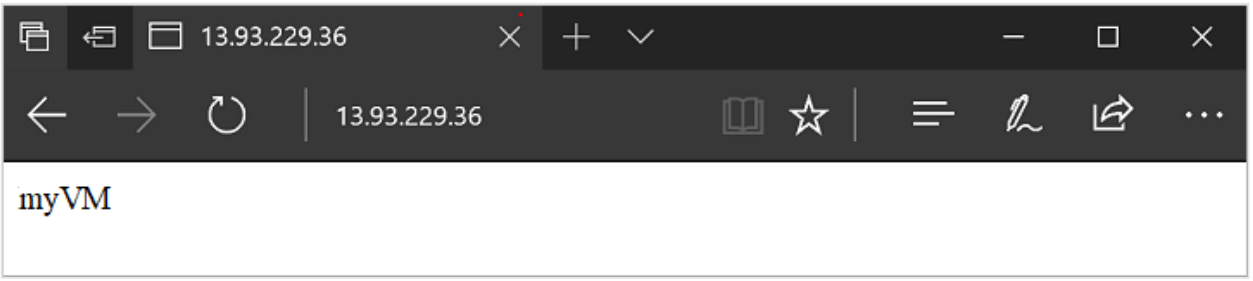
Test web site

Obtain the public IP address of your load balancer with Get-AzPublicIPAddress. The following example obtains the IP address for `myPublicIPAddress` created earlier:

Azure PowerShell

```
Get-AzPublicIPAddress `
    -ResourceGroupName "myResourceGroupAutomate" `
    -Name "myPublicIPAddress" | select IpAddress
```

You can then enter the public IP address in to a web browser. The website is displayed, including the hostname of the VM that the load balancer distributed traffic to as in the following example:



Next steps

In this tutorial, you automated the IIS install on a VM. You learned how to:

- ✓ Use the Custom Script Extension to install IIS.
- ✓ Create a VM that uses the Custom Script Extension.
- ✓ View a running IIS site after the extension is applied.

Advance to the next tutorial to learn how to create custom VM images.

Create custom VM images

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Create and manage Azure virtual networks for Windows virtual machines with Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Windows VMs

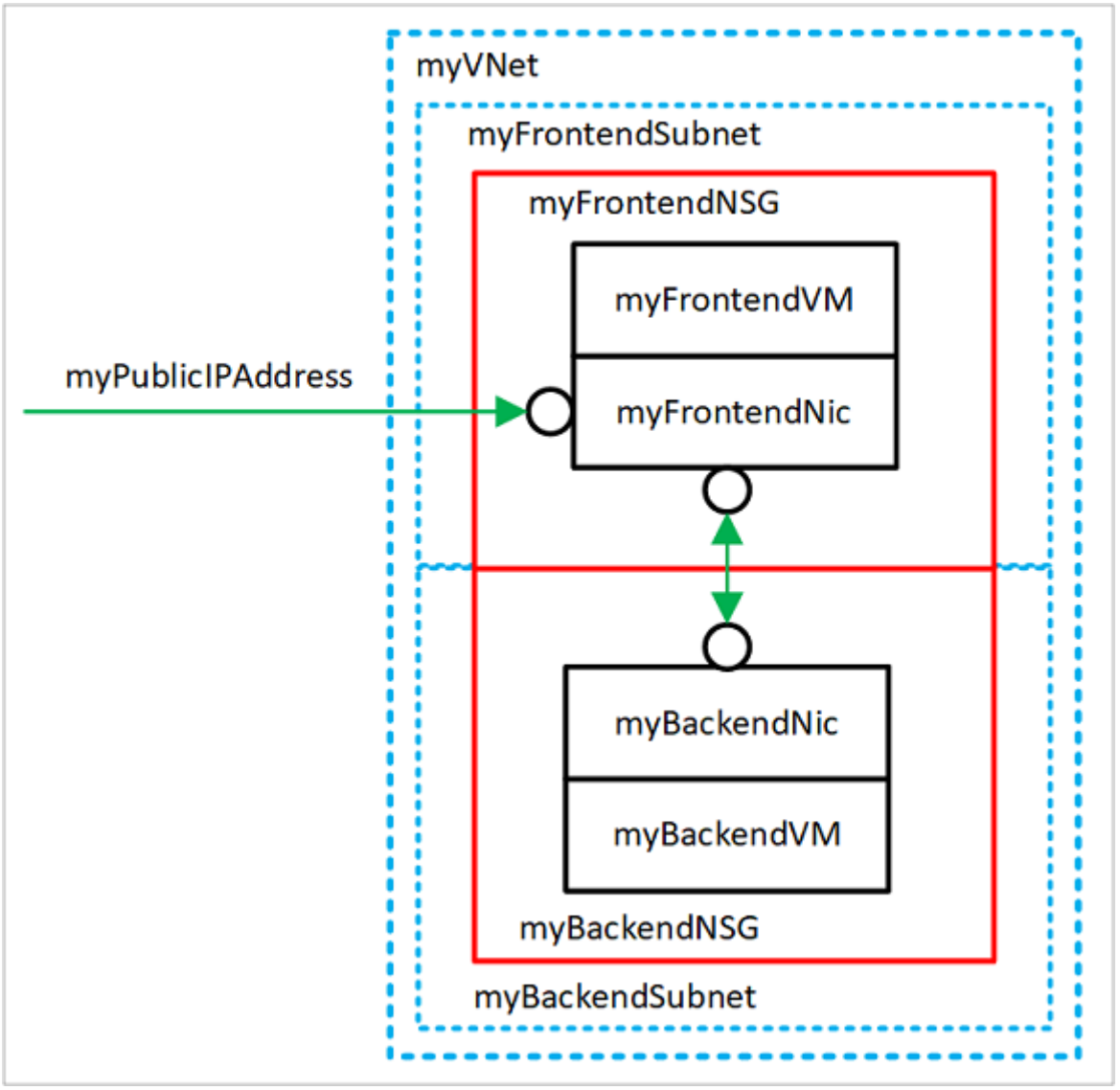
Azure virtual machines use Azure networking for internal and external network communication. This tutorial walks through deploying two virtual machines and configuring Azure networking for these VMs. The examples in this tutorial assume that the VMs are hosting a web application with a database back-end, however an application isn't deployed in the tutorial. In this tutorial, you learn how to:

- ✓ Create a virtual network and subnet
- ✓ Create a public IP address
- ✓ Create a front-end VM
- ✓ Secure network traffic
- ✓ Create back-end VM

VM networking overview

Azure virtual networks enable secure network connections between virtual machines, the internet, and other Azure services such as Azure SQL Database. Virtual networks are broken down into logical segments called subnets. Subnets are used to control network flow, and as a security boundary. When deploying a VM, it generally includes a virtual network interface, which is attached to a subnet.

While completing this tutorial, you can see these resources created:



- *myVNet* - The virtual network that the VMs use to communicate with each other and the internet.
- *myFrontendSubnet* - The subnet in *myVNet* used by the front-end resources.
- *myPublicIPAddress* - The public IP address used to access *myFrontendVM* from the internet.
- *myFrontendNic* - The network interface used by *myFrontendVM* to communicate with *myBackendVM*.
- *myFrontendVM* - The VM used to communicate between the internet and *myBackendVM*.
- *myBackendNSG* - The network security group that controls communication between the *myFrontendVM* and *myBackendVM*.
- *myBackendSubnet* - The subnet associated with *myBackendNSG* and used by the back-end resources.
- *myBackendNic* - The network interface used by *myBackendVM* to communicate with *myFrontendVM*.
- *myBackendVM* - The VM that uses port 1433 to communicate with *myFrontendVM*.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Create subnet

For this tutorial, a single virtual network is created with two subnets. A front-end subnet for hosting a web application, and a back-end subnet for hosting a database server.

Before you can create a virtual network, create a resource group using New-AzResourceGroup. The following example creates a resource group named *myRGNetwork* in the *EastUS* location:

Azure PowerShell

```
New-AzResourceGroup -ResourceGroupName myRGNetwork -Location EastUS
```

Create a subnet configuration named *myFrontendSubnet* using New-AzVirtualNetworkSubnetConfig:

Azure PowerShell

```
$frontendSubnet = New-AzVirtualNetworkSubnetConfig `
  -Name myFrontendSubnet `
  -AddressPrefix 10.0.0.0/24
```

And, create a subnet configuration named *myBackendSubnet*:

Azure PowerShell

```
$backendSubnet = New-AzVirtualNetworkSubnetConfig `
  -Name myBackendSubnet `
  -AddressPrefix 10.0.1.0/24
```

Create virtual network

Create a VNET named *myVNet* using *myFrontendSubnet* and *myBackendSubnet* using New-AzVirtualNetwork:

Azure PowerShell

```
$vnet = New-AzVirtualNetwork `
  -ResourceGroupName myRGNetwork `
  -Location EastUS `
  -Name myVNet `
  -AddressPrefix 10.0.0.0/16 `
  -Subnet $frontendSubnet, $backendSubnet
```

At this point, a network has been created and segmented into two subnets, one for front-end services, and another for back-end services. In the next section, virtual machines are created and connected to these subnets.

Create a public IP address

A public IP address allows Azure resources to be accessible on the internet. The allocation method of the public IP address can be configured as dynamic or static. By default, a public IP address is dynamically allocated. Dynamic IP addresses are released when a VM is deallocated. This behavior causes the IP address to change during any operation that includes a VM deallocation.

The allocation method can be set to static, which makes sure that the IP address stays assigned to a VM, even during a deallocated state. If you are using a static IP address, the IP address itself can't be specified. Instead, it's allocated from a pool of available addresses.

Create a public IP address named *myPublicIPAddress* using New-AzPublicIpAddress:

Azure PowerShell

```
$pip = New-AzPublicIpAddress `
  -ResourceGroupName myRGNetwork `
  -Location EastUS `
  -AllocationMethod Dynamic `
  -Name myPublicIPAddress
```

You could change the -AllocationMethod parameter to `Static` to assign a static public IP address.

Create a front-end VM

For a VM to communicate in a virtual network, it needs a virtual network interface (NIC). Create a NIC using New-AzNetworkInterface:

Azure PowerShell

```
$frontendNic = New-AzNetworkInterface `
  -ResourceGroupName myRGNetwork `
  -Location EastUS `
  -Name myFrontend `
  -SubnetId $vnet.Subnets[0].Id `
  -PublicIpAddressId $pip.Id
```

Set the username and password needed for the administrator account on the VM using Get-Credential. You use these credentials to connect to the VM in additional steps:

Azure PowerShell

```
$cred = Get-Credential
```

Create the VMs using New-AzVM.

Azure PowerShell

```
New-AzVM `
   -Credential $cred `
   -Name myFrontend `
   -PublicIpAddressName myPublicIPAddress `
   -ResourceGroupName myRGNetwork `
   -Location "EastUS" `
   -Size Standard_D1 `
   -SubnetName myFrontendSubnet `
   -VirtualNetworkName myVNet
```

Secure network traffic

A network security group (NSG) contains a list of security rules that allow or deny network traffic to resources connected to Azure Virtual Networks (VNet). NSGs can be associated to subnets or individual network interfaces. An NSG that is associated with a network interface only applies to the associated VM. When an NSG is associated to a subnet, the rules apply to all resources connected to the subnet.

Network security group rules

NSG rules define networking ports over which traffic is allowed or denied. The rules can include source and destination IP address ranges so that traffic is controlled between specific systems or subnets. NSG rules also include a priority (between 1—and 4096). Rules are evaluated in the order of priority. A rule with a priority of 100 is evaluated before a rule with priority 200.

All NSGs contain a set of default rules. The default rules can't be deleted, but because they are assigned the lowest priority, they can be overridden by the rules that you create.

- **Virtual network** - Traffic originating and ending in a virtual network is allowed both in inbound and outbound directions.
- **Internet** - Outbound traffic is allowed, but inbound traffic is blocked.
- **Load balancer** - Allow Azure's load balancer to probe the health of your VMs and role instances. If you are not using a load balanced set, you can override this rule.

Create network security groups

Create an inbound rule named *myFrontendNSGRule* to allow incoming web traffic on *myFrontendVM* using New-AzNetworkSecurityRuleConfig:

Azure PowerShell

```
$nsgFrontendRule = New-AzNetworkSecurityRuleConfig `
  -Name myFrontendNSGRule `
  -Protocol Tcp `
  -Direction Inbound `
  -Priority 200 `
  -SourceAddressPrefix * `
  -SourcePortRange * `
  -DestinationAddressPrefix * `
  -DestinationPortRange 80 `
  -Access Allow
```

You can limit internal traffic to *myBackendVM* from only *myFrontendVM* by creating an NSG for the back-end subnet. The following example creates an NSG rule named *myBackendNSGRule*:

Azure PowerShell

```
$nsgBackendRule = New-AzNetworkSecurityRuleConfig `
  -Name myBackendNSGRule `
  -Protocol Tcp `
  -Direction Inbound `
  -Priority 100 `
  -SourceAddressPrefix 10.0.0.0/24 `
```

```
  -SourcePortRange * `
  -DestinationAddressPrefix * `
  -DestinationPortRange 1433 `
  -Access Allow
```

Add a network security group named *myFrontendNSG* using New-AzNetworkSecurityGroup:

Azure PowerShell

```
$nsgFrontend = New-AzNetworkSecurityGroup `
  -ResourceGroupName myRGNetwork `
  -Location EastUS `
  -Name myFrontendNSG `
  -SecurityRules $nsgFrontendRule
```

Now, add a network security group named *myBackendNSG* using New-AzNetworkSecurityGroup:

Azure PowerShell

```
$nsgBackend = New-AzNetworkSecurityGroup `
  -ResourceGroupName myRGNetwork `
  -Location EastUS `
  -Name myBackendNSG `
  -SecurityRules $nsgBackendRule
```

Add the network security groups to the subnets:

Azure PowerShell

```
$vnet = Get-AzVirtualNetwork `
  -ResourceGroupName myRGNetwork `
  -Name myVNet
$frontendSubnet = $vnet.Subnets[0]
$backendSubnet = $vnet.Subnets[1]
$frontendSubnetConfig = Set-AzVirtualNetworkSubnetConfig `
  -VirtualNetwork $vnet `
  -Name myFrontendSubnet `
  -AddressPrefix $frontendSubnet.AddressPrefix `
  -NetworkSecurityGroup $nsgFrontend
$backendSubnetConfig = Set-AzVirtualNetworkSubnetConfig `
  -VirtualNetwork $vnet `
  -Name myBackendSubnet `
  -AddressPrefix $backendSubnet.AddressPrefix `
  -NetworkSecurityGroup $nsgBackend
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

Create a back-end VM

The easiest way to create the back-end VM for this tutorial is by using a SQL Server image. This tutorial only creates the VM with the database server, but doesn't provide information about accessing the database.

Create *myBackendNic*:

Azure PowerShell

```
$backendNic = New-AzNetworkInterface `
  -ResourceGroupName myRGNetwork `
  -Location EastUS `
  -Name myBackend `
  -SubnetId $vnet.Subnets[1].Id
```

Set the username and password needed for the administrator account on the VM with Get-Credential:

Azure PowerShell

```
$cred = Get-Credential
```

Create *myBackendVM*.

Azure PowerShell

```
New-AzVM `
   -Credential $cred `
   -Name myBackend `
   -ImageName "MicrosoftSQLServer:SQL2016SP1-WS2016:Enterprise:latest" `
   -ResourceGroupName myRGNetwork `
   -Location "EastUS" `
   -SubnetName MyBackendSubnet `
   -VirtualNetworkName myVNet
```

The image in this example has SQL Server installed, but it isn't used in this tutorial. It's included to show you how you can configure a VM to handle web traffic and a VM to handle database management.

Next steps

In this tutorial, you created and secured Azure networks as related to virtual machines.

- ✓ Create a virtual network and subnet

- ✓ Create a public IP address
- ✓ Create a front-end VM
- ✓ Secure network traffic
- ✓ Create a back-end VM

To learn about protecting your VM disks, see Backup and disaster recovery for disks.

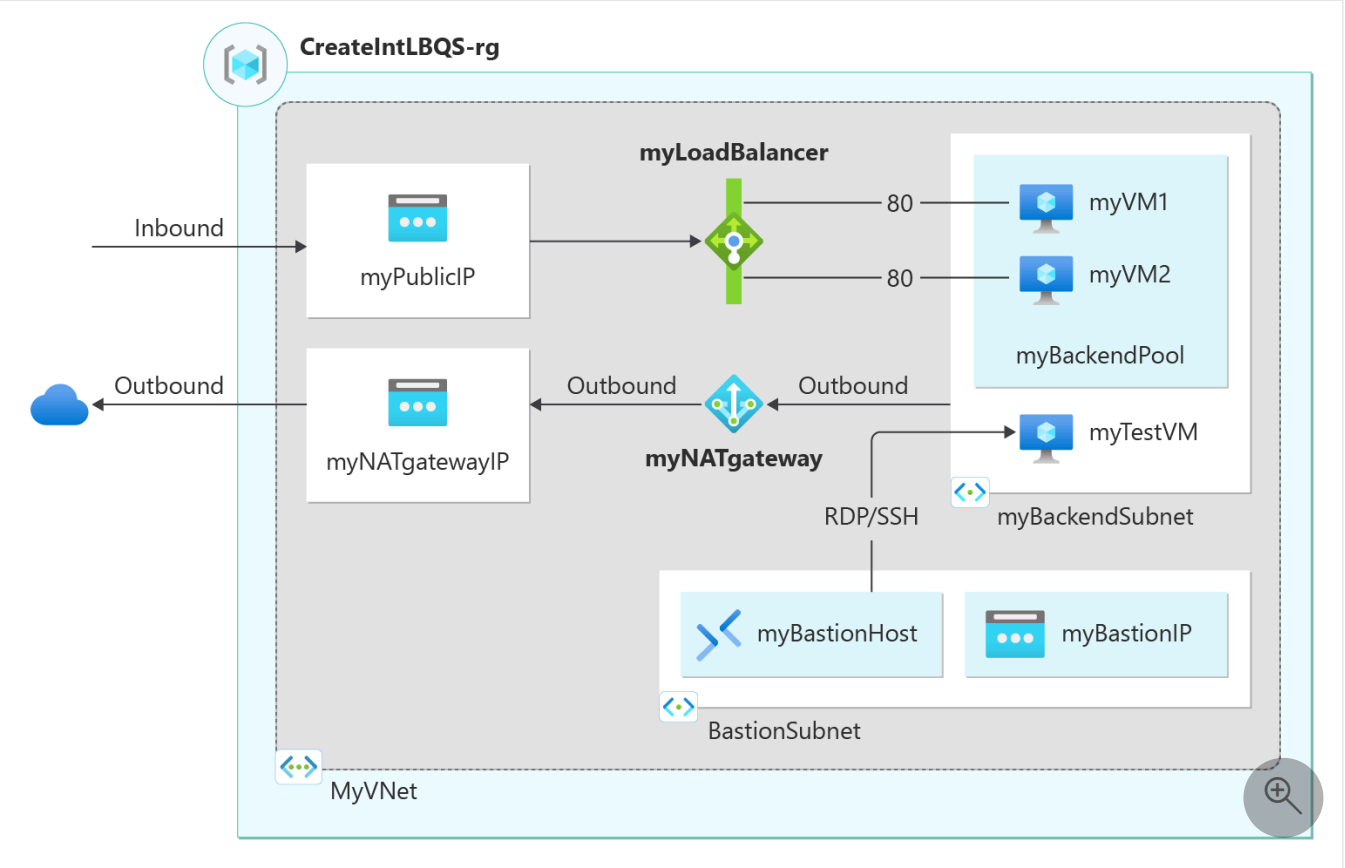
Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a public load balancer to load balance VMs using Azure PowerShell

Get started with Azure Load Balancer by using Azure PowerShell to create a public load balancer and two virtual machines. Also, you deploy other resources including Azure Bastion, NAT Gateway, a virtual network, and the required subnets.



Prerequisites

- An Azure account with an active subscription. Create an account for free
- Azure PowerShell installed locally or Azure Cloud Shell

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 5.4.1 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see Install Azure PowerShell module. If you're running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with New-AzResourceGroup:

Azure PowerShell

```
$rg = @{
    Name = 'CreatePubLBQS-rg'
    Location = 'westus2'
}
New-AzResourceGroup @rg
```

Create a public IP address

Use New-AzPublicIpAddress to create a public IP address.

Azure PowerShell

```
$publicip = @{
    Name = 'myPublicIP'
    ResourceGroupName = $rg.name
    Location = 'westus2'
    Sku = 'Standard'
    AllocationMethod = 'static'
    Zone = 1,2,3
}
New-AzPublicIpAddress @publicip
```

To create a zonal public IP address in zone 1, use the following command:

Azure PowerShell

```
$publicip = @{
    Name = 'myPublicIP'
    ResourceGroupName = $rg.name
    Location = 'westus2'
    Sku = 'Standard'
    AllocationMethod = 'static'
    Zone = 1
}
New-AzPublicIpAddress @publicip
```

Create a load balancer

This section details how you can create and configure the following components of the load balancer:

- Create a frontend IP with New-AzLoadBalancerFrontendIpConfig for the frontend IP pool. This IP receives the incoming traffic on the load balancer

- Create a backend address pool with New-AzLoadBalancerBackendAddressPoolConfig for traffic sent from the frontend of the load balancer. This pool is where your backend virtual machines are deployed
- Create a health probe with Add-AzLoadBalancerProbeConfig that determines the health of the backend VM instances
- Create a load balancer rule with Add-AzLoadBalancerRuleConfig that defines how traffic is distributed to the VMs
- Create a public load balancer with New-AzLoadBalancer

Azure PowerShell

```
## Place public IP created in previous steps into variable. ##
$pip = @{
    Name = 'myPublicIP'
    ResourceGroupName = $rg.name
}
$publicIp = Get-AzPublicIpAddress @pip

## Create load balancer frontend configuration and place in variable. ##
$fip = @{
    Name = 'myFrontEnd'
    PublicIpAddress = $publicIp
}
$feip = New-AzLoadBalancerFrontendIpConfig @fip

## Create backend address pool configuration and place in variable. ##
$bepool = New-AzLoadBalancerBackendAddressPoolConfig -Name 'myBackEndPool'

## Create the health probe and place in variable. ##
$probe = @{
    Name = 'myHealthProbe'
    Protocol = 'tcp'
    Port = '80'
    IntervalInSeconds = '360'
    ProbeCount = '5'
}
$healthprobe = New-AzLoadBalancerProbeConfig @probe

## Create the load balancer rule and place in variable. ##
$lbrule = @{
    Name = 'myHTTPRule'
    Protocol = 'tcp'
    FrontendPort = '80'
    BackendPort = '80'
    IdleTimeoutInMinutes = '15'
    FrontendIpConfiguration = $feip
    BackendAddressPool = $bePool
}
$rule = New-AzLoadBalancerRuleConfig @lbrule -EnableTcpReset -DisableOutboundSNAT
```

```
## Create the load balancer resource. ##
$loadbalancer = @{
    ResourceGroupName = $rg.name
    Name = 'myLoadBalancer'
    Location = 'westus2'
    Sku = 'Standard'
    FrontendIpConfiguration = $feip
    BackendAddressPool = $bePool
    LoadBalancingRule = $rule
    Probe = $healthprobe
}
New-AzLoadBalancer @loadbalancer
```

Configure virtual network

Before you deploy VMs and test your load balancer, create the supporting virtual network resources.

Create a virtual network for the backend virtual machines.

Create a network security group to define inbound connections to your virtual network.

Create an Azure Bastion host to securely manage the virtual machines in the backend pool.

Use a NAT gateway to provide outbound internet access to resources in the backend pool of your load balancer.

Create virtual network, network security group, bastion host, and NAT gateway

- Create a virtual network with New-AzVirtualNetwork
- Create a network security group rule with New-AzNetworkSecurityRuleConfig
- Create an Azure Bastion host with New-AzBastion
- Create a network security group with New-AzNetworkSecurityGroup
- Create the NAT gateway resource with New-AzNatGateway
- Use New-AzVirtualNetworkSubnetConfig to associate the NAT gateway to the subnet of the virtual network

> **Important**

Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

Azure PowerShell

```
## Create public IP address for NAT gateway ##
$ip = @{
    Name = 'myNATgatewayIP'
    ResourceGroupName = $rg.name
    Location = 'westus2'
    Sku = 'Standard'
    AllocationMethod = 'Static'
}
$publicIP = New-AzPublicIpAddress @ip

## Create NAT gateway resource ##
$nat = @{
    ResourceGroupName = $rg.name
    Name = 'myNATgateway'
    IdleTimeoutInMinutes = '10'
    Sku = 'Standard'
    Location = 'westus2'
    PublicIpAddress = $publicIP
}
$natGateway = New-AzNatGateway @nat

## Create backend subnet config ##
$subnet = @{
    Name = 'myBackendSubnet'
    AddressPrefix = '10.1.0.0/24'
    NatGateway = $natGateway
}
$subnetConfig = New-AzVirtualNetworkSubnetConfig @subnet

## Create Azure Bastion subnet. ##
$bastsubnet = @{
    Name = 'AzureBastionSubnet'
    AddressPrefix = '10.1.1.0/24'
}
$bastsubnetConfig = New-AzVirtualNetworkSubnetConfig @bastsubnet

## Create the virtual network ##
$net = @{
    Name = 'myVNet'
    ResourceGroupName = $rg.name
    Location = 'westus2'
    AddressPrefix = '10.1.0.0/16'
    Subnet = $subnetConfig,$bastsubnetConfig
}
$vnet = New-AzVirtualNetwork @net
```

```
## Create public IP address for bastion host. ##
$ip = @{
    Name = 'myBastionIP'
    ResourceGroupName = $rg.name
    Location = 'westus2'
    Sku = 'Standard'
    AllocationMethod = 'Static'
}
$publicip = New-AzPublicIpAddress @ip

## Create bastion host ##
$bastion = @{
    ResourceGroupName = $rg.name
    Name = 'myBastion'
    PublicIpAddress = $publicip
    VirtualNetwork = $vnet
}
New-AzBastion @bastion -AsJob

## Create rule for network security group and place in variable. ##
$nsgrule = @{
    Name = 'myNSGRuleHTTP'
    Description = 'Allow HTTP'
    Protocol = '*'
    SourcePortRange = '*'
    DestinationPortRange = '80'
    SourceAddressPrefix = 'Internet'
    DestinationAddressPrefix = '*'
    Access = 'Allow'
    Priority = '2000'
    Direction = 'Inbound'
}
$rule1 = New-AzNetworkSecurityRuleConfig @nsgrule

## Create network security group ##
$nsg = @{
    Name = 'myNSG'
    ResourceGroupName = $rg.name
    Location = 'westus2'
    SecurityRules = $rule1
}
New-AzNetworkSecurityGroup @nsg
```

Create virtual machines

In this section, you create the two virtual machines for the backend pool of the load balancer.

- Create two network interfaces with New-AzNetworkInterface
- Set an administrator username and password for the VMs with Get-Credential

- Create the virtual machines with:
  - New-AzVM
  - New-AzVMConfig
  - Set-AzVMOperatingSystem
  - Set-AzVMSourceImage
  - Add-AzVMNetworkInterface

Azure PowerShell

```
# Set the administrator and password for the VMs. ##
$cred = Get-Credential

## Place the virtual network into a variable. ##
$net = @{
    Name = 'myVNet'
    ResourceGroupName = $rg.name
}
$vnet = Get-AzVirtualNetwork @net

## Place the load balancer into a variable. ##
$lb = @{
    Name = 'myLoadBalancer'
    ResourceGroupName = $rg.name
}
$bepool = Get-AzLoadBalancer @lb  | Get-AzLoadBalancerBackendAddressPoolConfig

## Place the network security group into a variable. ##
$ns = @{
    Name = 'myNSG'
    ResourceGroupName = $rg.name
}
$nsg = Get-AzNetworkSecurityGroup @ns

## For loop with variable to create virtual machines for load balancer backend pool.
##
for ($i=1; $i -le 2; $i++){

    ## Command to create network interface for VMs ##
    $nic = @{
        Name = "myNicVM$i"
        ResourceGroupName = $rg.name
        Location = 'westus2'
        Subnet = $vnet.Subnets[0]
        NetworkSecurityGroup = $nsg
        LoadBalancerBackendAddressPool = $bepool
    }
    $nicVM = New-AzNetworkInterface @nic
```

```
    ## Create a virtual machine configuration for VMs ##
    $vmsz = @{
        VMName = "myVM$i"
        VMSize = 'Standard_DS1_v2'
    }
    $vmos = @{
        ComputerName = "myVM$i"
        Credential = $cred
    }
    $vmimage = @{
        PublisherName = 'MicrosoftWindowsServer'
        Offer = 'WindowsServer'
        Skus = '2019-Datacenter'
        Version = 'latest'
    }
    $vmConfig = New-AzVMConfig @vmsz `
        | Set-AzVMOperatingSystem @vmos -Windows `
        | Set-AzVMSourceImage @vmimage `
        | Add-AzVMNetworkInterface -Id $nicVM.Id

    ## Create the virtual machine for VMs ##
    $vm = @{
        ResourceGroupName = $rg.name
        Location = 'westus2'
        VM = $vmConfig
        Zone = "$i"
    }
    New-AzVM @vm -AsJob
}
```

The deployments of the virtual machines and bastion host are submitted as PowerShell jobs. To view the status of the jobs, use Get-Job:

Azure PowerShell

```
Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location
Command
--     ----            -------------   -----         -----------     --------
-------
1      Long Running O… AzureLongRunni… Completed     True            localhost
New-AzBastion
2      Long Running O… AzureLongRunni… Completed     True            localhost
New-AzVM
3      Long Running O… AzureLongRunni… Completed     True            localhost
New-AzVM
```

Ensure the **State** of the VM creation is **Completed** before moving on to the next steps.

> ⓘ Note

Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.

The default outbound access IP is disabled when one of the following events happens:

- A public IP address is assigned to the VM.
- The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
- An Azure NAT Gateway resource is assigned to the subnet of the VM.

VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.

For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Install IIS

Use Set-AzVMExtension to install the Custom Script Extension.

The extension runs `PowerShell Add-WindowsFeature Web-Server` to install the IIS webserver and then updates the Default.htm page to show the hostname of the VM:

ⓘ Important

Ensure the virtual machine deployments have completed from the previous steps before proceeding. Use `Get-Job` to check the status of the virtual machine deployment jobs.

Azure PowerShell

```
## For loop with variable to install custom script extension on virtual machines. ##
for ($i=1; $i -le 2; $i++)
{
$ext = @{
    Publisher = 'Microsoft.Compute'
    ExtensionType = 'CustomScriptExtension'
    ExtensionName = 'IIS'
    ResourceGroupName = $rg.name
    VMName = "myVM$i"
    Location = 'westus2'
    TypeHandlerVersion = '1.8'
    SettingString = '{"commandToExecute":"powershell Add-WindowsFeature Web-Server;
```

```
powershell Add-Content -Path \"C:\\inetpub\\wwwroot\\Default.htm\" -Value
$($env:computername)"}'
}
Set-AzVMExtension @ext -AsJob
}
```

The extensions are deployed as PowerShell jobs. To view the status of the installation jobs, use Get-Job:

Azure PowerShell

```
Get-Job

Id     Name            PSJobTypeName   State         HasMoreData     Location
Command
--     ----            -------------   -----         -----------     --------
-------
8      Long Running O… AzureLongRunni… Running       True            localhost
Set-AzVMExtension
9      Long Running O… AzureLongRunni… Running       True            localhost
Set-AzVMExtension
```

Ensure the **State** of the jobs is **Completed** before moving on to the next steps.

Test the load balancer

Use Get-AzPublicIpAddress to get the public IP address of the load balancer:

Azure PowerShell

```
$ip = @{
    ResourceGroupName = $rg.name
    Name = 'myPublicIP'
}
Get-AzPublicIPAddress @ip | select IpAddress
```

Copy the public IP address, and then paste it into the address bar of your browser. The default page of IIS Web server is displayed on the browser.



Clean up resources

When no longer needed, you can use the Remove-AzResourceGroup command to remove the resource group, load balancer, and the remaining resources.

Azure PowerShell

```
Remove-AzResourceGroup -Name $rg.name
```

Next steps

In this quickstart, you:

- Created an Azure Load Balancer
- Attached two VMs to the load balancer
- Tested the load balancer

To learn more about Azure Load Balancer, continue to:

What is Azure Load Balancer?

Last updated on 01/28/2026

Tutorial: Create and manage a Virtual Machine Scale Set with Azure PowerShell

Article • 08/22/2024

A Virtual Machine Scale Set allows you to deploy and manage a set of virtual machines. Throughout the lifecycle of a Virtual Machine Scale Set, you may need to run one or more management tasks. In this tutorial you learn how to:

- ✓ Create a resource group
- ✓ Create a Virtual Machine Scale Set
- ✓ Scale out and in
- ✓ Stop, Start and restart VM instances

If you don't have an Azure subscription, create a free account before you begin.

Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article, without having to install anything on your local environment.

To start Azure Cloud Shell:

Expand table

Option	Example/Link
Select **Try It** in the upper-right corner of a code or command block. Selecting **Try It** doesn't automatically copy the code or command to Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To use Azure Cloud Shell:

1. Start Cloud Shell.

2. Select the **Copy** button on a code block (or command block) to copy the code or command.
3. Paste the code or command into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux, or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code or command.

Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a Virtual Machine Scale Set. Create a resource group with the New-AzResourceGroup command. In this example, a resource group named *myResourceGroup* is created in the *EastUS* region.

Azure PowerShell

```
New-AzResourceGroup -ResourceGroupName "myResourceGroup" -Location "EastUS"
```

The resource group name is specified when you create or modify a scale set throughout this tutorial.

Create a Virtual Machine Scale Set

First, set an administrator username and password for the VM instances with Get-Credential:

Azure PowerShell

```
$cred = Get-Credential
```

Now create a Virtual Machine Scale Set with New-AzVmss. To distribute traffic to the individual VM instances, a load balancer is also created. The load balancer includes rules to distribute traffic on TCP port 80, and allow remote desktop traffic on TCP port 3389 and PowerShell remoting on TCP port 5985:

> **ⓘ Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to

[Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub](#)

Azure PowerShell

```
New-AzVmss `
  -ResourceGroupName "myResourceGroup" `
  -VMScaleSetName "myScaleSet" `
  -OrchestrationMode "Flexible" `
  -Location "EastUS" `
  -Credential $cred
```

It takes a few minutes to create and configure all the scale set resources and VM instances. To distribute traffic to the individual VM instances, a load balancer is also created.

View the VM instances in a scale set

To view a list of VM instances in a scale set, use Get-AzVM as follows:

Azure PowerShell

```
Get-AzVM -ResourceGroupName "myResourceGroup"
```

The following example output shows two VM instances in the scale set:

Output

```
ResourceGroupName                Name Location          VmSize  OsType
ProvisioningState
-----------------                ---- --------          ------  ------
-----------------
myResourceGroup   myScaleSet_instance1   eastus Standard_DS1_v2 Windows
Succeeded
myResourceGroup   myScaleSet_instance2   eastus Standard_DS1_v2 Windows
Succeeded
```

To view additional information about a specific VM instance, use Get-AzVM and specify the VM name.

Azure PowerShell

```
Get-AzVM -ResourceGroupName "myResourceGroup" -name "myScaleSet_instance1"
```

Output

```
ResourceGroupName      : myresourcegroup
Id                     :
/subscriptions/resourceGroups/myresourcegroup/providers/Microsoft.Compute/vi
rtualMachines/myScaleSet_instance1
VmId                   : d27b5fde-d469-4087-b08f-87d0bd8df786
Name                   : myScaleSet_instance1
Type                   : Microsoft.Compute/virtualMachines
Location               : eastus
Tags                   : {}
HardwareProfile        : {VmSize}
NetworkProfile         : {NetworkInterfaces}
OSProfile              : {ComputerName, AdminUsername, WindowsConfiguration,
Secrets, AllowExtensionOperations, RequireGuestProvisionSignal}
ProvisioningState      : Succeeded
StorageProfile         : {ImageReference, OsDisk, DataDisks}
VirtualMachineScaleSet : {Id}
TimeCreated            : 11/16/2022 11:02:02 PM
```

Create a scale set with a specific VM instance size

When you created a scale set at the start of the tutorial, a default VM SKU of *Standard_D1_v2* was provided for the VM instances. You can specify a different VM instance size with the `-VMSize` parameter to specify a VM instance size of *Standard_F1*.

Azure PowerShell

```
New-AzVmss `
  -ResourceGroupName "myResourceGroup" `
  -VMScaleSetName "myScaleSet" `
  -OrchestrationMode "Flexible" `
  -VMSize "Standard_F1" `
  -Location "EastUS" `
  -Credential $cred
```

Change the capacity of a scale set

When you created a scale set, two VM instances were deployed by default. To increase or decrease the number of VM instances in the scale set, you can manually change the capacity. The scale set creates or removes the required number of VM instances, then configures the load balancer to distribute traffic.

First, create a scale set object with Get-AzVmss, then specify a new value for `sku.capacity`. To apply the capacity change, use Update-AzVmss. The following example sets the number of VM instances in your scale set to *3*:

Azure PowerShell

```
# Get current scale set
$vmss = Get-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName
"myScaleSet"

# Set and update the capacity of your scale set
$vmss.sku.capacity = 3
Update-AzVmss -ResourceGroupName "myResourceGroup" -Name "myScaleSet" -
VirtualMachineScaleSet $vmss
```

It takes a few minutes to update the capacity of your scale set. To see the number of instances you now have in the scale set, use Get-Az:

Azure PowerShell

```
Get-AzVm -ResourceGroupName "myResourceGroup"
```

The following example output shows that the capacity of the scale set is now *3*:

Output

```
ResourceGroupName                Name Location          VmSize  OsType
ProvisioningState
-----------------                ---- --------          ------  ------    --
---------------
myResourceGroup   myScaleSet_instance1   eastus Standard_DS1_v2 Windows
Succeeded
myResourceGroup   myScaleSet_instance2   eastus Standard_DS1_v2 Windows
Succeeded
myResourceGroup   myScaleSet_instance3   eastus Standard_DS1_v2 Windows
Succeeded
```

Stop and deallocate VM instances in a scale set

To stop individual VM instances, use Stop-AzVm and specify the instance names.

Azure PowerShell

```
Stop-AzVM -ResourceGroupName "myResourceGroup" -name "myScaleSet_instance1"
```

By default, stopped VMs are deallocated and don't incur compute charges. If you wish the VM to remain in a provisioned state when stopped, add the `-StayProvisioned` parameter to the preceding command. Stopped VMs that remain provisioned incur regular compute charges.

Start VM instances in a scale set

To start all the VM instances in a scale set, use Start-AzVmss.

Azure PowerShell

```
Start-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName
"myScaleSet"
```

To start an individual VM instance in a scale set, use Start-AzVM and specify the instance name.

Azure PowerShell

```
Start-AzVM -ResourceGroupName "myResourceGroup" -name "myScaleSet_instance1"
```

Restart VM instances in a scale set

To restart all the VMs in a scale set, use Restart-AzVmss.

Azure PowerShell

```
Restart-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName
"myScaleSet"
```

To restart an individual instance, use [Restart-AzVM] and specify the instance name.

Azure PowerShell

```
Restart-AzVM -ResourceGroupName "myResourceGroup" -name
"myScaleSet_instance1"
```

Clean up resources

When you delete a resource group, all resources contained within, such as the VM instances, virtual network, and disks, are also deleted. The `-Force` parameter confirms

that you wish to delete the resources without an extra prompt to do so. The `-AsJob` parameter returns control to the prompt without waiting for the operation to complete.

Azure PowerShell

```
Remove-AzResourceGroup -Name "myResourceGroup" -Force -AsJob
```

Next steps

In this tutorial, you learned how to perform some basic scale set creation and management tasks with Azure PowerShell:

- ✓ Create a resource group
- ✓ Create a scale set
- ✓ View and use specific VM sizes
- ✓ Manually scale a scale set
- ✓ Perform common scale set management tasks such as stopping, starting and restarting your scale set

Advance to the next tutorial to learn how to connect to your scale set instances.

Use data disks with scale sets

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Virtual Machines

Service: Compute

API Version: 2025-11-01

Operations

Expand table

Assess Patches	Assess patches on the VM.
Attach Detach Data Disks	Attach and detach data disks to/from the virtual machine.
Capture	Captures the VM by copying virtual hard disks of the VM and outputs a template that can be used to create similar VMs.
Convert To Managed Disks	Converts virtual machine disks from blob-based to managed disks. Virtual machine must be stop-deallocated before invoking this operation.
Create Or Update	The operation to create or update a virtual machine. Please note some properties can be set only during virtual machine creation.
Deallocate	Shuts down the virtual machine and releases the compute resources. You are not billed for the compute resources that this virtual machine uses.
Delete	The operation to delete a virtual machine.
Generalize	Sets the OS state of the virtual machine to generalized. It is recommended to sysprep the virtual machine before performing this operation. For Windows, please ...
Get	Retrieves information about the model view or the instance view of a virtual machine.
Install Patches	Installs patches on the VM.
Instance View	Retrieves information about the run-time state of a virtual machine.
List	Lists all of the virtual machines in the specified resource group. Use the nextLink property in the response to get the next page of virtual machines.
List All	Lists all of the virtual machines in the specified subscription. Use the nextLink property in the response to get the next page of virtual machines.
List Available Sizes	Lists all available virtual machine sizes to which the specified virtual machine can be resized.
List By Location	Gets all the virtual machines under the specified subscription for the specified

	location.
migrate To VM Scale Set	Migrate a virtual machine from availability set to Flexible Virtual Machine Scale Set.
Perform Maintenance	The operation to perform maintenance on a virtual machine.
Power Off	The operation to power off (stop) a virtual machine. The virtual machine can be restarted with the same provisioned resources. You are still charged for this vi...
Reapply	The operation to reapply a virtual machine's state.
Redeploy	Shuts down the virtual machine, moves it to a new node, and powers it back on.
Reimage	Reimages (upgrade the operating system) a virtual machine which don't have a ephemeral OS disk, for virtual machines who have a ephemeral OS disk the virtual ma...
Restart	The operation to restart a virtual machine.
Retrieve Boot Diagnostics Data	The operation to retrieve SAS URIs for a virtual machine's boot diagnostic logs.
Run Command	Run command on the VM.
Simulate Eviction	The operation to simulate the eviction of spot virtual machine.
Start	The operation to start a virtual machine.
Update	The operation to update a virtual machine.

Azure Compute SDK for .NET - latest

Packages - latest

Expand table

Reference	Package	Source
Resource Management - Compute	Azure.ResourceManager.Compute	GitHub

Last updated on 04/22/2026

Create and manage Windows VMs in Azure using Java

Article • 08/22/2024

Applies to: ✔ Windows VMs

An Azure Virtual Machine (VM) needs several supporting Azure resources. This article covers creating, managing, and deleting VM resources using Java. You learn how to:

- ✔ Create a Maven project
- ✔ Add dependencies
- ✔ Create credentials
- ✔ Create resources
- ✔ Perform management tasks
- ✔ Delete resources
- ✔ Run the application

It takes about 20 minutes to do these steps.

Create a Maven project

1. If you haven't already done so, install Java.
2. Install Maven.
3. Create a new folder and the project:

```
mkdir java-azure-test
cd java-azure-test

mvn archetype:generate -DgroupId=com.fabrikam -DartifactId=testAzureApp
-DarchetypeArtifactId=maven-archetype-quickstart -
DinteractiveMode=false
```

Add dependencies

1. Under the `testAzureApp` folder, open the `pom.xml` file and add the build configuration to <project> to enable the building of your application:

XML

```xml
<build>
  <plugins>
    <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <configuration>
            <mainClass>com.fabrikam.testAzureApp.App</mainClass>
        </configuration>
    </plugin>
  </plugins>
</build>
```

2. Add the dependencies that are needed to access the Azure Java SDK.

XML

```xml
<dependency>
  <groupId>com.azure</groupId>
  <artifactId>azure-identity</artifactId>
  <version>1.3.6</version>
</dependency>
<dependency>
  <groupId>com.azure.resourcemanager</groupId>
  <artifactId>azure-resourcemanager</artifactId>
  <version>2.8.0</version>
</dependency>
```

3. Save the file.

Set up authentication

Learn how to set up authentication.

Create the management client

1. Open the `App.java` file under `src\main\java\com\fabrikam` and make sure this package statement is at the top:

Java

```java
package com.fabrikam.testAzureApp;
```

2. Create AzureResourceManager:

Java

```
TokenCredential credential = new EnvironmentCredentialBuilder()
            .authorityHost(AzureAuthorityHosts.AZURE_PUBLIC_CLOUD)
            .build();

// Please finish 'Set up authentication' step first to set the four
environment variables: AZURE_SUBSCRIPTION_ID, AZURE_CLIENT_ID,
AZURE_CLIENT_SECRET, AZURE_TENANT_ID
AzureProfile profile = new AzureProfile(AzureEnvironment.AZURE);

AzureResourceManager azureResourceManager =
AzureResourceManager.configure()
        .withLogLevel(HttpLogDetailLevel.BASIC)
        .authenticate(credential, profile)
        .withDefaultSubscription();
```

Create resources

Create the resource group

All resources must be contained in a Resource group.

To specify values for the application and create the resource group, add this code to the try block in the main method:

Java

```
System.out.println("Creating resource group...");
ResourceGroup resourceGroup = azure.resourceGroups()
    .define("myResourceGroup")
    .withRegion(Region.US_EAST)
    .create();
```

Create the availability set

Availability sets make it easier for you to maintain the virtual machines used by your application.

To create the availability set, add this code to the try block in the main method:

Java

```
System.out.println("Creating availability set...");
AvailabilitySet availabilitySet = azure.availabilitySets()
    .define("myAvailabilitySet")
```

```
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .create();
```

Create the public IP address

A Public IP address is needed to communicate with the virtual machine.

To create the public IP address for the virtual machine, add this code to the try block in the main method:

Java

```
System.out.println("Creating public IP address...");
PublicIpAddress publicIPAddress = azure.publicIpAddresses()
    .define("myPublicIP")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withDynamicIP()
    .create();
```

Create the virtual network

A virtual machine must be in a subnet of a Virtual network.

To create a subnet and a virtual network, add this code to the try block in the main method:

Java

```
System.out.println("Creating virtual network...");
Network network = azure.networks()
    .define("myVN")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withAddressSpace("10.0.0.0/16")
    .withSubnet("mySubnet", "10.0.0.0/24")
    .create();
```

Create the network interface

A virtual machine needs a network interface to communicate on the virtual network.

To create a network interface, add this code to the try block in the main method:

```
```

Java

```
System.out.println("Creating network interface...");
NetworkInterface networkInterface = azure.networkInterfaces()
    .define("myNIC")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withExistingPrimaryNetwork(network)
    .withSubnet("mySubnet")
    .withPrimaryPrivateIPAddressDynamic()
    .withExistingPrimaryPublicIPAddress(publicIPAddress)
    .create();
```

Create the virtual machine

Now that you created all the supporting resources, you can create a virtual machine.

To create the virtual machine, add this code to the try block in the main method:

Java

```
System.out.println("Creating virtual machine...");
VirtualMachine virtualMachine = azure.virtualMachines()
    .define("myVM")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withExistingPrimaryNetworkInterface(networkInterface)
    .withLatestWindowsImage("MicrosoftWindowsServer", "WindowsServer",
"2012-R2-Datacenter")
    .withAdminUsername("azureuser")
    .withAdminPassword("Azure12345678")
    .withComputerName("myVM")
    .withExistingAvailabilitySet(availabilitySet)
    .withSize("Standard_DS1")
    .create();
Scanner input = new Scanner(System.in);
System.out.println("Press enter to get information about the VM...");
input.nextLine();
```

> ⓘ **Note**
>
> This tutorial creates a virtual machine running a version of the Windows Server operating system. To learn more about selecting other images, see Navigate and select Azure virtual machine images with Windows PowerShell and the Azure CLI.

If you want to use an existing disk instead of a marketplace image, use this code:

Java

```
Disk managedDisk = azure.disks().define("myosdisk")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")

.withWindowsFromVhd("https://mystorage.blob.core.windows.net/vhds/myosdisk.v
hd")
    .withStorageAccountName("mystorage")
    .withSizeInGB(128)
    .withSku(DiskSkuTypes.PREMIUM_LRS)
    .create();

azure.virtualMachines().define("myVM")
    .withRegion(Region.US_EAST)
    .withExistingResourceGroup("myResourceGroup")
    .withExistingPrimaryNetworkInterface(networkInterface)
    .withSpecializedOSDisk(managedDisk, OperatingSystemTypes.WINDOWS)
    .withExistingAvailabilitySet(availabilitySet)
    .withSize(VirtualMachineSizeTypes.STANDARD_DS1)
    .create();
```

Perform management tasks

During the lifecycle of a virtual machine, you may want to run management tasks such as starting, stopping, or deleting a virtual machine. Additionally, you may want to create code to automate repetitive or complex tasks.

When you need to do anything with the VM, you need to get an instance of it. Add this code to the try block of the main method:

Java

```
VirtualMachine vm =
azure.virtualMachines().getByResourceGroup("myResourceGroup", "myVM");
```

Get information about the VM

To get information about the virtual machine, add this code to the try block in the main method:

Java

```
System.out.println("hardwareProfile");
System.out.println("    vmSize: " + vm.size());
System.out.println("storageProfile");
System.out.println("  imageReference");
```

```
System.out.println("    publisher: " +
vm.storageProfile().imageReference().publisher());
System.out.println("    offer: " +
vm.storageProfile().imageReference().offer());
System.out.println("    sku: " +
vm.storageProfile().imageReference().sku());
System.out.println("    version: " +
vm.storageProfile().imageReference().version());
System.out.println("  osDisk");
System.out.println("    osType: " + vm.storageProfile().osDisk().osType());
System.out.println("    name: " + vm.storageProfile().osDisk().name());
System.out.println("    createOption: " +
vm.storageProfile().osDisk().createOption());
System.out.println("    caching: " +
vm.storageProfile().osDisk().caching());
System.out.println("osProfile");
System.out.println("    computerName: " + vm.osProfile().computerName());
System.out.println("    adminUserName: " + vm.osProfile().adminUsername());
System.out.println("    provisionVMAgent: " +
vm.osProfile().windowsConfiguration().provisionVMAgent());
System.out.println(
        "    enableAutomaticUpdates: " +
vm.osProfile().windowsConfiguration().enableAutomaticUpdates());
System.out.println("networkProfile");
System.out.println("    networkInterface: " +
vm.primaryNetworkInterfaceId());
System.out.println("vmAgent");
System.out.println("  vmAgentVersion: " +
vm.instanceView().vmAgent().vmAgentVersion());
System.out.println("    statuses");
for (InstanceViewStatus status : vm.instanceView().vmAgent().statuses()) {
    System.out.println("    code: " + status.code());
    System.out.println("    displayStatus: " + status.displayStatus());
    System.out.println("    message: " + status.message());
    System.out.println("    time: " + status.time());
}
System.out.println("disks");
for (DiskInstanceView disk : vm.instanceView().disks()) {
    System.out.println("  name: " + disk.name());
    System.out.println("  statuses");
    for (InstanceViewStatus status : disk.statuses()) {
        System.out.println("    code: " + status.code());
        System.out.println("    displayStatus: " + status.displayStatus());
        System.out.println("    time: " + status.time());
    }
}
System.out.println("VM general status");
System.out.println("  provisioningStatus: " + vm.provisioningState());
System.out.println("  id: " + vm.id());
System.out.println("  name: " + vm.name());
System.out.println("  type: " + vm.type());
System.out.println("VM instance status");
for (InstanceViewStatus status : vm.instanceView().statuses()) {
    System.out.println("  code: " + status.code());
    System.out.println("  displayStatus: " + status.displayStatus());
```

```
}
System.out.println("Press enter to continue...");
input.nextLine();
```

Stop the VM

You can stop a virtual machine and keep all its settings, but continue to be charged for it, or you can stop a virtual machine and deallocate it. When a virtual machine is deallocated, all resources associated with it are also deallocated and billing ends for it.

To stop the virtual machine without deallocating it, add this code to the try block in the main method:

Java

```
System.out.println("Stopping vm...");
vm.powerOff();
System.out.println("Press enter to continue...");
input.nextLine();
```

If you want to deallocate the virtual machine, change the PowerOff call to this code:

Java

```
vm.deallocate();
```

Start the VM

To start the virtual machine, add this code to the try block in the main method:

Java

```
System.out.println("Starting vm...");
vm.start();
System.out.println("Press enter to continue...");
input.nextLine();
```

Resize the VM

Many aspects of deployment should be considered when deciding on a size for your virtual machine. For more information, see VM sizes.

To change size of the virtual machine, add this code to the try block in the main method:

Java

```
System.out.println("Resizing vm...");
vm.update()
    .withSize(VirtualMachineSizeTypes.STANDARD_DS2)
    .apply();
System.out.println("Press enter to continue...");
input.nextLine();
```

Add a data disk to the VM

To add a data disk to the virtual machine that is 2 GB in size, has a LUN of 0, and a caching type of ReadWrite, add this code to the try block in the main method:

Java

```
System.out.println("Adding data disk...");
vm.update()
    .withNewDataDisk(2, 0, CachingTypes.READ_WRITE)
    .apply();
System.out.println("Press enter to delete resources...");
input.nextLine();
```

Delete resources

Because you are charged for resources used in Azure, it is always good practice to delete resources that are no longer needed. If you want to delete the virtual machines and all the supporting resources, all you have to do is delete the resource group.

1. To delete the resource group, add this code to the try block in the main method:

   Java

   ```
   System.out.println("Deleting resources...");
   azure.resourceGroups().deleteByName("myResourceGroup");
   ```

2. Save the App.java file.

Run the application

It should take about five minutes for this console application to run completely from start to finish.

1. To run the application, use this Maven command:

```
mvn compile exec:java
```

2. Before you press **Enter** to start deleting resources, you could take a few minutes to verify the creation of the resources in the Azure portal. Click the deployment status to see information about the deployment.

Next steps

- Learn more about using the Azure libraries for Java.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

com.azure.resourcemanager.compute

Package: com.azure.resourcemanager.compute

Maven Artifact: com.azure.resourcemanager:azure-resourcemanager-compute:2.56.3

Package containing the classes for ComputeManagementClient. Compute Client.

Classes

Expand table

ComputeManager	Entry point to Azure compute resource management.

Interfaces

Expand table

ComputeManager.Configurable	The interface allowing configurations to be set.

Use the Azure SDK for Go for control plane operations

Learn how to provision, configure, and manage Azure resources programmatically by using the Azure SDK for Go management libraries. Common control-plane workflows include creating resource groups, managing storage and networking infrastructure, and handling virtual machine (VM) lifecycle operations such as create, start, stop, resize, update, and delete. If you want the higher-level introduction to how management libraries fit into the Azure SDK for Go, start with Overview of the Azure SDK for Go management libraries. This article focuses on the Go control-plane patterns you reuse across services, and links to data plane guidance when the runtime path moves from resource management to working with service data.

What is the Azure control plane?

The Azure control plane is the set of APIs that control the lifecycle of Azure resources - creating, updating, configuring, and deleting them. Every operation you perform in the Azure portal, Azure CLI, or infrastructure-as-code tool ultimately calls these control plane APIs.

The Azure SDK for Go exposes the control plane through a family of `arm*` packages under `github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/`. Each package maps to an Azure resource provider and follows a consistent pattern:

1. Authenticate by using the `azidentity` package.
2. Create a typed client for the resource you want to manage.
3. Call methods on the client to create, read, update, or delete resources.
4. Handle long-running operations by using pollers.

Common scenarios for Go control plane automation include:

- Provisioning infrastructure for deployment pipelines
- Managing VM lifecycle operations such as create, update, delete, start, stop, and resize
- Building custom CLIs and operators for platform teams
- Implementing GitOps-style infrastructure reconciliation
- Automating compliance auditing and drift detection

Authentication

All management operations require an authenticated credential from the azidentity package. The package provides credential types for every environment including local development, CI/CD pipelines, and production workloads running in Azure. All credential types implement the same `azcore.TokenCredential` interface, so you can swap them without changing client code.

After you get a credential, create a client factory for the package and then ask it for the typed client you need:

Go

```go
// Create credential that auto-discovers authentication (Azure CLI, env vars,
managed identity)
cred, err := azidentity.NewDefaultAzureCredential(nil)

// Construct a client factory, then the typed client for management operations
clientFactory, err := armresources.NewClientFactory(subscriptionID, cred, nil)
rgClient := clientFactory.NewResourceGroupsClient()
```

Current `arm*` package docs usually show the client factory pattern because it centralizes shared configuration for related clients. Many packages also expose direct `New<ResourceType>Client(subscriptionID, credential, options)` constructors, but `NewClientFactory(...).New<ResourceType>Client()` is the pattern you'll most often see on pkg.go.dev. For local development, `DefaultAzureCredential` usually picks up your Azure CLI sign-in. In CI/CD and deployed workloads, you can switch to environment-based credentials or managed identity without changing the rest of your client code.

For a full guide on credential types and best practices, see Authentication with the Azure SDK for Go and the azidentity package documentation.

Client packages and typed clients

Management packages are under `github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/<service>/arm<service>`. Install the identity package and only the `arm*` packages you plan to use. For example, if you're only managing VMs and resource groups, you only need `armcompute` and `armresources`. Each package contains clients for the resources in that service. For example, `armcompute` has clients for virtual machines, disks, images, and related compute resources.

A single management package often contains several clients, with each client focused on one resource type or operation group. For example, `armcompute` includes clients for virtual

machines, disks, images, and related resources. After you choose the package for a service, create one client factory and reuse it to create the typed clients that match the resources you want to manage.

Go

```go
clientFactory, err := armcompute.NewClientFactory(subscriptionID, cred, nil)
if err != nil {
    return err
}
vmClient := clientFactory.NewVirtualMachinesClient()
```

This package and client factory pattern is consistent across the `resourcemanager` modules. It's a useful shortcut when you're scanning pkg.go.dev or asking an agent to find the right client for a task.

Long-running operations

Many management operations, such as creating clusters, deleting resource groups, and upgrading infrastructure, run asynchronously. Methods prefixed with `Begin` start the server-side work and return a poller immediately. Your code can decide whether to wait or keep doing other work:

Go

```go
// Start an asynchronous operation (returns immediately)
poller, err := client.BeginCreateOrUpdate(ctx, resourceGroupName, parameters, nil)
if err != nil {
    return err
}

// Block until the operation completes or fails
result, err := poller.PollUntilDone(ctx, nil)
if err != nil {
    return err
}
```

A successful `Begin*` call only means Azure accepted the request. The operation can still fail later while the poller runs. That's why both the initial call and `PollUntilDone` need error handling. Use `PollUntilDone` when you want the simplest flow. Use `poller.Poll` and `poller.Done` when you need custom wait logic or progress reporting.

For more details on patterns, see the Common usage patterns in Azure SDK for Go.

Error handling

Management operations return structured errors you can inspect for specific error codes:

Go

```go
import "github.com/Azure/azure-sdk-for-go/sdk/azcore"

// Check if the error is an Azure service error with structured details
var respErr *azcore.ResponseError
if errors.As(err, &respErr) {
    fmt.Printf("Error code: %s\n", respErr.ErrorCode)
    fmt.Printf("Status code: %d\n", respErr.StatusCode)
}
```

Most `CreateOrUpdate` operations are idempotent. Calling them on an existing resource updates the resource instead of failing.

Provision a resource example

This example shows the common control plane pattern: authenticate, create a resource with tags and timeout, and check the result. Use this pattern as a template for all management operations because the credential, context, and subscription ID pattern applies to all `arm*` clients.

Go

```go
package main

import (
    "context"
    "fmt"
    "log"
    "os"
    "time"

    "github.com/Azure/azure-sdk-for-go/sdk/azcore/to"
    "github.com/Azure/azure-sdk-for-go/sdk/azidentity"
    "github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/resources/armresources"
)

func main() {
    // Read subscription ID from environment (avoid hardcoding)
    subscriptionID := os.Getenv("AZURE_SUBSCRIPTION_ID")
    if subscriptionID == "" {
        log.Fatal("AZURE_SUBSCRIPTION_ID not set")
    }
```

```
    // Create credential that auto-discovers authentication
    cred, err := azidentity.NewDefaultAzureCredential(nil)
    if err != nil {
        log.Fatalf("failed to create credential: %v", err)
    }

    // Set a timeout for the entire operation (prevents hanging indefinitely)
    ctx, cancel := context.WithTimeout(context.Background(), 2*time.Minute)
    defer cancel()

    // Create a client factory for this management package
    clientFactory, err := armresources.NewClientFactory(subscriptionID, cred, nil)
    if err != nil {
        log.Fatalf("failed to create client factory: %v", err)
    }

    // Create the typed client for resource groups
    rgClient := clientFactory.NewResourceGroupsClient()

    // Many ARM models use pointer fields for optional values.
    resp, err := rgClient.CreateOrUpdate(ctx, "example-rg",
armresources.ResourceGroup{
        Location: to.Ptr("eastus"),
        Tags: map[string]*string{
            "env":  to.Ptr("dev"),
            "team": to.Ptr("platform"),
        },
    }, nil)
    if err != nil {
        log.Fatalf("failed to create or update resource group: %v", err)
    }

    fmt.Printf("resource group %s ready in %s\n", *resp.Name, *resp.Location)
}
```

Resource groups

The armresources package manages resource groups - the fundamental organizational containers in Azure. Every Azure resource exists within a resource group, making this the starting point for any provisioning workflow.

Use it to create and update resource groups with location and tags, list groups across a subscription, and delete groups along with all contained resources. Resource group creation is synchronous and idempotent. Deletion is asynchronous and permanent.

Listing resource groups also introduces an important control-plane pattern: many read operations use pagers. When you enumerate resource groups or other large ARM collections, create a pager and iterate until `pager.More()` returns `false`.

Go

```
pager := rgClient.NewListPager(nil)
for pager.More() {
    page, err := pager.NextPage(ctx)
    if err != nil {
        return err
    }

    for _, group := range page.ResourceGroupListResult.Value {
        fmt.Println(*group.Name)
    }
}
```

Resource group management code sample.

For a getting started guide, see the armresources package documentation.

Virtual machines

The armcompute package is a canonical control-plane example because VM management is mostly lifecycle work: create or update a VM, start or stop it, resize it, and delete it. In Go, these workflows use the same `DefaultAzureCredential`, `context.Context`, and client factory pattern shown in the resource group example, so once that pattern is in place you can apply it across compute operations without changing your authentication approach.

If you need a quick starting point, create the compute client factory and then ask it for the typed VM client:

Go

```
clientFactory, err := armcompute.NewClientFactory(subscriptionID, cred, nil)
if err != nil {
    return err
}
vmClient := clientFactory.NewVirtualMachinesClient()
```

For full VM samples and operation-specific guidance, see the existing virtual machine management samples and the armcompute package documentation. Use those references for complete request models and long-running operation details instead of duplicating large VM templates in this article.

Key Vault

The armkeyvault package manages the lifecycle of Azure Key Vault instances. This package handles the control plane for vault infrastructure. Use the separate `azsecrets`, `azkeys`, and `azcertificates` data plane packages to read and write secrets, keys, and certificates.

Use this package to provision vaults with the appropriate SKU and security settings, such as soft delete and purge protection. You can also manage access policies for principals, configure network access and private endpoints, and enable diagnostic logging. You can integrate vault provisioning into application onboarding workflows.

Key Vault management code sample.

For runtime-side Key Vault clients, see Use the Azure SDK for Go for data plane operations.

For a getting started guide, see the armkeyvault package documentation.

AKS clusters

The armcontainerservice package manages Azure Kubernetes Service clusters across their full lifecycle.

Use this package to create clusters with configurable networking, Kubernetes version, and managed identity. You can add and scale node pools, upgrade control plane and node versions, enable add-ons like Azure Policy and monitoring, and query cluster health for operational dashboards. All cluster operations are long-running and follow the poller pattern.

AKS management code sample.

For a getting started guide, see the armcontainerservice package documentation.

RBAC and authorization

The armauthorization package manages Azure Role-Based Access Control. Use it to automate least-privilege access policies across subscriptions and resource groups.

Use it to list and search built-in roles, assign roles to principals (users, service principals, managed identities, or groups) at any scope, create custom role definitions with fine-grained permissions, and audit assignments for compliance reporting and drift detection. Assign roles to groups rather than individuals, and use built-in roles where possible.

For a getting started guide, see the armauthorization package documentation.

Virtual networks and network security

The armnetwork package manages Azure virtual networking infrastructure.

Use it to create virtual networks and subnets, configure network security groups with inbound and outbound rules, set up private endpoints for PaaS services, automate network peering across regions, and implement hub-and-spoke topologies programmatically.

Network management code sample.

For a getting started guide, see the armnetwork package documentation.

Container registry

The armcontainerregistry package manages Azure Container Registry instances.

Use it to provision registries with the appropriate SKU and geo-replication, configure authentication (admin, service principal, or managed identity), manage webhooks for CI/CD, enable vulnerability scanning, and apply retention policies to images. You often use Container Registry alongside Azure Kubernetes Service. First, provision the registry, and then reference it during cluster creation.

Container Registry management code sample.

For a getting started guide, see the armcontainerregistry package documentation.

Storage accounts

The armstorage package manages Azure Storage accounts.

Use it to create storage accounts with the right performance tier and redundancy, manage access keys and shared access signatures, configure blob lifecycle policies, and set up diagnostic logging. Storage accounts are a common dependency for many applications, so automating their provisioning and configuration is a common control-plane scenario.

Storage account management code sample.

For a getting started guide, see the armstorage package documentation.

Next steps

- Overview of the Azure SDK for Go management libraries
- Use the Azure SDK for Go for data plane operations
- Azure SDK for Go authentication
- Azure SDK for Go samples on GitHub

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/10/2026

Azure Resource Graph sample queries for Azure Virtual Machines

Article • 08/22/2024

This page is a collection of Azure Resource Graph sample queries for Azure Virtual Machines.

Sample queries

Count of OS update installation done

Returns a list of status of OS update installation runs done for your machines in last seven days.

Kusto

```
PatchAssessmentResources
| where type !has 'softwarepatches'
| extend machineName = tostring(split(id, '/', 8)), resourceType =
tostring(split(type, '/', 0)), tostring(rgName = split(id, '/', 4))
| extend prop = parse_json(properties)
| extend lTime = todatetime(prop.lastModifiedDateTime), OS =
tostring(prop.osType), installedPatchCount =
tostring(prop.installedPatchCount), failedPatchCount =
tostring(prop.failedPatchCount), pendingPatchCount =
tostring(prop.pendingPatchCount), excludedPatchCount =
tostring(prop.excludedPatchCount), notSelectedPatchCount =
tostring(prop.notSelectedPatchCount)
| where lTime > ago(7d)
| project lTime, RunID=name,machineName, rgName, resourceType, OS,
installedPatchCount, failedPatchCount, pendingPatchCount,
excludedPatchCount, notSelectedPatchCount
```

Azure CLI

Azure CLI

```
az graph query -q "PatchAssessmentResources | where type !has
'softwarepatches' | extend machineName = tostring(split(id, '/', 8)),
resourceType = tostring(split(type, '/', 0)), tostring(rgName =
split(id, '/', 4)) | extend prop = parse_json(properties) | extend lTime
= todatetime(prop.lastModifiedDateTime), OS = tostring(prop.osType),
installedPatchCount = tostring(prop.installedPatchCount),
failedPatchCount = tostring(prop.failedPatchCount), pendingPatchCount =
```

```
tostring(prop.pendingPatchCount), excludedPatchCount =
tostring(prop.excludedPatchCount), notSelectedPatchCount =
tostring(prop.notSelectedPatchCount) | where lTime > ago(7d) | project
lTime, RunID=name,machineName, rgName, resourceType, OS,
installedPatchCount, failedPatchCount, pendingPatchCount,
excludedPatchCount, notSelectedPatchCount"
```

Count of virtual machines by availability state and Subscription ID

Returns the count of virtual machines (type `Microsoft.Compute/virtualMachines`) aggregated by their availability state across each of your subscriptions.

Kusto

```
HealthResources
| where type =~ 'microsoft.resourcehealth/availabilitystatuses'
| summarize count() by subscriptionId, AvailabilityState =
tostring(properties.availabilityState)
```

Azure CLI

Azure CLI

```
az graph query -q "HealthResources | where type =~
'microsoft.resourcehealth/availabilitystatuses' | summarize count() by
subscriptionId, AvailabilityState =
tostring(properties.availabilityState)"
```

Count of virtual machines by power state

Returns count of virtual machines (type `Microsoft.Compute/virtualMachines`) categorized according to their power state. For more information on power states, see Power states overview.

Kusto

```
Resources
| where type == 'microsoft.compute/virtualmachines'
| summarize count() by PowerState =
tostring(properties.extended.instanceView.powerState.code)
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type ==
'microsoft.compute/virtualmachines' | summarize count() by PowerState =
tostring(properties.extended.instanceView.powerState.code)"
```

Count virtual machines by OS type

Building on the previous query, we're still limiting by Azure resources of type `Microsoft.Compute/virtualMachines`, but are no longer limiting the number of records returned. Instead, we used `summarize` and `count()` to define how to group and aggregate the values by property, which in this example is `properties.storageProfile.osDisk.osType`. For an example of how this string looks in the full object, see explore resources - virtual machine discovery.

Kusto

```
Resources
| where type =~ 'Microsoft.Compute/virtualMachines'
| summarize count() by tostring(properties.storageProfile.osDisk.osType)
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type =~
'Microsoft.Compute/virtualMachines' | summarize count() by
tostring(properties.storageProfile.osDisk.osType)"
```

Count virtual machines by OS type with extend

A different way to write the 'Count virtual machines by OS type' query is to `extend` a property and give it a temporary name for use within the query, in this case **os**. **os** is then used by `summarize` and `count()` as in the referenced example.

Kusto

```
Resources
| where type =~ 'Microsoft.Compute/virtualMachines'
| extend os = properties.storageProfile.osDisk.osType
| summarize count() by tostring(os)
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type =~
'Microsoft.Compute/virtualMachines' | extend os =
properties.storageProfile.osDisk.osType | summarize count() by
tostring(os)"
```

Get all New alerts from the past 30 days

This query provides a list of all the user's New alerts, from the past 30 days.

Kusto

```
iotsecurityresources
| where type == 'microsoft.iotsecurity/locations/devicegroups/alerts'
| where todatetime(properties.startTimeUtc) > ago(30d) and properties.status
== 'New'
```

Azure CLI

Azure CLI

```
az graph query -q "iotsecurityresources | where type ==
'microsoft.iotsecurity/locations/devicegroups/alerts' | where
todatetime(properties.startTimeUtc) > ago(30d) and properties.status ==
'New'"
```

Get virtual machine scale set capacity and size

This query looks for virtual machine scale set resources and gets various details including the virtual machine size and the capacity of the scale set. The query uses the `toint()` function to cast the capacity to a number so that it can be sorted. Finally, the columns are renamed into custom named properties.

Kusto

```
Resources
| where type=~ 'microsoft.compute/virtualmachinescalesets'
| where name contains 'contoso'
| project subscriptionId, name, location, resourceGroup, Capacity =
toint(sku.capacity), Tier = sku.name
| order by Capacity desc
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type=~
'microsoft.compute/virtualmachinescalesets' | where name contains
'contoso' | project subscriptionId, name, location, resourceGroup,
Capacity = toint(sku.capacity), Tier = sku.name | order by Capacity
desc"
```

List all extensions installed on a virtual machine

First, this query uses `extend` on the virtual machines resource type to get the ID in uppercase (`toupper()`) the ID, get the operating system name and type, and get the virtual machine size. Getting the resource ID in uppercase is a good way to prepare to join to another property. Then, the query uses `join` with `kind` as `leftouter` to get virtual machine extensions by matching an uppercase `substring` of the extension ID. The portion of the ID before `/extensions/\<ExtensionName\>` is the same format as the virtual machines ID, so we use this property for the `join`. `summarize` is then used with `make_list` on the name of the virtual machine extension to combine the name of each extension where *ID*, *OSName*, *OSType*, and *VMSize* are the same into a single array property. Lastly, we `order by` lowercase *OSName* with `asc`. By default, `order by` is descending.

Kusto

```
Resources
| where type == 'microsoft.compute/virtualmachines'
| extend
  JoinID = toupper(id),
  OSName = tostring(properties.osProfile.computerName),
  OSType = tostring(properties.storageProfile.osDisk.osType),
  VMSize = tostring(properties.hardwareProfile.vmSize)
| join kind=leftouter(
  Resources
```

```
  | where type == 'microsoft.compute/virtualmachines/extensions'
  | extend
    VMId = toupper(substring(id, 0, indexof(id, '/extensions'))),
    ExtensionName = name
) on $left.JoinID == $right.VMId
| summarize Extensions = make_list(ExtensionName) by id, OSName, OSType,
VMSize
| order by tolower(OSName) asc
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type ==
'microsoft.compute/virtualmachines' | extend JoinID = toupper(id),
OSName = tostring(properties.osProfile.computerName), OSType =
tostring(properties.storageProfile.osDisk.osType), VMSize =
tostring(properties.hardwareProfile.vmSize) | join kind=leftouter(
Resources | where type == 'microsoft.compute/virtualmachines/extensions'
| extend  VMId = toupper(substring(id, 0, indexof(id, '/extensions'))),
ExtensionName = name ) on \$left.JoinID == \$right.VMId | summarize
Extensions = make_list(ExtensionName) by id, OSName, OSType, VMSize |
order by tolower(OSName) asc"
```

List available OS updates for all your machines grouped by update category

Returns a list of pending OS for your machines.

Kusto

```
PatchAssessmentResources
| where type !has 'softwarepatches'
| extend prop = parse_json(properties)
| extend lastTime = properties.lastModifiedDateTime
| extend updateRollupCount =
prop.availablePatchCountByClassification.updateRollup, featurePackCount =
prop.availablePatchCountByClassification.featurePack, servicePackCount =
prop.availablePatchCountByClassification.servicePack, definitionCount =
prop.availablePatchCountByClassification.definition, securityCount =
prop.availablePatchCountByClassification.security, criticalCount =
prop.availablePatchCountByClassification.critical, updatesCount =
prop.availablePatchCountByClassification.updates, toolsCount =
prop.availablePatchCountByClassification.tools, otherCount =
prop.availablePatchCountByClassification.other, OS = prop.osType
| project lastTime, id, OS, updateRollupCount, featurePackCount,
```

```
servicePackCount, definitionCount, securityCount, criticalCount,
updatesCount, toolsCount, otherCount
```

Azure CLI

Azure CLI

```
az graph query -q "PatchAssessmentResources | where type !has
'softwarepatches' | extend prop = parse_json(properties) | extend
lastTime = properties.lastModifiedDateTime | extend updateRollupCount =
prop.availablePatchCountByClassification.updateRollup, featurePackCount
= prop.availablePatchCountByClassification.featurePack, servicePackCount
= prop.availablePatchCountByClassification.servicePack, definitionCount
= prop.availablePatchCountByClassification.definition, securityCount =
prop.availablePatchCountByClassification.security, criticalCount =
prop.availablePatchCountByClassification.critical, updatesCount =
prop.availablePatchCountByClassification.updates, toolsCount =
prop.availablePatchCountByClassification.tools, otherCount =
prop.availablePatchCountByClassification.other, OS = prop.osType |
project lastTime, id, OS, updateRollupCount, featurePackCount,
servicePackCount, definitionCount, securityCount, criticalCount,
updatesCount, toolsCount, otherCount"
```

List of Linux OS update installation done

Returns a list of status of Linux Server - OS update installation runs done for your machines in last seven days.

Kusto

```
PatchAssessmentResources
| where type has 'softwarepatches' and properties has 'version'
| extend machineName = tostring(split(id, '/', 8)), resourceType =
tostring(split(type, '/', 0)), tostring(rgName = split(id, '/', 4)),
tostring(RunID = split(id, '/', 10))
| extend prop = parse_json(properties)
| extend lTime = todatetime(prop.lastModifiedDateTime), patchName =
tostring(prop.patchName), version = tostring(prop.version),
installationState = tostring(prop.installationState), classifications =
tostring(prop.classifications)
| where lTime > ago(7d)
| project lTime, RunID, machineName, rgName, resourceType, patchName,
version, classifications, installationState
| sort by RunID
```

Azure CLI

Azure CLI

```
az graph query -q "PatchAssessmentResources | where type has
'softwarepatches' and properties has 'version' | extend machineName =
tostring(split(id, '/', 8)), resourceType = tostring(split(type, '/',
0)), tostring(rgName = split(id, '/', 4)), tostring(RunID = split(id,
'/', 10)) | extend prop = parse_json(properties) | extend lTime =
todatetime(prop.lastModifiedDateTime), patchName =
tostring(prop.patchName), version = tostring(prop.version),
installationState = tostring(prop.installationState), classifications =
tostring(prop.classifications) | where lTime > ago(7d) | project lTime,
RunID, machineName, rgName, resourceType, patchName, version,
classifications, installationState | sort by RunID"
```

List of virtual machines and associated availability states by resource IDs

Returns the latest list of virtual machines (type `Microsoft.Compute/virtualMachines`) aggregated by availability state. The query also provides the associated resource ID based on `properties.targetResourceId`, for easy debugging and mitigation. Availability states can be one of four values: Available, Unavailable, Degraded, and Unknown. For more information on what each of the availability states mean, see Azure Resource Health overview.

Kusto

```
HealthResources
| where type =~ 'microsoft.resourcehealth/availabilitystatuses'
| summarize by ResourceId = tolower(tostring(properties.targetResourceId)),
AvailabilityState = tostring(properties.availabilityState)
```

Azure CLI

Azure CLI

```
az graph query -q "HealthResources | where type =~
'microsoft.resourcehealth/availabilitystatuses' | summarize by
ResourceId = tolower(tostring(properties.targetResourceId)),
AvailabilityState = tostring(properties.availabilityState)"
```

List of virtual machines by availability state and power state with Resource IDs and resource Groups

Returns list of virtual machines (type `Microsoft.Compute/virtualMachines`) aggregated on their power state and availability state to provide a cohesive state of health for your virtual machines. The query also provides details on the resource group and resource ID associated with each entry for detailed visibility into your resources.

Kusto

```
Resources
| where type =~ 'microsoft.compute/virtualmachines'
| project resourceGroup, Id = tolower(id), PowerState = tostring(
properties.extended.instanceView.powerState.code)
| join kind=leftouter (
  HealthResources
  | where type =~ 'microsoft.resourcehealth/availabilitystatuses'
  | where tostring(properties.targetResourceType) =~
'microsoft.compute/virtualmachines'
  | project targetResourceId =
tolower(tostring(properties.targetResourceId)), AvailabilityState =
tostring(properties.availabilityState))
  on $left.Id == $right.targetResourceId
| project-away targetResourceId
| where PowerState != 'PowerState/deallocated'
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type =~
'microsoft.compute/virtualmachines' | project resourceGroup, Id =
tolower(id), PowerState = tostring(
properties.extended.instanceView.powerState.code) | join kind=leftouter
( HealthResources | where type =~
'microsoft.resourcehealth/availabilitystatuses' | where
tostring(properties.targetResourceType) =~
'microsoft.compute/virtualmachines' | project targetResourceId =
tolower(tostring(properties.targetResourceId)), AvailabilityState =
tostring(properties.availabilityState)) on \$left.Id ==
\$right.targetResourceId | project-away targetResourceId | where
PowerState != 'PowerState/deallocated'"
```

List of virtual machines that aren't Available by Resource Ids

Returns the latest list of virtual machines (type `Microsoft.Compute/virtualMachines`) aggregated by their availability state. The populated list only highlights virtual machines whose availability state isn't *Available* to ensure you're aware of all the concerning states your virtual machines are in. When all your virtual machines are Available, you can expect to receive no results.

Kusto

```
HealthResources
| where type =~ 'microsoft.resourcehealth/availabilitystatuses'
| where tostring(properties.availabilityState) != 'Available'
| summarize by ResourceId = tolower(tostring(properties.targetResourceId)),
AvailabilityState = tostring(properties.availabilityState)
```

Azure CLI

Azure CLI

```
az graph query -q "HealthResources | where type =~
'microsoft.resourcehealth/availabilitystatuses' | where
tostring(properties.availabilityState) != 'Available' | summarize by
ResourceId = tolower(tostring(properties.targetResourceId)),
AvailabilityState = tostring(properties.availabilityState)"
```

List of Windows Server OS update installation done

Returns a list of status of Windows Server - OS update installation runs done for your machines in last seven days.

Kusto

```
PatchAssessmentResources
| where type has 'softwarepatches' and properties !has 'version'
| extend machineName = tostring(split(id, '/', 8)), resourceType =
tostring(split(type, '/', 0)), tostring(rgName = split(id, '/', 4)),
tostring(RunID = split(id, '/', 10))
| extend prop = parse_json(properties)
| extend lTime = todatetime(prop.lastModifiedDateTime), patchName =
tostring(prop.patchName), kbId = tostring(prop.kbId), installationState =
tostring(prop.installationState), classifications =
tostring(prop.classifications)
| where lTime > ago(7d)
| project lTime, RunID, machineName, rgName, resourceType, patchName, kbId,
classifications, installationState
| sort by RunID
```

Azure CLI

Azure CLI

```
az graph query -q "PatchAssessmentResources | where type has
'softwarepatches' and properties !has 'version' | extend machineName =
tostring(split(id, '/', 8)), resourceType = tostring(split(type, '/',
0)), tostring(rgName = split(id, '/', 4)), tostring(RunID = split(id,
'/', 10)) | extend prop = parse_json(properties) | extend lTime =
todatetime(prop.lastModifiedDateTime), patchName =
tostring(prop.patchName), kbId = tostring(prop.kbId), installationState
= tostring(prop.installationState), classifications =
tostring(prop.classifications) | where lTime > ago(7d) | project lTime,
RunID, machineName, rgName, resourceType, patchName, kbId,
classifications, installationState | sort by RunID"
```

List virtual machines with their network interface and public IP

This query uses two `leftouter` `join` commands to bring together virtual machines created with the Resource Manager deployment model, their related network interfaces, and any public IP address related to those network interfaces.

Kusto

```
Resources
| where type =~ 'microsoft.compute/virtualmachines'
| extend nics=array_length(properties.networkProfile.networkInterfaces)
| mv-expand nic=properties.networkProfile.networkInterfaces
| where nics == 1 or nic.properties.primary =~ 'true' or isempty(nic)
| project vmId = id, vmName = name,
vmSize=tostring(properties.hardwareProfile.vmSize), nicId = tostring(nic.id)
| join kind=leftouter (
  Resources
  | where type =~ 'microsoft.network/networkinterfaces'
  | extend ipConfigsCount=array_length(properties.ipConfigurations)
  | mv-expand ipconfig=properties.ipConfigurations
  | where ipConfigsCount == 1 or ipconfig.properties.primary =~ 'true'
  | project nicId = id, publicIpId =
tostring(ipconfig.properties.publicIPAddress.id))
  on nicId
| project-away nicId1
| summarize by vmId, vmName, vmSize, nicId, publicIpId
| join kind=leftouter (
  Resources
  | where type =~ 'microsoft.network/publicipaddresses'
  | project publicIpId = id, publicIpAddress = properties.ipAddress)
```

```
on publicIpId
| project-away publicIpId1
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type =~
'microsoft.compute/virtualmachines' | extend
nics=array_length(properties.networkProfile.networkInterfaces) | mv-
expand nic=properties.networkProfile.networkInterfaces | where nics == 1
or nic.properties.primary =~ 'true' or isempty(nic) | project vmId = id,
vmName = name, vmSize=tostring(properties.hardwareProfile.vmSize), nicId
= tostring(nic.id) | join kind=leftouter ( Resources | where type =~
'microsoft.network/networkinterfaces' | extend
ipConfigsCount=array_length(properties.ipConfigurations) | mv-expand
ipconfig=properties.ipConfigurations | where ipConfigsCount == 1 or
ipconfig.properties.primary =~ 'true' | project nicId = id, publicIpId =
tostring(ipconfig.properties.publicIPAddress.id)) on nicId | project-
away nicId1 | summarize by vmId, vmName, vmSize, nicId, publicIpId |
join kind=leftouter ( Resources | where type =~
'microsoft.network/publicipaddresses' | project publicIpId = id,
publicIpAddress = properties.ipAddress) on publicIpId | project-away
publicIpId1"
```

Show all virtual machines ordered by name in descending order

To list only virtual machines (which are type `Microsoft.Compute/virtualMachines`), we can match the property **type** in the results. Similar to the previous query, `desc` changes the `order by` to be descending. The `=~` in the type match tells Resource Graph to be case insensitive.

Kusto

```
Resources
| project name, location, type
| where type =~ 'Microsoft.Compute/virtualMachines'
| order by name desc
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | project name, location, type | where type
=~ 'Microsoft.Compute/virtualMachines' | order by name desc"
```

Show first five virtual machines by name and their OS type

This query uses `top` to only retrieve five matching records that are ordered by name. The type of the Azure resource is `Microsoft.Compute/virtualMachines`. `project` tells Azure Resource Graph which properties to include.

Kusto

```
Resources
| where type =~ 'Microsoft.Compute/virtualMachines'
| project name, properties.storageProfile.osDisk.osType
| top 5 by name desc
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type =~
'Microsoft.Compute/virtualMachines' | project name,
properties.storageProfile.osDisk.osType | top 5 by name desc"
```

Summarize virtual machine by the power states extended property

This query uses the extended properties on virtual machines to summarize by power states.

Kusto

```
Resources
| where type == 'microsoft.compute/virtualmachines'
| summarize count() by
tostring(properties.extended.instanceView.powerState.code)
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type ==
'microsoft.compute/virtualmachines' | summarize count() by
tostring(properties.extended.instanceView.powerState.code)"
```

Virtual machines matched by regex

This query looks for virtual machines that match a regular expression (known as *regex*). The **matches regex @** allows us to define the regex to match, which is `^Contoso(.*)[0-9]+$`. That regex definition is explained as:

- `^` - Match must start at the beginning of the string.
- `Contoso` - The case-sensitive string.
- `(.*)` - A subexpression match:
  - `.` - Matches any single character (except a new line).
  - `*` - Matches previous element zero or more times.
- `[0-9]` - Character group match for numbers 0 through 9.
- `+` - Matches previous element one or more times.
- `$` - Match of the previous element must occur at the end of the string.

After matching by name, the query projects the name and orders by name ascending.

Kusto

```
Resources
| where type =~ 'microsoft.compute/virtualmachines' and name matches regex
@'^Contoso(.*)[0-9]+$'
| project name
| order by name asc
```

Azure CLI

Azure CLI

```
az graph query -q "Resources | where type =~
'microsoft.compute/virtualmachines' and name matches regex
@'^Contoso(.*)[0-9]+\$' | project name | order by name asc"
```

Next steps

- Learn more about the query language.
- Learn more about how to explore resources.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Red Hat workloads on Azure

Article • 10/28/2024

Applies to: ✔ Linux VMs

Red Hat workloads are supported through a variety of offerings on Azure. Red Hat Enterprise Linux (RHEL) images are at the core of RHEL workloads, as is the Red Hat Update Infrastructure (RHUI). Red Hat JBoss EAP is also supported on Azure, see Red Hat JBoss EAP.

For more information about running RHEL workloads on Azure, see the following video: https://www.youtube-nocookie.com/embed/11arYjecfxU?si=m0X-28xd1-zYUoqG

Red Hat Enterprise Linux images

Azure offers a wide offering of RHEL images on Azure. These images are made available through two different licensing models: pay-as-you-go and bring-your-own-subscription (BYOS). New RHEL images on Azure are published when new RHEL versions are released and updated throughout their lifecycles, as necessary.

Pay-as-you-go images

Azure offers a variety of RHEL pay-as-you-go images. These images come properly entitled for RHEL and are attached to a source of updates (Red Hat Update Infrastructure). These images charge a premium fee for the RHEL entitlement and updates. RHEL pay-as-you-go image variants include:

- RHEL
- RHEL for SAP
- RHEL for SAP with High Availability (HA) and Update Services
- RHEL with High Availability (HA) and Update Services

You might want to use the pay-as-you-go images if you don't want to worry about paying separately for the appropriate number of subscriptions.

Red Hat Gold Images

Azure also offers Red Hat Gold Images (`rhel-byos`). These images might be useful to customers who have existing Red Hat subscriptions and want to use them in Azure. You're required to enable your existing Red Hat subscriptions for Red Hat Cloud Access

before you can use them in Azure. Access to these images is granted automatically when your Red Hat subscriptions are enabled for Cloud Access and meet the eligibility requirements. Using these images allows a customer to avoid double billing that might be incurred from using the pay-as-you-go images.

- Learn how to enable your Red Hat subscriptions for Cloud Access with Azure.
- Learn how to locate Red Hat Gold Images in the Azure portal, the Azure CLI, or PowerShell cmdlet.

> **Note**
>
> Double billing is incurred when a user pays twice for RHEL subscriptions. This scenario usually happens when a customer uses Red Hat Subscription-Manager to attach an entitlement on a RHEL pay-as-you-go VM. For example, a customer who uses Subscription-Manager to attach an entitlement for SAP packages on a RHEL pay-as-you-go image is indirectly double billed because they pay twice for RHEL. They pay once through the pay-as-you-go premium fee and once through their SAP subscription. This scenario doesn't happen to BYOS image users.

Generation 2 images

Generation 2 virtual machines (VMs) provide some newer features compared to Generation 1 VMs. For more information, see the Generation 2 documentation. The key difference from a RHEL image perspective is that Generation 2 VMs use a UEFI instead of BIOS firmware interface. They also use a GUID Partition Table (GPT) instead of a master boot record (MBR) on boot time. Use of a GPT allows for, among other things, OS disk sizes larger than 2 TB. In addition, the Mv2 series VMs run only on Generation 2 images.

RHEL Generation 2 images are available in the Azure Marketplace. Look for "gen2" in the image SKU in the list of all images that appears when you use the Azure CLI. Go to the **Advanced** tab in the VM deploy process to deploy a Generation 2 VM.

Red Hat Update Infrastructure

Azure provides Red Hat Update Infrastructure only for pay-as-you-go RHEL VMs. RHUI is effectively a mirror of the Red Hat CDNs but is only accessible to the Azure pay-as-you-go RHEL VMs. You have access to the appropriate packages depending on which RHEL image you've deployed. For example, a RHEL for SAP image has access to the SAP packages in addition to base RHEL packages.

RHUI update behavior

RHEL images connected to RHUI update by default to the latest minor version of RHEL when a `yum update` is run. This behavior means that a RHEL 7.4 VM might get upgraded to RHEL 7.7 if a `yum update` operation is run on it. This behavior is by design for RHUI. To mitigate this upgrade behavior, switch from regular RHEL repositories to Extended Update Support repositories.

Red Hat JBoss EAP

Microsoft and Azure have partnered to develop a variety of solutions for running Red Hat Middleware on Azure. Learn more about JBoss EAP on Azure Virtual Machines, Azure App service and Azure Red Hat OpenShift at Red Hat JBoss EAP on Azure.

Next steps

- Learn more about RHEL images on Azure.
- Learn more about Red Hat Update Infrastructure.
- Learn more about the Red Hat Gold Image (rhel-byos) offer.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Ubuntu Server to Ubuntu Pro in-place upgrade on Azure

Article • 03/17/2025

Customers can now upgrade their Ubuntu Server (version 16.04 or higher) virtual machines to Ubuntu Pro without redeployment or downtime. This method is recommended for customers wishing to convert their servers from versions that are End of Life (EOL) status.

> **ⓘ Important**
>
> EOL versions are not officially supported unless they have the PRO entitlement. For an updated list of versions that are EOL, check the official Canonical Release Cycle page.

> **⚠ Warning**
>
> - Only Ubuntu images on the Azure Marketplace **published by Canonical** can be converted to "UBUNTU_PRO"
> - Images published by any other vendors or custom images are not supported

What's Ubuntu Pro?

Ubuntu Pro is a cross-cloud OS, optimized for Azure, and security maintained for 10 years. The secure use of open-source software allows the operating system to use the latest technologies while meeting internal governance and compliance requirements. Ubuntu Pro provides Extended Security Maintenance (ESM) for infrastructure and applications support, providing security patching for all Ubuntu packages.

Why developers and devops choose Ubuntu Pro for Azure

- Access to security updates for 23,000+ packages including Apache Kafka, NGINX, MongoDB, Redis, and PostgreSQL integrated into system tools (for example Azure Update Manager, apt)

- Security hardening and audit tools (CIS) to establish a security baseline across your systems (helps meeting the Azure Linux Security Baseline policy)
- Federal Information Processing Standard (FIPS) 140-2 certified modules
- Common Criteria (CC) EAL2 provisioning packages
- Kernel Live patch: kernel patches delivered immediately, without the need to reboot
- Optimized performance: optimized kernel, with improved boot speed, outstanding runtime performance, and advanced device support
- 10-year security maintenance: Ubuntu Pro 18.04 Long-Term Support (LTS) provides security maintenance until April 2028
- Developer friendly: Ubuntu offers developers the latest libraries and tools to innovate with the latest technologies
- Nonstop security: Canonical publishes images ensuring security is present from the moment an instance launches
- Portability: Ubuntu is available in all regions with content mirrors to reduce the need to go across regions or out to the Internet for updates
- Consistent experience across platforms: from edge to multicloud, Ubuntu provides the same experience regardless of the platform. It ensures consistency of your CI/CD pipelines and management mechanisms.

> **ⓘ Important**
>
> Converting to Ubuntu Pro is an **irreversible** process.

Convert to Ubuntu Pro using the Azure CLI

The following command enables Ubuntu Pro on a virtual machine in Azure:

Azure

```
az vm update -g myResourceGroup -n myVmName --license-type UBUNTU_PRO
```

Execute these commands inside the VM:

Bash

```
sudo apt install ubuntu-pro-client
sudo pro auto-attach
```

> **ⓘ Important**
>
> The change of the "licenseType" property may take some time to propagate thru the system. If the auto-attach process fails, wait for a few minutes, and try again. If the auto-attach process continues to fail, open a support ticket with Microsoft. If the `pro --version` is lower than 28, execute this command:

Bash

```
sudo apt install ubuntu-pro-client
```

Validate the license

use the `pro status --all` command to validate the license:

Expected output:

Output

```
SERVICE      ENTITLED    STATUS    DESCRIPTION
cc-eal       yes         disabled  Common Criteria EAL2 Provisioning
Packages
cis          yes         disables  Security compliance and audit tools
esm-apps     yes         enabled   Expanded Security Maintenance and audit
tools
esm-infra    yes         enabled   Expanded Security Maintenance for
infrastructure
fips         yes         disabled  NIST-certified core packages
fips-updates yes         disabled  NIST-certified core packages with
priority security updates
livepatch    yes         enabled   Canonical Livepatch service
```

Create an Ubuntu Pro VM using the Azure CLI

You can create a new VM using the Ubuntu Server images and apply Ubuntu Pro at the time of creation.

Both of the following commands enable Ubuntu Pro on a virtual machine in Azure. Choose which to run based on whether you want to use SSH keys or an administrator user and password.

To create the VM and use **SSH keys**:

Azure

```
az vm create -g myResourceGroup -n myVmName --generate-ssh-keys --license-type UBUNTU_PRO --image ubuntu2204
```

To create the VM and use an **administrator user and password**:

Azure

```
az vm create -g myResourceGroup -n myVmName --admin-username Username --license-type UBUNTU_PRO --image ubuntu2204
```

> **Note**
>
> For more information on ways to connect to Linux machines, see Connect to a Linux VM.

Execute these commands inside the VM:

Bash

```
sudo apt install ubuntu-pro-client
sudo pro auto-attach
```

> **Note**
>
> For systems with advantage tools using version 28 or higher, installed the system will perform a `pro attach` during a reboot.

Check licensing model using the Azure CLI

> **Tip**
>
> You can query the metadata in *Azure Instance Metadata Service* to determine the virtual machine's *licenseType* value. You can use the `az vm get-instance-view` command to check the status. Look for the *licenseType* field in the response. If the field exists and the value is UBUNTU_PRO, your virtual machine has Ubuntu Pro enabled. Learn more about attested metadata.

Azure

```
az vm get-instance-view -g MyResourceGroup -n MyVm
```

Billing

Visit the pricing calculator for more details on Ubuntu Pro pricing. To cancel the Pro subscription during the preview period, open a support ticket through the Azure portal.

Next steps after launching an Ubuntu Pro VM

With the availability of outbound internet access, Ubuntu Pro automatically enables premium features including Live Patch and Extended Security Maintenance for Main and Universe repositories. Should any specific hardening be required, check `usg` to harden your servers for CIP and FIPS tutorials. Learn more about networking requirements (such as egress traffic, endpoints and ports) by reading Ubuntu Pro Client network requirements.

Frequently Asked Questions

Does shutting down the machine stop billing?

Launching Ubuntu Pro from Azure Marketplace is you pay as you go and only charges for running machines.

Are there volume discounts?

Yes. Contact your Microsoft sales representative.

Are Reserved Instances available?

Yes.

If the customer doesn't perform the `auto attach` function, will they still get attached to pro on reboot?

If the customer doesn't perform the *auto attach*, they still get the Pro attached upon reboot. However, this action only applies if they're using version 28 of the Pro client.

- For Ubuntu 20.04+, this process works as expected.
- For Ubuntu 16.04 and 18.04, this process doesn't work due to older versions of the Pro client installed.

Do images converted to UbuntuPro or UbuntuPro images include support from Canonical?

No, at present, neither UbuntuPro images nor virtual machines converted to UbuntuPro come with support from Canonical.

- For issues that are reproducible in Azure and affect all customers, Microsoft Azure support will collaborate with Canonical to resolve the problem together.
- For dedicated support, please use the following link to contact Canonical.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy OpenShift Container Platform 4.x in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Deployment of OpenShift Container Platform (OCP) 4.2 is now supported in Azure via the Installer-Provisioned Infrastructure (IPI) model. The landing page for trying OpenShift 4 is try.openshift.com. To install OCP 4.2 in Azure, visit the Red Hat OpenShift Cluster Manager page. Red Hat credentials are required to access this site.

Notes

- A Microsoft Entra service principal (SP) is required to install and run OCP 4.x in Azure
  - The SP must be granted the API permission of **Application.ReadWrite.OwnedBy** for Azure Active Directory Graph
  - A Microsoft Entra tenant administrator must grant Admin Consent for this API permission to take effect
  - The SP must be granted **Contributor** and **User Access Administrator** roles to the subscription
- The installation model for OCP 4.x is different than 3.x and there are no Azure Resource Manager templates available for deploying OCP 4.x in Azure
- If issues are encountered during the installation process, contact the appropriate company (Microsoft or Red Hat)

Expand table

Issue Description	Contact Point
Azure specific issues (Microsoft Entra ID, SP, Azure Subscription, etc.)	Microsoft
OpenShift-specific issues (Installation failures / errors, Red Hat subscription, etc.)	Red Hat

Next steps

- Getting started with OpenShift Container Platform

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

OpenShift Container Platform or OKD deployment to Azure Stack Hub

Applies to: ✔ Linux VMs ✔ Flexible scale sets

You can deploy OpenShift in Azure Stack Hub. Some key differences exist between Azure and Azure Stack Hub, so deployment and capabilities differ slightly.

Currently, the Azure Cloud Provider doesn't work in Azure Stack Hub. You can't use disk attach for persistent storage in Azure Stack Hub. Instead, you can configure other storage options such as Network File System (NFS), Internet Small Computer Systems Interface (iSCSI), and GlusterFS. Or, you can enable Container Native Storage (CNS) and use GlusterFS for persistent storage. If you enable CNS, the deployment adds three more nodes with storage for GlusterFS usage.

Deploy OpenShift 4.x On Azure Stack Hub

Red Hat manages the Red Hat Enterprise Linux CoreOS (RHCOS) image for OpenShift 4.x. The deployment process gets the image from a Red Hat endpoint. As a result, you don't need to get an image from the Azure Stack hub Marketplace.

Follow the steps in the OpenShift documentation at Installing a cluster on Azure Stack Hub using ARM templates.

> ⚠ **Warning**
>
> If you have an issue with OpenShift, contact Red Hat for support.

Next steps

Installing a cluster on Azure Stack Hub using ARM templates.

Last updated on 11/24/2025

SAP on Azure documentation

SAP on Azure provides multiple options for running, managing and monitoring SAP workloads on Azure.



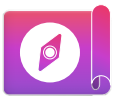
OVERVIEW
What SAP on Azure offerings are available?



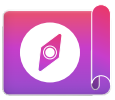
GET STARTED
SAP on Azure workloads



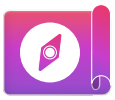
OVERVIEW
Azure Center for SAP solutions



OVERVIEW
SAP on Azure deployment automation...



OVERVIEW
What is Azure Monitor for SAP solutions?



OVERVIEW
What is SAP HANA on Azure Large...

SAP workloads on Azure VMs

Overview

- How do you run SAP workloads on Azure VMs?
- Documentation change log
- SAP certifications on Azure
- Supported SAP scenarios on Azure VMs
- Supported SAP software on Azure VMs

Concept

- Planning guidance
- Azure storage types for SAP workloads
- High availability for SAP components
- Azure Availability Zones for SAP workloads

How-to guides

- How to deploy SAP on Azure VMs
- How to deploy DBMS on Azure VMs

Azure Center / Monitor for SAP solutions

Overview

Quickstart

How-to guide

- Prepare for deployment

- What is Azure Center for SAP solutions?
- FAQ about Azure Center for SAP solutions
- What is Azure Monitor for SAP solutions?
- FAQ about Azure Monitor for SAP solutions

- Deploy Azure Monitor for SAP solutions (Azure portal)
- Deploy Azure Monitor for SAP solutions (PowerShell)

- Deploy SAP infrastructure
- Install SAP software
- Register an existing SAP system
- Start and stop SAP systems
- Manage a Virtual Instance for SAP solutions
- Get quality checks and insights
- View cost analysis for SAP system

SAP on Azure deployment automation framework

Overview

- What is SAP on Azure deployment automation framework?

Concept

- Supported platform and features
- Get started with the automation framework
- Plan your deployment of the automation framework

Quickstart

- Configure Azure DevOps Services for automation framework

SAP on Azure Large Instances / SAP and Microsoft integrations

Integrations - overview

- What scenarios are available and how do I get started?
- How do you integrate with SAP RISE?

Integrations - How-to-guide

- Enable SAP Principal Propagation and SSO for Power Platform
- How to configure M365 Exchange Online for SAP
- How to onboard SAP Edge Integration Cell with Azure
- How to configure SAP printing with Microsoft Universal Print

Integrations - Quickstart

- Deploy an ERP extension using SAP's Cloud SDK on Azure in one click
- Use free developer accounts for SAP BTP, M365 and Azure
- Use SAP ABAP platform and SAP BTP, ABAP environment to integrate with Microsoft

Oracle on Azure Documentation

Oracle applications and solutions on Azure.

Oracle on Azure Documentation

OVERVIEW

Oracle on Azure

Oracle on Azure Virtual Machines

REFERENCE

Oracle on Azure Virtual Machines

Oracle AI Database@Azure

REFERENCE

Oracle AI Database@Azure

Onboard with Oracle AI Database@Azure

Region availability

Network planning

Known issues

Manage access to Oracle AI Database@Azure

Plan IP address space

Delegated subnet limits

FAQs

Azure gaming documentation

Learn how to build games using Microsoft Azure services.

Cloud Build Pipelines

OVERVIEW

Azure Cloud Builds

Incredibuild - Scale out your cloud builds

HOW-TO GUIDE

Perforce - Using AAD together with Perforce

Perforce - Logging on with a Managed Identity

Perforce - Set up a Proxy in Azure

AI in games

OVERVIEW

Azure AI services

Content Moderation

Text Translation

Text to speech

Speech to text

Game streaming

OVERVIEW

Unreal Pixel Streaming in Azure

HOW-TO GUIDE

Deploying Unreal Pixel Streaming in Azure

Unreal Pixel Streaming at Scale in Azure

Set up a STUN and TURN Server in Azure

Game hosting and servers

OVERVIEW

Managed game servers

Managed game services

OVERVIEW

Azure PlayFab

Achievements and leaderboards

Authentication

Game servers

Game data and tools for analytics

Game insights

In-game economy

Matchmaking and lobby

Text/video chat and peer-to-peer networking

User generated content

Intro to Azure for developers

OVERVIEW

Overview

Manage access

HOW-TO GUIDE

Set up Azure account

REFERENCE

Glossary

Install the Elastic Stack (ELK) on an Azure VM

Article • 05/02/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article walks you through how to deploy Elasticsearch, Logstash, and Kibana, on an Ubuntu VM in Azure. To see the Elastic Stack in action, you can optionally connect to Kibana and work with some sample logging data.

Additionally, you can follow the Deploy Elastic on Azure Virtual Machines module for a more guided tutorial on deploying Elastic on Azure Virtual Machines.

In this tutorial you learn how to:

- ✔ Create an Ubuntu VM in an Azure resource group
- ✔ Install Elasticsearch, Logstash, and Kibana on the VM
- ✔ Send sample data to Elasticsearch with Logstash
- ✔ Open ports and work with data in the Kibana console

This deployment is suitable for basic development with the Elastic Stack. For more on the Elastic Stack, including recommendations for a production environment, see the Elastic documentation and the Azure Architecture Center.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Get started with Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Authenticate to Azure using Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use and manage extensions with the Azure CLI.

    - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

- This article requires version 2.0.4 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a resource group

In this section, environment variables are declared for use in subsequent commands. A random suffix is appended to resource names for uniqueness.

Bash

```
export RANDOM_SUFFIX=$(openssl rand -hex 3)
export RESOURCE_GROUP="myResourceGroup$RANDOM_SUFFIX"
export REGION="eastus2"
az group create --name $RESOURCE_GROUP --location $REGION
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxx/resourceGroups/myResourceGroupxxxxxx",
  "location": "eastus",
  "managedBy": null,
  "name": "myResourceGroupxxxxxx",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```


Create a virtual machine

This section creates a VM with a unique name, while also generating SSH keys if they do not already exist. A random suffix is appended to ensure uniqueness.

Bash

```
export VM_NAME="myVM$RANDOM_SUFFIX"
az vm create \
    --resource-group $RESOURCE_GROUP \
    --name $VM_NAME \
    --image Ubuntu2204 \
```

```
    --admin-username azureuser \
    --generate-ssh-keys
```

When the VM has been created, the Azure CLI shows information similar to the following example. Take note of the publicIpAddress. This address is used to access the VM.

Results:

JSON


```
{
  "fqdns": "",
  "id":
"/subscriptions/xxxxx/resourceGroups/myResourceGroupxxxxxx/providers/Microsoft.Com
pute/virtualMachines/myVMxxxxxx",
  "location": "eastus",
  "macAddress": "xx:xx:xx:xx:xx:xx",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "x.x.x.x",
  "resourceGroup": "$RESOURCE_GROUP"
}
```


SSH into your VM

If you don't already know the public IP address of your VM, run the following command to list it:

Azure CLI

```
az network public-ip list --resource-group $RESOURCE_GROUP --query [].ipAddress
```

Use the following command to create an SSH session with the virtual machine. Substitute the correct public IP address of your virtual machine. In this example, the IP address is *40.68.254.142*.

Bash

```
export PUBLIC_IP_ADDRESS=$(az network public-ip list --resource-group
$RESOURCE_GROUP --query [].ipAddress -o tsv)
```

Install the Elastic Stack

In this section, you import the Elasticsearch signing key and update your APT sources list to include the Elastic package repository. This is followed by installing the Java runtime environment which is required for the Elastic Stack components.

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-5.x.list
"
```

Install the Java Virtual Machine on the VM and configure the JAVA_HOME variable:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
sudo apt install -y openjdk-8-jre-headless
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
"
```

Run the following command to update Ubuntu package sources and install Elasticsearch, Kibana, and Logstash.

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
  wget -qO elasticsearch.gpg https://artifacts.elastic.co/GPG-KEY-elasticsearch
  sudo mv elasticsearch.gpg /etc/apt/trusted.gpg.d/

  echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee
/etc/apt/sources.list.d/elastic-7.x.list

  sudo apt update

  # Now install the ELK stack
  sudo apt install -y elasticsearch kibana logstash
"
```

> ⓘ **Note**
>
> Detailed installation instructions, including directory layouts and initial configuration, are maintained in Elastic's documentation

Start Elasticsearch

Start Elasticsearch on your VM with the following command:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
sudo systemctl start elasticsearch.service
"
```

This command produces no output, so verify that Elasticsearch is running on the VM with this curl command:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
sleep 11
sudo curl -XGET 'localhost:9200/'
"
```

If Elasticsearch is running, you see output like the following:

Results:

JSON

```
{
  "name" : "w6Z4NwR",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "SDzCajBoSK2EkXmHvJVaDQ",
  "version" : {
    "number" : "5.6.3",
    "build_hash" : "1a2f265",
    "build_date" : "2017-10-06T20:33:39.012Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.1"
  },
  "tagline" : "You Know, for Search"
}
```

Start Logstash and add data to Elasticsearch

Start Logstash with the following command:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
sudo systemctl start logstash.service
```

```
"
```

Test Logstash to make sure it's working correctly:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
# Time-limited test with file input instead of stdin
sudo timeout 11s /usr/share/logstash/bin/logstash -e 'input { file { path =>
"/var/log/syslog" start_position => "end" sincedb_path => "/dev/null"
stat_interval => "1 second" } } output { stdout { codec => json } }' || echo
"Logstash test completed"
"
```

This is a basic Logstash pipeline that echoes standard input to standard output.

Set up Logstash to forward the kernel messages from this VM to Elasticsearch. To create the Logstash configuration file, run the following command which writes the configuration to a new file called vm-syslog-logstash.conf:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
cat << 'EOF' > vm-syslog-logstash.conf
input {
    stdin {
        type => "stdin-type"
    }

    file {
        type => "syslog"
        path => [ "/var/log/*.log", "/var/log/*/*.log", "/var/log/messages",
"/var/log/syslog" ]
        start_position => "beginning"
    }
}

output {

    stdout {
        codec => rubydebug
    }
    elasticsearch {
        hosts  => "localhost:9200"
    }
}
EOF
"
```

Test this configuration and send the syslog data to Elasticsearch:

Bash

```
# Run Logstash with the configuration for 60 seconds
sudo timeout 60s /usr/share/logstash/bin/logstash -f vm-syslog-logstash.conf &
LOGSTASH_PID=$!

# Wait for data to be processed
echo "Processing logs for 60 seconds..."
sleep 65

# Verify data was sent to Elasticsearch with proper error handling
echo "Verifying data in Elasticsearch..."
ES_COUNT=$(sudo curl -s -XGET 'localhost:9200/_cat/count?v' | tail -n 1 | awk
'{print $3}' 2>/dev/null || echo "0")

# Make sure ES_COUNT is a number or default to 0
if ! [[ "$ES_COUNT" =~ ^[0-9]+$ ]]; then
    ES_COUNT=0
    echo "Warning: Could not get valid document count from Elasticsearch"
fi

echo "Found $ES_COUNT documents in Elasticsearch"

if [ "$ES_COUNT" -gt 0 ]; then
    echo "✅ Logstash successfully sent data to Elasticsearch"
else
    echo "❌ No data found in Elasticsearch, there might be an issue with Logstash
configuration"
fi
```

You see the syslog entries in your terminal echoed as they are sent to Elasticsearch. Use CTRL+C to exit out of Logstash once you've sent some data.

Start Kibana and visualize the data in Elasticsearch

Edit the Kibana configuration file (/etc/kibana/kibana.yml) and change the IP address Kibana listens on so you can access it from your web browser:

text

```
server.host: "0.0.0.0"
```

Start Kibana with the following command:

Bash

```
ssh azureuser@$PUBLIC_IP_ADDRESS -o StrictHostKeyChecking=no "
sudo systemctl start kibana.service
```

```
"
```

Open port 5601 from the Azure CLI to allow remote access to the Kibana console:

Azure CLI

```
az vm open-port --port 5601 --resource-group $RESOURCE_GROUP --name $VM_NAME
```

Next steps

In this tutorial, you deployed the Elastic Stack into a development VM in Azure. You learned how to:

- ✓ Create an Ubuntu VM in an Azure resource group
- ✓ Install Elasticsearch, Logstash, and Kibana on the VM
- ✓ Send sample data to Elasticsearch from Logstash
- ✓ Open ports and work with data in the Kibana console

What is confidential computing?

Article • 05/07/2025

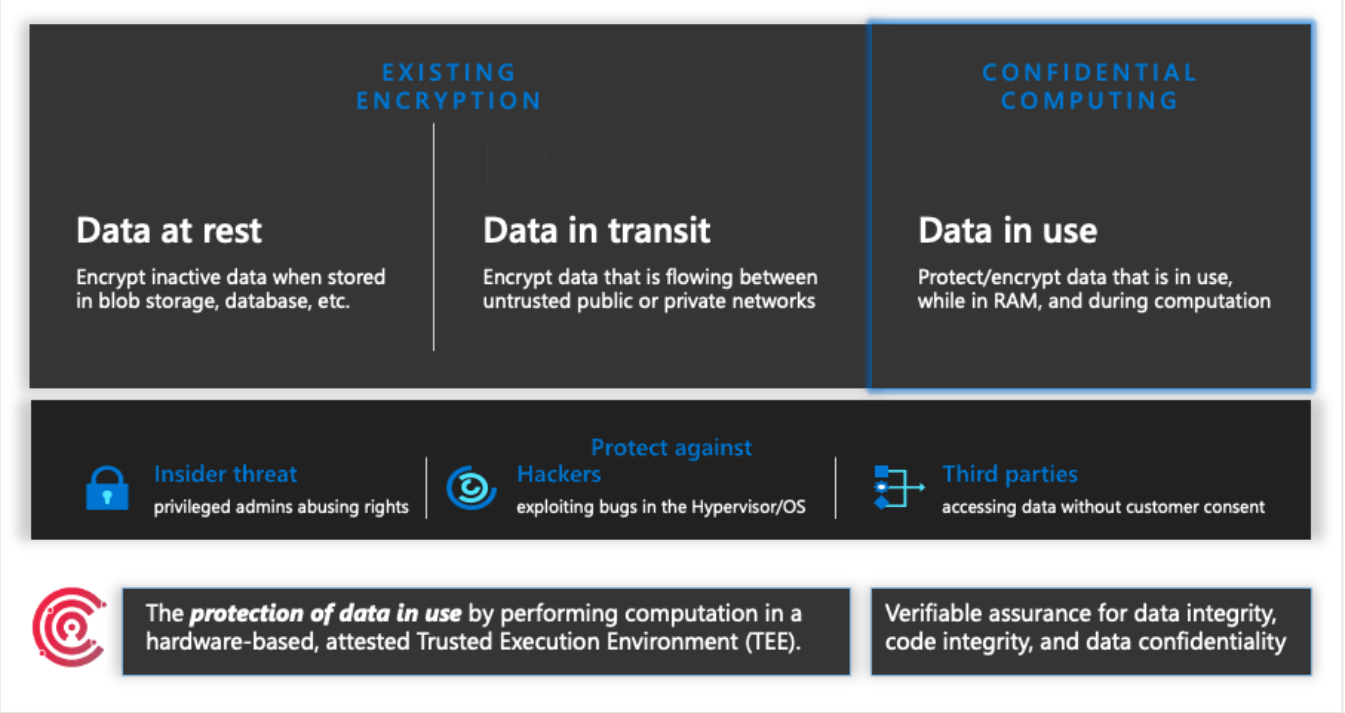
Confidential computing is an industry term established by the Confidential Computing Consortium (CCC), which is part of the Linux Foundation. The CCC defines confidential computing in this way:

"Confidential Computing protects data in use by performing computation in a hardware-based, attested Trusted Execution Environment.

"These secure and isolated environments prevent unauthorized access or modification of applications and data while they are in use, thereby increasing the security level of organizations that manage sensitive and regulated data."

Microsoft is one of the founding members of the CCC and provides Trusted Execution Environments (TEEs) in Azure based on this CCC definition.

Reducing the attack surface



Azure already encrypts data at rest and in transit. Confidential computing helps to protect data in use, including protection for cryptographic keys. Azure confidential computing helps customers prevent unauthorized access to data in use, including from the cloud operator, by processing data in a hardware-based and attested TEE. When Azure confidential computing is enabled and properly configured, Microsoft can't access unencrypted customer data.

The threat model aims to reduce trust or remove the ability for a cloud provider operator or other actors in the tenant's domain from accessing code and data while it's being executed.

Azure uses a hardware root of trust that isn't controlled by the cloud provider, which is designed to prevent unauthorized access or modification of the environment.

When confidential computing is used with data encryption at rest and in transit, it extends data protections further to protect data while confidential computing is in use. This capability is beneficial for organizations that seek further protection for sensitive data and applications hosted in cloud environments.

Industry partnership

The CCC brings together hardware vendors, cloud providers, and software developers to accelerate the adoption of TEE technologies and standards. Microsoft helped to cofound the CCC in 2019 and has chaired both the governing body and the Technical Advisory Council.

Related content

- To support your journey to confidentiality, explore offerings that span infrastructure as a service (IaaS), platform as a service (PaaS), and developer tools.
- To learn more about confidential computing, see Overview of Azure confidential computing.

SQL Server on Azure VM documentation

Find concepts, quickstarts, tutorials, and samples for SQL Server installed to Azure virtual machines, both Windows and Linux.

SQL Server on Azure VM

QUICKSTART

Create Windows SQL VM (portal)

Create Windows SQL VM (PowerShell)

Create Linux SQL VM (portal)

VIDEO

SQL Server on Azure VM overview

OVERVIEW

What's new?

What is SQL Server on Windows VMs?

What is SQL Server on Linux VMs?

Security considerations

Performance guidelines

Pricing guidance

Manage

CONCEPT

SQL Server IaaS Agent extension

Manage with Azure portal

Register with SQL VM resource provider

Automated patching

Change license type

Change edition of SQL Server

Move to new region

Integrate with Azure Key Vault

Business continuity

OVERVIEW

High availability & disaster recovery

Backup and restore

Availability groups

HOW-TO GUIDE

Availability group (Azure CLI)

Clusterless availability group

FCI (Storage Spaces Direct)

FCI (Premium File Share)

TUTORIAL

Availability group (manual)

Fencing availability group (RHEL)

Fencing availability group (SLES)

Fencing availability group (Ubuntu)

Availability group listener

Learn Azure SQL

TRAINING

Azure SQL for beginners

Azure SQL fundamentals

Azure SQL hands-on labs

Azure SQL bootcamp

Educational SQL resources

Reference

DEPLOY

Azure portal

Azure CLI

PowerShell samples

ARM template samples

DOWNLOAD

SQL Server Management Studio (SSMS)

SQL Server Data Tools

Visual Studio 2022

REFERENCE

Migration guide

Transact-SQL (T-SQL)

Azure CLI

PowerShell

REST API

High-performance computing (HPC) on Azure

Introduction to HPC

https://www.youtube-nocookie.com/embed/rKURT32faJk

High-performance computing (HPC), also called *big compute*, uses a large number of CPU or GPU-based computers to solve complex mathematical tasks.

Many industries use HPC to solve some of their most difficult problems. These include workloads such as:

- Genomics
- Oil and gas simulations
- Finance
- Semiconductor design
- Engineering
- Weather modeling

How is HPC different on the cloud?

One of the primary differences between an on-premises HPC system and one in the cloud is the ability for resources to dynamically be added and removed as they're needed. Dynamic scaling removes compute capacity as a bottleneck and instead allow customers to right size their infrastructure for the requirements of their jobs.

The following articles provide more detail about this dynamic scaling capability.

- Big Compute Architecture Style
- Autoscaling best practices

Implementation checklist

As you're looking to implement your own HPC solution on Azure, ensure you've reviewed the following topics:

- ✓ Choose the appropriate architecture based on your requirements

- ✓ Know which compute options is right for your workload
- ✓ Identify the right storage solution that meets your needs
- ✓ Decide how you're going to manage all your resources
- ✓ Optimize your application for the cloud
- ✓ Secure your Infrastructure

Infrastructure

There are many infrastructure components that are necessary to build an HPC system. Compute, storage, and networking provide the underlying components, no matter how you choose to manage your HPC workloads.

Compute

Azure offers a range of sizes that are optimized for both CPU & GPU intensive workloads.

CPU-based virtual machines

- Linux VMs
- Windows VMs

GPU-enabled virtual machines

N-series VMs feature NVIDIA GPUs designed for compute-intensive or graphics-intensive applications including artificial intelligence (AI) learning and visualization.

- Linux VMs
- Windows VMs

Storage

Large-scale Batch and HPC workloads have demands for data storage and access that exceed the capabilities of traditional cloud file systems. There are many solutions that manage both the speed and capacity needs of HPC applications on Azure:

- Azure NetApp Files
- Storage Optimized Virtual Machines
- Blob, table, and queue storage
- Azure SMB File storage
- Azure Managed Lustre

For more information comparing Lustre, GlusterFS, and BeeGFS on Azure, review the Parallel Files Systems on Azure e-book and the Lustre on Azure blog.

Networking

H16r, H16mr, A8, and A9 VMs can connect to a high throughput back-end RDMA network. This network can improve the performance of tightly coupled parallel applications running under Microsoft Message Passing Interface better known as MPI or Intel MPI.

- RDMA Capable Instances
- Virtual Network
- ExpressRoute

Management

Do-it-yourself

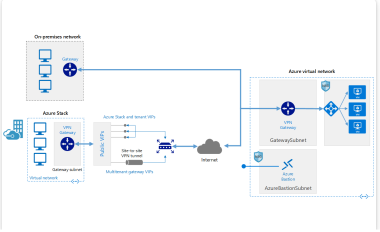
Building an HPC system from scratch on Azure offers a significant amount of flexibility, but it is often very maintenance intensive.

1. Set up your own cluster environment in Azure virtual machines or Virtual Machine Scale Sets.
2. Use Azure Resource Manager templates to deploy leading workload managers, infrastructure, and applications.
3. Choose HPC and GPU VM sizes that include specialized hardware and network connections for MPI or GPU workloads.
4. Add high-performance storage for I/O-intensive workloads.

Hybrid and cloud Bursting

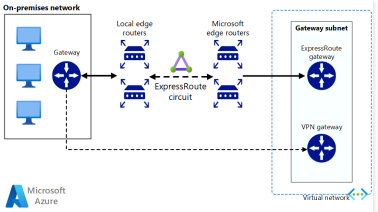
If you have an existing on-premises HPC system that you'd like to connect to Azure, there are several resources to help get you started.

First, review the Options for connecting an on-premises network to Azure article in the documentation. From there, you can find additional information on these connectivity options:



Connect an on-premises network to Azure using a VPN gateway

This reference architecture shows how to extend an on-premises network to Azure, using a site-to-site virtual private network (VPN).



Connect an on-premises network to Azure using ExpressRoute with VPN failover

Implement a highly available and secure site-to-site network architecture that spans an Azure virtual network and an on-premises network connected using ExpressRoute with VPN gateway failover.

Once network connectivity is securely established, you can start using cloud compute resources on-demand with the bursting capabilities of your existing workload manager.

Marketplace solutions

There are many workload managers offered in the Microsoft Marketplace.

- SUSE Linux Enterprise Server for HPC
- Azure Data Science VM for Windows and Linux
- UberCloud

Azure Batch

Azure Batch is a platform service for running large-scale parallel and HPC applications efficiently in the cloud. Azure Batch schedules compute-intensive work to run on a managed

pool of virtual machines, and can automatically scale compute resources to meet the needs of your jobs.

SaaS providers or developers can use the Batch SDKs and tools to integrate HPC applications or container workloads with Azure, stage data to Azure, and build job execution pipelines.

In Azure Batch all the services are running on the Cloud. The following image shows how the architecture looks with Azure Batch, having the scalability and job schedule configurations running in the Cloud while the results and reports can be sent to your on-premises environment.



Azure CycleCloud

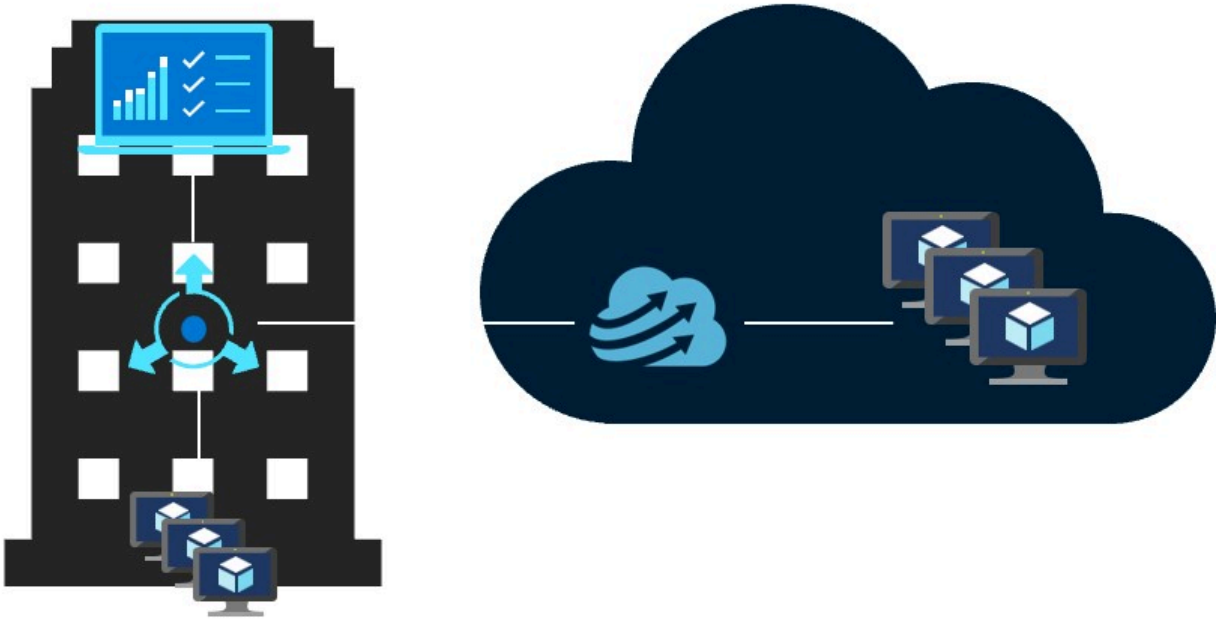
Azure CycleCloud Provides the simplest way to manage HPC workloads using any scheduler (like Slurm, Grid Engine, HPC Pack, HTCondor, LSF, PBS Pro, or Symphony), on Azure

CycleCloud allows you to:

- Deploy full clusters and other resources, including scheduler, compute VMs, storage, networking, and cache
- Orchestrate job, data, and cloud workflows
- Give admins full control over which users can run jobs, as well as where and at what cost
- Customize and optimize clusters through advanced policy and governance features, including cost controls, Active Directory integration, monitoring, and reporting
- Use your current job scheduler and applications without modification
- Take advantage of built-in autoscaling and battle-tested reference architectures for a wide range of HPC workloads and industries

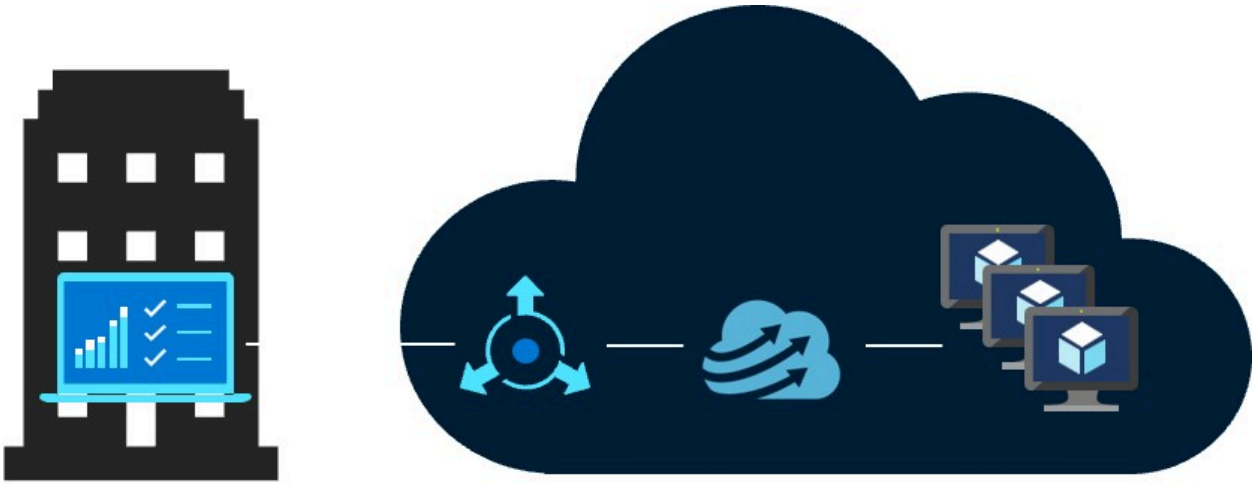
Hybrid / cloud bursting model

In this Hybrid example diagram, we can see clearly how these services are distributed between the cloud and the on-premises environment. Having the opportunity to run jobs in both workloads.



Cloud native model

The following cloud native model example diagram shows how the workload in the cloud will handle everything while still conserving the connection to the on-premises environment.



Comparison chart

Expand table

Feature	Azure Batch	Azure CycleCloud
Scheduler	Batch APIs and tools and command-line scripts in the Azure portal (Cloud Native).	Use standard HPC schedulers such as Slurm, PBS Pro, LSF, Grid Engine, and HTCondor, or extend CycleCloud autoscaling plugins to work with your own scheduler.

Feature	Azure Batch	Azure CycleCloud
Compute Resources	Software as a Service Nodes – Platform as a Service	Platform as a Service Software – Platform as a Service
Monitor Tools	Azure Monitor	Azure Monitor, Grafana
Customization	Custom image pools, Third Party images, Batch API access.	Use the comprehensive RESTful API to customize and extend functionality, deploy your own scheduler, and support into existing workload managers
Integration	Data Factory in Microsoft Fabric, Azure Data Factory, Azure CLI	Built-In CLI for Windows and Linux
User type	Developers	Classic HPC administrators and users
Work Type	Batch, Workflows	Tightly coupled (Message Passing Interface/MPI).
Windows Support	Yes	Varies, depending on scheduler choice

Workload managers

The following are examples of cluster and workload managers that can run in Azure infrastructure. Create stand-alone clusters in Azure VMs or burst to Azure VMs from an on-premises cluster.

- Alces Flight Compute
- Altair PBS Works
- Rescale
- Altair Grid Engine
- Microsoft HPC Pack
  - HPC Pack for Windows
  - HPC Pack for Linux

Containers

Containers can also be used to manage some HPC workloads. Services like the Azure Kubernetes Service (AKS) make it simple to deploy a managed Kubernetes cluster in Azure.

- Azure Kubernetes Service (AKS)
- Container Registry

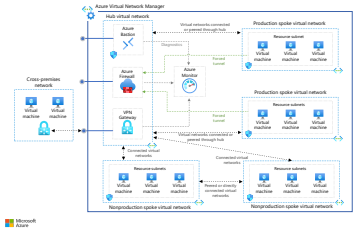
Cost management

Managing your HPC cost on Azure can be done through a few different ways. Ensure you've reviewed the Azure purchasing options to find the method that works best for your organization.

Security

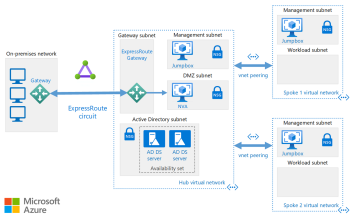
For an overview of security best practices on Azure, review the Azure Security Documentation.

In addition to the network configurations available in the Cloud Bursting section, you can implement a hub/spoke configuration to isolate your compute resources:



Implement a hub-spoke network topology in Azure

The hub is a virtual network (VNet) in Azure that acts as a central point of connectivity to your on-premises network. The spokes are VNets that peer with the hub, and can be used to isolate workloads.



Implement a hub-spoke network topology with shared services in Azure

This reference architecture builds on the hub-spoke reference architecture to include shared services in the hub that can be consumed by all spokes.

HPC applications

Run custom or commercial HPC applications in Azure. Several examples in this section are benchmarked to scale efficiently with additional VMs or compute cores. Visit the Microsoft Marketplace for ready-to-deploy solutions.

> **Note**
>
> Check with the vendor of any commercial application for licensing or other restrictions for running in the cloud. Not all vendors offer pay-as-you-go licensing. You might need a licensing server in the cloud for your solution, or connect to an on-premises license server.

Engineering applications

- MATLAB Distributed Computing Server
- StarCCM+

Graphics and rendering

- Autodesk Maya, 3ds Max, and Arnold on Azure Batch

AI and deep learning

- Microsoft Cognitive Toolkit

MPI providers

- Microsoft MPI

Remote visualization

Run GPU-powered virtual machines in Azure in the same region as the HPC output for the lowest latency, access, and to visualize remotely through Azure Virtual Desktop.

- GPU-optimized virtual machine sizes
- Configure GPU acceleration for Azure Virtual Desktop

Performance benchmarks

- Compute benchmarks

Other important information

- Ensure your vCPU quota has been increased before attempting to run large-scale workloads.

Next steps

For the latest announcements, see the following resources:

- Microsoft HPC and Batch team blog
- Visit the Azure blog.

Microsoft Batch Examples

These tutorials will provide you with details on running applications on Microsoft Batch:

- Get started developing with Batch
- Use low-priority VMs with Batch
- Use compute-intensive VMs in Batch pools

Related resources

- Big compute architecture style

Last updated on 03/12/2026

High-performance computing on InfiniBand enabled HB-series and N-series VMs

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure's InfiniBand enabled H-series and N-series VMs are designed to deliver leadership-class performance, Message Passing Interface (MPI) scalability, and cost efficiency for a variety of real-world HPC and AI workloads. These high-performance computing (HPC) optimized VMs are used to solve some of the most computationally intensive problems in science and engineering such as: fluid dynamics, earth modeling, weather simulations, etc.

These articles describe how to get started on the InfiniBand-enabled H-series and N-series VMs on Azure as well as optimal configuration of the HPC and AI workloads on the VMs for scalability.

Features and capabilities

The InfiniBand enabled HB-series and N-series VMs are designed to provide the best HPC performance, MPI scalability, and cost efficiency for HPC workloads. See HB-series and N-series VMs to learn more about the features and capabilities of the VMs.

RDMA and InfiniBand

RDMA capable HB-series and N-series VMs communicate over the low latency and high bandwidth InfiniBand network. The RDMA capability over such an interconnect is critical to boost the scalability and performance of distributed-node HPC and AI workloads. The InfiniBand enabled H-series and N-series VMs are connected in a non-blocking fat tree with a low-diameter design for optimized and consistent RDMA performance. See Enable InfiniBand to learn more about setting up InfiniBand on the InfiniBand enabled VMs.

Message passing interface

The SR-IOV enabled HB-series and N-series support almost all MPI libraries and versions. Some of the most commonly used MPI libraries are: Intel MPI, OpenMPI, HPC-X, MVAPICH2, MPICH, Platform MPI. All remote direct memory access (RDMA) verbs are supported. See Set up MPI to learn more about installing various supported MPI libraries and their optimal configuration.

Get started

The first step is to select the HB-series and N-series VM type optimal for the workload based on the VM specifications and RDMA capability. Second, configure the VM by enabling InfiniBand. There are various methods to doing this including using optimized VM images with drivers baked-in; see Optimization for Linux and Enable InfiniBand for details. Third, for distributed node workloads, choosing and configuring MPI appropriately is critical. See Set up MPI for details. Fourth, for performance and scalability, optimally configure the workloads by following guidance specific to the VM family, such as for HBv3-series overview and HC-series overview.

Next steps

- Learn about configuring and optimizing the InfiniBand enabled HB-series and N-series VMs.
- Review the HBv3-series overview and HC-series overview to learn about optimally configuring workloads for performance and scalability.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- Test your knowledge with a learning module on optimizing HPC applications on Azure.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 08/22/2024

Set up Azure HPC or AI VMs

08/22/2024

This how-to guide explains how to create a basic Azure virtual machine (VM) for HPC and AI with NVIDIA or AMD GPUs. These VM sizes are intended for workloads that require high-performance computing (HPC sizes), or GPU-accelerated computing (AI sizes).

Choose your VM size

Azure VMs have many different options, called VM sizes. There are different series of VM sizes for HPC and VM sizes GPU-optimized computing. Select the appropriate VM size for the workload you want to use. For help with selecting sizes, see the VM selector tool.

Not all Azure products are available in all Azure regions. For more information, see the current list of products available by region.

Create your VM

Before you can deploy a workload, you need to create your VM through the Azure portal.

Depending on your VM's operating system, review either the Linux VM quickstart or Windows VM quickstart. Then, create your VM with the following settings:

1. For **Subscription**, select the Azure subscription that you want to use for this VM.
2. For **Region**, select a region with capacity available for your VM size.
3. For **Image**, select the image of the VM you chose in the previous section.

   > **Note**
   >
   > For the purpose of example, this guide uses the image **NVIDIA GPU-Optimized Image for AI & HPC – v21.04.1 – Gen 1**. If you're using another image, you might need to install other software, like the NVIDIA driver and Docker, before proceeding.

4. For **Size**, select the HPC or GPU instance type. For more information, see how to choose your VM size.
5. For **SSH public key source**, select **Generate a new key pair**.
6. Wait for key validation to complete.

7. When prompted, select **Download private key and create resource**.

   > ⓘ Note
   >
   > Downloading the key pair is necessary to SSH into your VM for later configuration.

8. For **Key pair name**, enter a name for your key pair.

9. Under the **Networking** tab, make sure **Accelerated Networking** is disabled.

10. Optionally, add a data disk to your VM. For more information, see how to add a data disk to a Linux VM or to a Windows VM.

    > ⓘ Note
    >
    > Adding a data disk helps you store models, data sets, and other necessary components for benchmarking.

11. Select **Review + create** to create your VM.

Connect to your VM

Connect to your new VM using SSH, which allows you to perform further configuration. Some connection methods include:

- Connect over SSH on Linux or macOS
- Connect over SSH on Windows
- Connect over SSH using Azure Bastion

Set up VM

Set up your new VM for HPC or AI workloads. Install the newest NVIDIA or AMD GPU driver, which maps to your VM size.

- Install NVIDIA GPU drivers on N-series VMs running Linux
- Install NVIDIA GPU drivers on N-series VMs running Windows
- Install AMD GPU drivers on N-series VMs running Windows

Next steps

- High performance computing VM sizes

- GPU optimized virtual machine sizes

Configure and optimize the InfiniBand enabled H-series and N-series VMs

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article shares some guidance on configuring and optimizing the InfiniBand-enabled HB-series and N-series VMs for HPC.

VM images

On InfiniBand (IB) enabled VMs, the appropriate IB drivers are required to enable RDMA.

- The Ubuntu-HPC VM images in the Marketplace come preconfigured with the appropriate NVIDIA IB drivers and GPU drivers.
- The AlmaLinux-HPC VM images in the Marketplace come preconfigured with the appropriate NVIDIA IB drivers and GPU drivers.

These VM images are based on the base Ubuntu and AlmaLinux marketplace VM images. Scripts used in the creation of these VM images from their base marketplace images are on the azhpc-images repo.

On GPU enabled N-series VMs, the appropriate GPU drivers are additionally required. This can be available by the following methods:

- Use the Ubuntu-HPC VM images or AlmaLinux-HPC VM images that come preconfigured with the NVIDIA GPU drivers and GPU compute software stack (CUDA, NCCL).
- Add the GPU drivers through the VM extensions.
- Install the GPU drivers manually.
- Some other VM images on the Marketplace also come preinstalled with the NVIDIA GPU drivers, including some VM images from NVIDIA.

Depending on the workloads' Linux distro and version needs, Ubuntu-HPC VM images and AlmaLinux-HPC VM images on the Marketplace are the easiest way to get started with HPC and AI workloads on Azure. It's also recommended to create custom VM images with workload specific customization and configuration for reuse.

VM sizes supported by the HPC VM images

InfiniBand OFED support

The latest Azure HPC marketplace images come with Mellanox OFED 5.1 and above, which do not support ConnectX3-Pro InfiniBand cards. ConnectX-3 Pro InfiniBand cards require MOFED 4.9 LTS version. These VM images only support ConnextX-5 and newer InfiniBand cards. The following VM size support matrix for the InfiniBand OFED in these HPC VM images:

- HB-series: HB, HC, HBv2, HBv3, HBv4
- N-series: NDv2, NDv4

GPU driver support

Currently only the Ubuntu-HPC VM images and AlmaLinux-HPC VM images come preconfigured with the NVIDIA GPU drivers and GPU compute software stack (CUDA, NCCL).

The VM size support matrix for the GPU drivers in supported HPC VM images is as follows:

- N-series: NDv2, NDv4 VM sizes are supported with the NVIDIA GPU drivers and GPU compute software stack (CUDA, NCCL).
- The other 'NC' and 'ND' VM sizes in the N-series are supported with the NVIDIA GPU drivers.

All of the VM sizes in the N-series support Gen 2 VMs, though some older ones also support Gen 1 VMs. Gen 2 support is also indicated with a "01" at the end of the image URN or version.

SR-IOV enabled VMs

Ubuntu-HPC VM images

For SR-IOV enabled RDMA capable VMs, Ubuntu-HPC VM images versions 18.04, 20.04, and 22.04 are suitable. These VM images come preconfigured with the Mellanox OFED drivers for RDMA, NVIDIA GPU drivers, GPU compute software stack (CUDA, NCCL), and commonly used MPI libraries and scientific computing packages. Refer to the VM size support matrix.

- The available or latest versions of the VM images can be listed with the following information using CLI or Marketplace.

  Output

  ```
  "publisher": "Microsoft-DSVM",
  "offer": "Ubuntu-HPC",
  ```

- Scripts used in the creation of the Ubuntu-HPC VM images from a base Ubuntu Marketplace image are on the azhpc-images repo.

AlmaLinux-HPC VM images

For SR-IOV enabled RDMA capable VMs, AlmaLinux-HPC VM images versions 8.5, 8.6, and 8.7 are suitable. These VM images come preconfigured with the Mellanox OFED drivers for RDMA, NVIDIA GPU drivers, GPU compute software stack (CUDA, NCCL), and commonly used MPI libraries and scientific computing packages. Refer to the VM size support matrix.

- The available or latest versions of the VM images can be listed with the following information using CLI or Marketplace.

  Output

  ```
  "publisher": "AlmaLinux",
  "offer": "AlmaLinux-HPC",
  ```

- Scripts used in the creation of the AlmaLinux-HPC VM images from a base AlmaLinux Marketplace image are on the azhpc-images repo.

Additionally, more details on what's included in the Ubuntu-HPC VM images and AlmaLinux-HPC VM images, and how to deploy them are in Azure HPC VM images.

RHEL VM images

The base RHEL-based non-HPC VM images on the Marketplace can be configured for use on the SR-IOV enabled RDMA capable VMs. Learn more about enabling InfiniBand and setting up MPI on the VMs.

Ubuntu VM images

The base Ubuntu Server 20.04 LTS and 22.04 LTS VM images in the Marketplace are supported for both SR-IOV and non-SR-IOV RDMA capable VMs. Learn more about enabling InfiniBand and setting up MPI on the VMs.

- Instructions for enabling InfiniBand on the Ubuntu VM images are in a TechCommunity article.

> **Note**
>
> Mellanox OFED 5.1 and above don't support ConnectX3-Pro InfiniBand cards on SR-IOV enabled N-series VM sizes with FDR InfiniBand (e.g. NCv3). Please use LTS Mellanox OFED version 4.9-0.1.7.0 or older on the N-series VM's with ConnectX3-Pro cards. For more information, see **Linux InfiniBand Drivers**.

SUSE Linux Enterprise Server VM images

SLES 12 SP3 for HPC, SLES 12 SP3 for HPC (Premium), SLES 12 SP1 for HPC, SLES 12 SP1 for HPC (Premium), SLES 12 SP4 and SLES 15 VM images in the Marketplace are supported. These VM images come preloaded with the Network Direct drivers for RDMA (on the non-SR-IOV VM sizes) and Intel MPI version 5.1. Learn more about setting up MPI on the VMs.

Optimize VMs

The following are some optional optimization settings for improved performance on the VM.

Update LIS

If necessary for functionality or performance, Linux Integration Services (LIS) drivers can be installed or updated on supported OS distros, especially is deploying using a custom image or an older OS version such as RHEL 6.x or earlier version of 7.x.

Bash

```
wget https://aka.ms/lis
tar xzf lis
pushd LISISO
sudo ./upgrade.sh
```

Reclaim memory

Improve performance by automatically reclaiming memory to avoid remote memory access.

Bash

```
sudo echo 1 >/proc/sys/vm/zone_reclaim_mode
```

Keep reclaim memory mode persistent after VM reboots:

Bash

```
sudo echo "vm.zone_reclaim_mode = 1" >> /etc/sysctl.conf sysctl -p
```

Disable firewall and SELinux

Bash

```
sudo systemctl stop iptables.service
sudo systemctl disable iptables.service
sudo systemctl mask firewalld
sudo systemctl stop firewalld.service
sudo systemctl disable firewalld.service
sudo iptables -nL
sudo sed -i -e's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

Disable cpupower

Bash

```
sudo service cpupower status
```

If enabled, disable it:

Bash

```
sudo service cpupower stop
sudo systemctl disable cpupower
```

Configure WALinuxAgent

Bash

```
sudo sed -i -e 's/# OS.EnableRDMA=y/OS.EnableRDMA=y/g' /etc/waagent.conf
```

Optionally, the WALinuxAgent may be disabled before running a job then enabled post-job for maximum VM resource availability to the HPC workload.

Next steps

- Learn more about enabling InfiniBand on the InfiniBand-enabled HB-series and N-series VMs.
- Learn more about installing and running various supported MPI libraries on the VMs.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 08/22/2024

Enable InfiniBand

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

RDMA capable HB-series and N-series VMs communicate over the low latency and high bandwidth InfiniBand network. The RDMA capability over such an interconnect is critical to boost the scalability and performance of distributed-node HPC and AI workloads. The InfiniBand enabled HB-series and N-series VMs are connected in a non-blocking fat tree with a low-diameter design for optimized and consistent RDMA performance.

There are various ways to enable InfiniBand on the capable VM sizes.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

VM Images with InfiniBand drivers

See VM Images for a list of supported VM Images on the Marketplace, which come pre-loaded with InfiniBand drivers (for SR-IOV or non-SR-IOV VMs) or can be configured with the appropriate drivers for RDMA capable VMs. The Ubuntu-HPC and AlmaLinux-HPC VM images in the marketplace are the easiest way to get started.

InfiniBand Driver VM Extensions

On Linux, the InfiniBandDriverLinux VM extension can be used to install the Mellanox OFED drivers and enable InfiniBand on the SR-IOV enabled HB-series and N-series VMs.

On Windows, the InfiniBandDriverWindows VM extension installs Windows Network Direct drivers (on non-SR-IOV VMs) or Mellanox OFED drivers (on SR-IOV VMs) for RDMA connectivity. In certain deployments of A8 and A9 instances, the HpcVmDrivers extension is added automatically. Note that the HpcVmDrivers VM extension is being deprecated; it will not be updated.

To add the VM extension to a VM, you can use Azure PowerShell cmdlets. For more information, see Virtual machine extensions and features. You can also work with extensions for

VMs deployed in the classic deployment model.

Manual installation

Mellanox OpenFabrics drivers (OFED) can be manually installed on the SR-IOV enabled HB-series and N-series VMs.

Linux

The OFED drivers for Linux can be installed with the example below. Though the example here is for RHEL, but the steps are general and can be used for any compatible Linux operating system such as Ubuntu (22.04, 24.04) and SLES (12 SP4+ and 15). More examples for other distros are on the azhpc-images repo. The inbox drivers also work as well, but the Mellanox OFED drivers provide more features.

Bash

```
MLNX_OFED_DOWNLOAD_URL=http://content.mellanox.com/ofed/MLNX_OFED-5.0-
2.1.8.0/MLNX_OFED_LINUX-5.0-2.1.8.0-rhel7.7-x86_64.tgz
# Optionally verify checksum
wget --retry-connrefused --tries=3 --waitretry=5 $MLNX_OFED_DOWNLOAD_URL
tar zxvf MLNX_OFED_LINUX-5.0-2.1.8.0-rhel7.7-x86_64.tgz

KERNEL=( $(rpm -q kernel | sed 's/kernel\-//g') )
KERNEL=${KERNEL[-1]}
# Uncomment the lines below if you are running this on a VM
#RELEASE=( $(cat /etc/redhat-release | awk '{print $4}') )
#yum -y install
http://olcentgbl.trafficmanager.net/redhat/${RELEASE}/updates/x86_64/kernel-
devel-${KERNEL}.rpm
sudo yum install -y kernel-devel-${KERNEL}
sudo ./MLNX_OFED_LINUX-5.0-2.1.8.0-rhel7.7-x86_64/mlnxofedinstall --kernel $KERNEL -
-kernel-sources /usr/src/kernels/${KERNEL} --add-kernel-support --skip-repo
```

Windows

For Windows, download and install the Mellanox OFED for Windows drivers.

Enable IP over InfiniBand (IB)

If you plan to run MPI jobs, you typically don't need IPoIB. The MPI library will use the verbs interface for IB communication (unless you explicitly use the TCP/IP channel of MPI library). But if you have an app that uses TCP/IP for communication and you want to run over IB, you can

use IPoIB over the IB interface. Use the following commands (for RHEL) to enable IP over InfiniBand.

> **Important**
>
> To avoid issues, ensure you aren't running older versions of Microsoft Azure Linux Agent (waagent). We recommend using at least version 2.4.0.2 before enabling IP over IB.

Bash

```
sudo sed -i -e 's/# OS.EnableRDMA=n/OS.EnableRDMA=y/g' /etc/waagent.conf
sudo systemctl restart waagent
```

Next steps

- Learn more about installing and running various supported MPI libraries on the VMs.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 12/03/2025

Set up Message Passing Interface for HPC

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Message Passing Interface (MPI) is an open library and defacto standard for distributed memory parallelization. It's commonly used across many HPC workloads. HPC workloads on the RDMA capable HB-series and N-series VMs can use MPI to communicate over the low latency and high bandwidth InfiniBand network.

- The SR-IOV enabled VM sizes on Azure allow almost any flavor of MPI to be used with Mellanox OFED.
- On non-SR-IOV enabled VMs, supported MPI implementations use the Microsoft Network Direct (ND) interface to communicate between VMs. Hence, only Microsoft MPI (MS-MPI) 2012 R2 or later and Intel MPI 5.x versions are supported. Later versions (2017, 2018) of the Intel MPI runtime library may or may not be compatible with the Azure RDMA drivers.

For SR-IOV enabled RDMA capable VMs, Ubuntu-HPC VM images and AlmaLinux-HPC VM images are suitable. These VM images come optimized and preloaded with the OFED drivers for RDMA and various commonly used MPI libraries and scientific computing packages and are the easiest way to get started.

Though the examples here are for RHEL, but the steps are general and can be used for any compatible Linux operating system such as Ubuntu (18.04, 20.04, 22.04) and SLES (12 SP4 and 15 SP4). More examples for setting up other MPI implementations on others distros is on the azhpc-images repo.

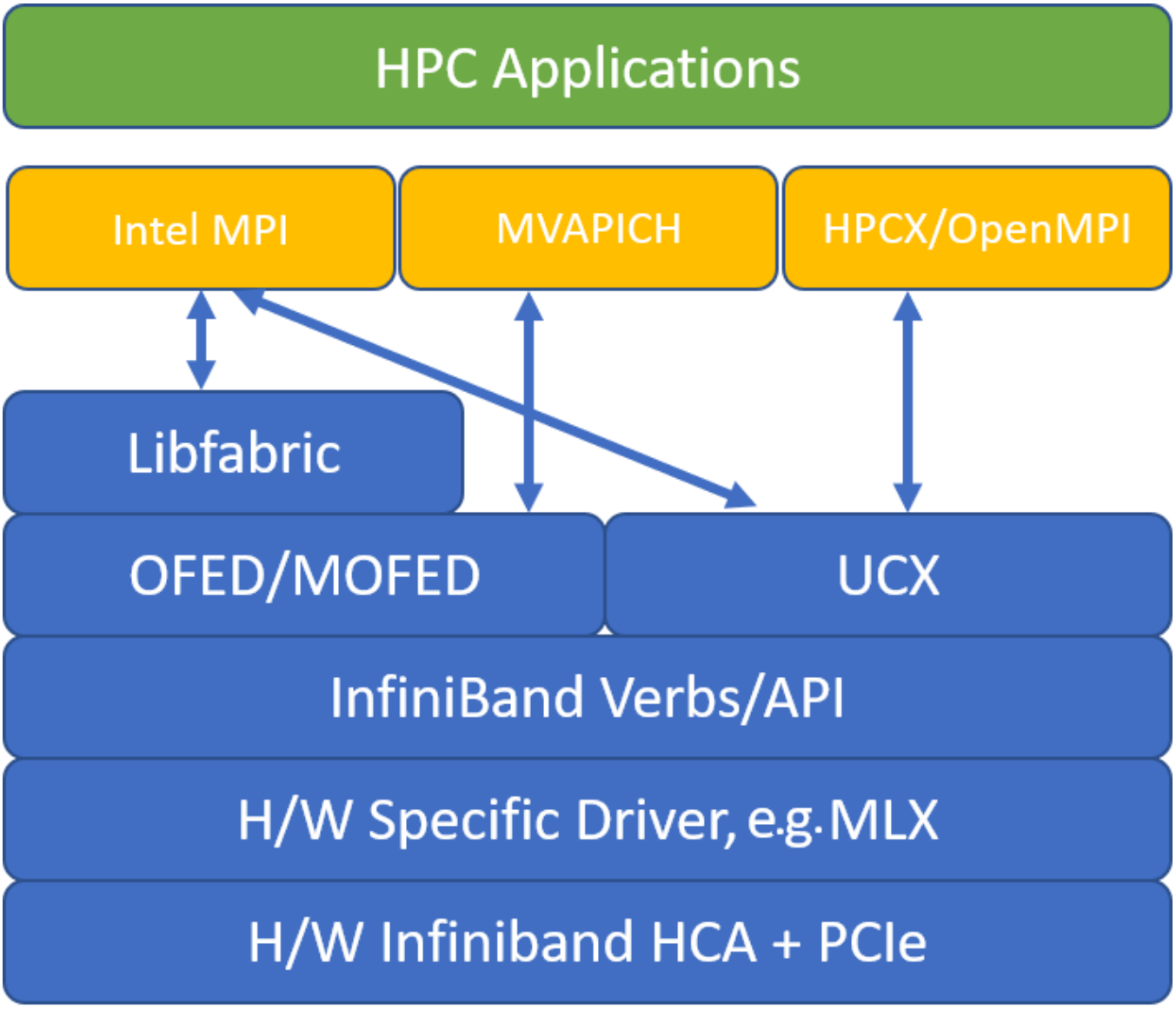
> **Note**
>
> Running MPI jobs on SR-IOV enabled VMs with certain MPI libraries (such as Platform MPI) may require setting up of partition keys (p-keys) across a tenant for isolation and security. Follow the steps in the **Discover partition keys** section for details on determining the p-key values and setting them correctly for an MPI job with that MPI library.

> **Note**
>
> The code snippets below are examples. We recommend using the latest stable versions of the packages, or referring to the **azhpc-images repo**.

Choosing MPI library

If an HPC application recommends a particular MPI library, try that version first. If you have flexibility regarding which MPI you can choose, and you want the best performance, try HPC-X. Overall, the HPC-X MPI performs the best by using the UCX framework for the InfiniBand interface, and takes advantage of all the Mellanox InfiniBand hardware and software capabilities. Additionally, HPC-X and OpenMPI are ABI compatible, so you can dynamically run an HPC application with HPC-X that was built with OpenMPI. Similarly, Intel MPI, MVAPICH, and MPICH are ABI compatible.

The following figure illustrates the architecture for the popular MPI libraries.



HPC-X

The HPC-X software toolkit contains UCX and HCOLL and can be built against UCX.

Bash

```
HPCX_VERSION="v2.6.0"
HPCX_DOWNLOAD_URL=https://azhpcstor.blob.core.windows.net/azhpc-images-store/hpcx-
```

```
v2.6.0-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-redhat7.7-x86_64.tbz
wget --retry-connrefused --tries=3 --waitretry=5 $HPCX_DOWNLOAD_URL
tar -xvf hpcx-${HPCX_VERSION}-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-redhat7.7-x86_64.tbz
mv hpcx-${HPCX_VERSION}-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-redhat7.7-x86_64
${INSTALL_PREFIX}
HPCX_PATH=${INSTALL_PREFIX}/hpcx-${HPCX_VERSION}-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-
redhat7.7-x86_64
```

The following command illustrates some recommended mpirun arguments for HPC-X and OpenMPI.

Bash

```
mpirun -n $NPROCS --hostfile $HOSTFILE --map-by
ppr:$NUMBER_PROCESSES_PER_NUMA:numa:pe=$NUMBER_THREADS_PER_PROCESS -report-bindings
$MPI_EXECUTABLE
```

where:

Expand table

Parameter	Description
`NPROCS`	Specifies the number of MPI processes. For example: `-n 16`.
`$HOSTFILE`	Specifies a file containing the hostname or IP address, to indicate the location of where the MPI processes run. For example: `--hostfile hosts`.
`$NUMBER_PROCESSES_PER_NUMA`	Specifies the number of MPI processes that run in each NUMA domain. For example, to specify four MPI processes per NUMA, you use `--map-by ppr:4:numa:pe=1`.
`$NUMBER_THREADS_PER_PROCESS`	Specifies the number of threads per MPI process. For example, to specify one MPI process and four threads per NUMA, you use `--map-by ppr:1:numa:pe=4`.
`-report-bindings`	Prints MPI processes mapping to cores, which is useful to verify that your MPI process pinning is correct.
`$MPI_EXECUTABLE`	Specifies the MPI executable built linking in MPI libraries. MPI compiler wrappers do this automatically. For example: `mpicc` or `mpif90`.

An example of running the OSU latency microbenchmark is as follows:

Bash

```
${HPCX_PATH}mpirun -np 2 --map-by ppr:2:node -x UCX_TLS=rc
${HPCX_PATH}/ompi/tests/osu-micro-benchmarks-5.3.2/osu_latency
```

Optimizing MPI collectives

MPI Collective communication primitives offer a flexible, portable way to implement group communication operations. They're widely used across various scientific parallel applications and have a significant impact on the overall application performance. Refer to the TechCommunity article for details on configuration parameters to optimize collective communication performance using HPC-X and HCOLL library for collective communication.

As an example, if you suspect your tightly coupled MPI application is doing an excessive amount of collective communication, you can try enabling hierarchical collectives (HCOLL). To enable those features, use the following parameters.

Bash

```
-mca coll_hcoll_enable 1 -x HCOLL_MAIN_IB=<MLX device>:<Port>
```

> **Note**
>
> With HPC-X 2.7.4+, it may be necessary to explicitly pass LD_LIBRARY_PATH if the UCX version on MOFED vs. that in HPC-X is different.

OpenMPI

Install UCX as described above. HCOLL is part of the HPC-X software toolkit and doesn't requires special installation.

OpenMPI can be installed from the packages available in the repo.

Bash

```
sudo yum install -y openmpi
```

We recommend building a latest, stable release of OpenMPI with UCX.

Bash

```
OMPI_VERSION="4.0.3"
OMPI_DOWNLOAD_URL=https://download.open-mpi.org/release/open-
mpi/v4.0/openmpi-${OMPI_VERSION}.tar.gz
wget --retry-connrefused --tries=3 --waitretry=5 $OMPI_DOWNLOAD_URL
tar -xvf openmpi-${OMPI_VERSION}.tar.gz
cd openmpi-${OMPI_VERSION}
./configure --prefix=${INSTALL_PREFIX}/openmpi-${OMPI_VERSION} --with-
```

```
ucx=${UCX_PATH} --with-hcoll=${HCOLL_PATH} --enable-mpirun-prefix-by-default --with-platform=contrib/platform/mellanox/optimized && make -j$(nproc) && make install
```

For optimal performance, run OpenMPI with `ucx` and `hcoll`. Also see the example with HPC-X.

Bash

```
${INSTALL_PREFIX}/bin/mpirun -np 2 --map-by node --hostfile ~/hostfile -mca pml ucx --mca btl ^vader,tcp,openib -x UCX_NET_DEVICES=mlx5_0:1  -x UCX_IB_PKEY=0x0003 ./osu_latency
```

Check your partition key as mentioned above.

Intel MPI

Download your choice of version of Intel MPI. The Intel MPI 2019 release switched from the Open Fabrics Alliance (OFA) framework to the Open Fabrics Interfaces (OFI) framework, and currently supports libfabric. There are two providers for InfiniBand support: mlx and verbs. Change the I_MPI_FABRICS environment variable depending on the version.

- Intel MPI 2019 and 2021: use `I_MPI_FABRICS=shm:ofi`, `I_MPI_OFI_PROVIDER=mlx`. The `mlx` provider uses UCX. Usage of verbs has been found to be unstable and less performant. See the TechCommunity article for more details.
- Intel MPI 2018: use `I_MPI_FABRICS=shm:ofa`
- Intel MPI 2016: use `I_MPI_DAPL_PROVIDER=ofa-v2-ib0`

Here are some suggested mpirun arguments for Intel MPI 2019 update 5+.

Bash

```
export FI_PROVIDER=mlx
export I_MPI_DEBUG=5
export I_MPI_PIN_DOMAIN=numa

mpirun -n $NPROCS -f $HOSTFILE $MPI_EXECUTABLE
```

where:

Expand table

Parameter	Description
`FI_PROVIDER`	Specifies which libfabric provider to use, which will affect the API, protocol, and network used. verbs is another option, but generally mlx gives you better performance.

Parameter	Description
`I_MPI_DEBUG`	Specifies the level of extra debug output, which can provide details about where processes are pinned, and which protocol and network are used.
`I_MPI_PIN_DOMAIN`	Specifies how you want to pin your processes. For example, you can pin to cores, sockets, or NUMA domains. In this example, you set this environmental variable to numa, which means processes will be pinned to NUMA node domains.

Optimizing MPI collectives

There are some other options that you can try, especially if collective operations are consuming a significant amount of time. Intel MPI 2019 update 5+ supports the provide mlx and uses the UCX framework to communicate with InfiniBand. It also supports HCOLL.

Bash

```
export FI_PROVIDER=mlx
export I_MPI_COLL_EXTERNAL=1
```

Non SR-IOV VMs

For non SR-IOV VMs, an example of downloading the 5.x runtime free evaluation version is as follows:

Bash

```
wget http://registrationcenter-
download.intel.com/akdlm/irc_nas/tec/9278/l_mpi_p_5.1.3.223.tgz
```

For installation steps, see the Intel MPI Library Installation Guide. Optionally, you may want to enable ptrace for non-root non-debugger processes (needed for the most recent versions of Intel MPI).

Bash

```
echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope
```

SUSE Linux

For SUSE Linux Enterprise Server VM image versions - SLES 12 SP3 for HPC, SLES 12 SP3 for HPC (Premium), SLES 12 SP1 for HPC, SLES 12 SP1 for HPC (Premium), SLES 12 SP4 and SLES

15, the RDMA drivers are installed and Intel MPI packages are distributed on the VM. Install Intel MPI by running the following command:

```bash
sudo rpm -v -i --nodeps /opt/intelMPI/intel_mpi_packages/*.rpm
```

MVAPICH

The following is an example of building MVAPICH2. Note newer versions may be available than what is used below.

```bash
wget http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.3.tar.gz
tar -xv mvapich2-2.3.tar.gz
cd mvapich2-2.3
./configure --prefix=${INSTALL_PREFIX}
make -j 8 && make install
```

An example of running the OSU latency microbenchmark is as follows:

```bash
${INSTALL_PREFIX}/bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=48
./osu_latency
```

The following list contains several recommended `mpirun` arguments.

```bash
export MV2_CPU_BINDING_POLICY=scatter
export MV2_CPU_BINDING_LEVEL=numanode
export MV2_SHOW_CPU_BINDING=1
export MV2_SHOW_HCA_BINDING=1

mpirun -n $NPROCS -f $HOSTFILE $MPI_EXECUTABLE
```

where:

Expand table

Parameter	Description
`MV2_CPU_BINDING_POLICY`	Specifies which binding policy to use, which will affect how processes are pinned to core IDs. In this case, you specify `scatter`, so processes are evenly

Parameter	Description
	scattered among the NUMA domains.
`MV2_CPU_BINDING_LEVEL`	Specifies where to pin processes. In this case, you set it to numanode, which means processes are pinned to units of NUMA domains.
`MV2_SHOW_CPU_BINDING`	Specifies if you want to get debug information about where the processes are pinned.
`MV2_SHOW_HCA_BINDING`	Specifies if you want to get debug information about which host channel adapter each process is using.

Platform MPI

Install required packages for Platform MPI Community Edition.

Bash

```
sudo yum install libstdc++.i686
sudo yum install glibc.i686
Download platform MPI at
https://www.ibm.com/developerworks/downloads/im/mpi/index.html
sudo ./platform_mpi-09.01.04.03r-ce.bin
```

Follow the installation process.

MPICH

Install UCX as described above. Build MPICH.

Bash

```
wget https://www.mpich.org/static/downloads/3.3/mpich-3.3.tar.gz
tar -xvf mpich-3.3.tar.gz
cd mpich-3.3
./configure --with-ucx=${UCX_PATH} --prefix=${INSTALL_PREFIX} --with-device=ch4:ucx
make -j 8 && make install
```

Running MPICH.

Bash

```
${INSTALL_PREFIX}/bin/mpiexec -n 2 -hostfile ~/hostfile -env UCX_IB_PKEY=0x0003 -
bind-to hwthread ./osu_latency
```

Check your partition key as mentioned above.

OSU MPI Benchmarks

Download OSU MPI Benchmarks and untar.

```Bash
wget http://mvapich.cse.ohio-state.edu/download/mvapich/osu-micro-benchmarks-5.5.tar.gz
tar -xvf osu-micro-benchmarks-5.5.tar.gz
cd osu-micro-benchmarks-5.5
```

Build Benchmarks using a particular MPI library:

```Bash
CC=<mpi-install-path/bin/mpicc>CXX=<mpi-install-path/bin/mpicxx> ./configure
make
```

MPI Benchmarks are under `mpi/` folder.

Discover partition keys

Discover partition keys (p-keys) for communicating with other VMs within the same tenant (Availability Set or Virtual Machine Scale Set).

```Bash
/sys/class/infiniband/mlx5_0/ports/1/pkeys/0
/sys/class/infiniband/mlx5_0/ports/1/pkeys/1
```

The larger of the two is the tenant key that should be used with MPI. Example: If the following are the p-keys, 0x800b should be used with MPI.

```Bash
cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/0
0x800b
cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/1
0x7fff
```

Note interfaces are named as `mlx5_ib*` inside HPC VM images.

Also note that as long as the tenant (Availability Set or Virtual Machine Scale Set) exists, the PKEYs remain the same. This is true even when nodes are added/deleted. New tenants get different PKEYs.

Set up user limits for MPI

Set up user limits for MPI.

Bash

```
cat << EOF | sudo tee -a /etc/security/limits.conf
*               hard    memlock         unlimited
*               soft    memlock         unlimited
*               hard    nofile          65535
*               soft    nofile          65535
EOF
```

Set up SSH keys for MPI

Set up SSH keys for MPI types that require it.

Bash

```
ssh-keygen -f /home/$USER/.ssh/id_rsa -t rsa -N ''
cat << EOF > /home/$USER/.ssh/config
Host *
    StrictHostKeyChecking no
EOF
cat /home/$USER/.ssh/id_rsa.pub >> /home/$USER/.ssh/authorized_keys
chmod 600 /home/$USER/.ssh/authorized_keys
chmod 644 /home/$USER/.ssh/config
```

The above syntax assumes a shared home directory, else `.ssh` directory must be copied to each node.

Next steps

- Learn about the InfiniBand enabled HB-series and N-series VMs.
- Review the HBv3-series overview and HC-series overview.
- Read Optimal MPI process placement for HB-series VMs.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 09/17/2024

Scaling HPC applications

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Optimal scale-up and scale-out performance of HPC applications on Azure requires performance tuning and optimization experiments for the specific workload. This section and the VM series-specific pages offer general guidance for scaling your applications.

Application setup

The azurehpc repo contains many examples of:

- Setting up and running applications optimally.
- Configuration of file systems, and clusters.
- Tutorials on how to get started easily with some common application workflows.

Optimally scaling MPI

The following suggestions apply for optimal application scaling efficiency, performance, and consistency:

- For smaller scale jobs (< 256K connections) use: `UCX_TLS=rc,sm`
- For larger scale jobs (> 256K connections) use: `UCX_TLS=dc,sm`
- To calculate the number of connections for your MPI job, use: `Max Connections = (processes per node) x (number of nodes per job) x (number of nodes per job)`

Adaptive Routing

Adaptive Routing (AR) allows Azure Virtual Machines (VMs) running EDR and HDR InfiniBand to automatically detect and avoid network congestion by dynamically selecting optimal network paths. As a result, AR offers improved latency and bandwidth on the InfiniBand network, which in turn drives higher performance and scaling efficiency. For more information, see TechCommunity article.

Process pinning

- Pin processes to cores using a sequential pinning approach (as opposed to an autobalance approach).
- Binding by Numa/Core/HwThread is better than default binding.

- For hybrid parallel applications (OpenMP+MPI), use four threads and one MPI rank per [CCX](HB-series virtual machines overview including info on CCXs) on HB and HBv2 VM sizes.
- For pure MPI applications, experiment with between one to four MPI ranks per CCX for optimal performance on HB and HBv2 VM sizes.
- Some applications with extreme sensitivity to memory bandwidth may benefit from using a reduced number of cores per CCX. For these applications, using three or two cores per CCX may reduce memory bandwidth contention and yield higher real-world performance or more consistent scalability. In particular, MPI 'Allreduce' may benefit from this approach.
- For larger scale runs, it's recommended to use UD or hybrid RC+UD transports. Many MPI libraries/runtime libraries use these transports internally (such as UCX or MVAPICH2). Check your transport configurations for large-scale runs.

Compiling applications

▼ Click to expand

Though not necessary, compiling applications with appropriate optimization flags provides the best scale-up performance on HB and HC-series VMs.

AMD Optimizing C/C++ Compiler

The AMD Optimizing C/C++ Compiler (AOCC) compiler system offers a high level of advanced optimizations, multi-threading, and processor support that includes global optimization, vectorization, inter-procedural analyses, loop transformations, and code generation. AOCC compiler binaries are suitable for Linux systems having GNU C Library (glibc) version 2.17 and above. The compiler suite consists of a C/C++ compiler (clang), a Fortran compiler (FLANG), and a Fortran front end to Clang (Dragon Egg).

Clang

Clang is a C, C++, and Objective-C compiler handling preprocessing, parsing, optimization, code generation, assembly, and linking. Clang supports the `-march=znver1` flag to enable best code generation and tuning for AMD's Zen based x86 architecture.

FLANG

The FLANG compiler is a recent addition to the AOCC suite (added April 2018) and is currently in prerelease for developers to download and test. Based on Fortran 2008, AMD extends the GitHub version of FLANG. The FLANG compiler supports all Clang compiler options and other number of FLANG-specific compiler options.

DragonEgg

DragonEgg is a gcc plugin that replaces GCC's optimizers and code generators from the LLVM project. DragonEgg that comes with AOCC works with gcc-4.8.x, has been tested for x86-32/x86-64 targets, and has been successfully used on various Linux platforms.

GFortran is the actual frontend for Fortran programs responsible for preprocessing, parsing, and semantic analysis generating the GCC GIMPLE intermediate representation (IR). DragonEgg is a GNU plugin, plugging into GFortran compilation flow. It implements the GNU plugin API. With the plugin architecture, DragonEgg becomes the compiler driver, driving the different phases of compilation. After following the download and installation instructions, Dragon Egg can be invoked using:

Bash

```
gfortran [gFortran flags]
   -fplugin=/path/AOCC-1.2-Compiler/AOCC-1.2-
   FortranPlugin/dragonegg.so [plugin optimization flags]
   -c xyz.f90 $ clang -O3 -lgfortran -o xyz xyz.o $./xyz
```

PGI Compiler

PGI Community Edition 17 is confirmed to work with AMD EPYC. A PGI-compiled version of STREAM does deliver full memory bandwidth of the platform. The newer Community Edition 18.10 (Nov 2018) should likewise work well. Use this CLI command to compile with the Intel Compiler:

Bash

```
pgcc $(OPTIMIZATIONS_PGI) $(STACK) -DSTREAM_ARRAY_SIZE=800000000 stream.c -o
stream.pgi
```

Intel Compiler

Intel Compiler 18 is confirmed to work with AMD EPYC. Use this CLI command to compile with the Intel Compiler.

Bash

```
icc -o stream.intel stream.c -DSTATIC -DSTREAM_ARRAY_SIZE=800000000 -mcmodel=large -
shared-intel -Ofast –qopenmp
```

GCC Compiler

For HPC workloads, AMD recommends GCC compiler 7.3 or newer. Older versions, such as 4.8.5 included with RHEL 7.4, aren't recommended. GCC 7.3, and newer, delivers higher performance on HPL, HPCG, and DGEMM tests.

Bash

```
gcc $(OPTIMIZATIONS) $(OMP) $(STACK) $(STREAM_PARAMETERS) stream.c -o stream.gcc
```

Next steps

- Test your knowledge with a learning module on optimizing HPC applications on Azure.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- Learn more about HPC on Azure.

Last updated on 08/22/2024

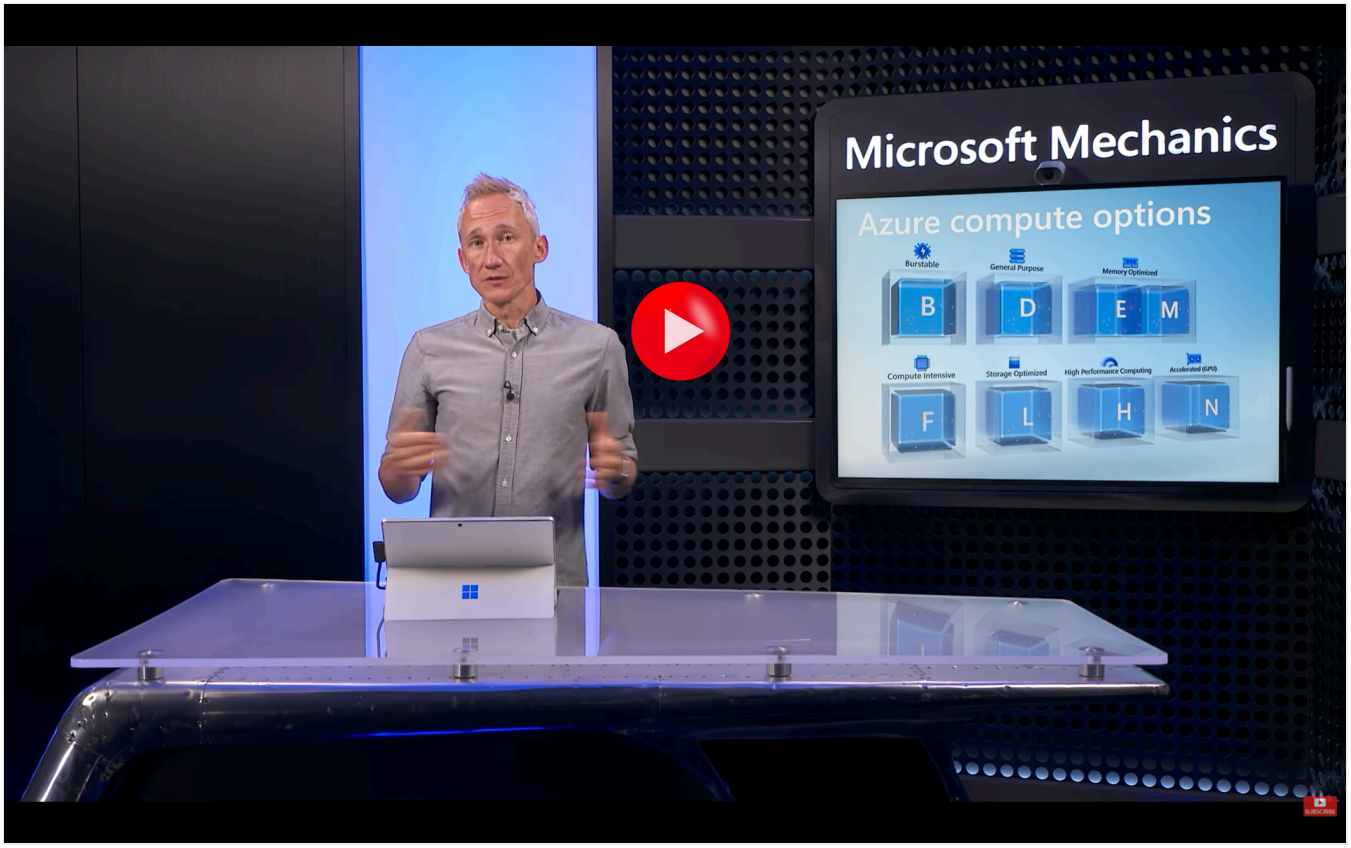
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

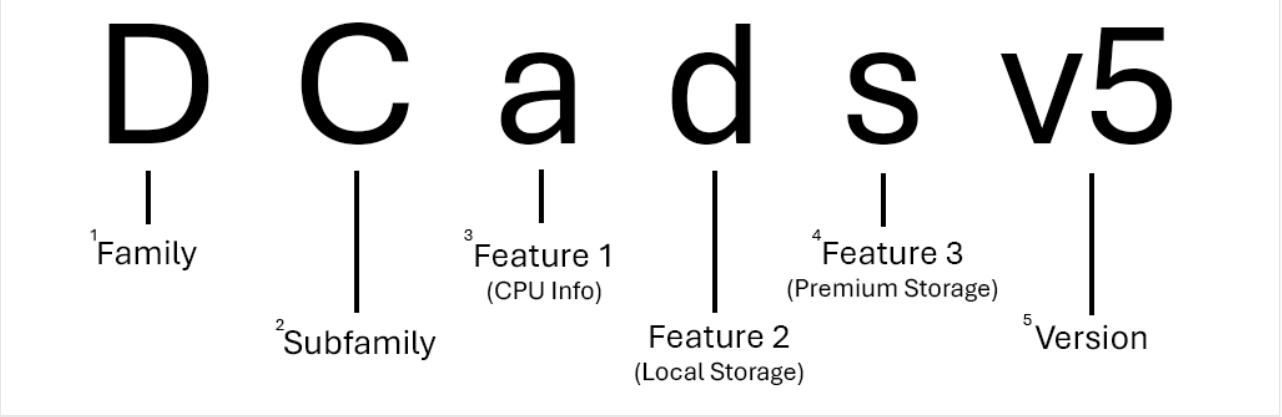
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

Azure virtual machine sizes naming conventions

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This page outlines the naming conventions used for Azure VMs. VMs use these naming conventions to denote varying features and specifications.

Naming convention explanation

[Family] + *[Sub-family]** + **[# of vCPUs]** + *[Constrained vCPUs]** + **[Additive Features]** + *[Accelerator Type]** + *[Memory Capacity]** + **[Version]**

Expand table

Value	Explanation
Family	Indicates the VM Family Series. For more information, see the list of VM size families by type.
*Subfamily	Used for specialized VM differentiations such as: B = memory bandwidth optimized C = confidential (for DC, EC, NCC series) C = compute intensive (for HC, NC, NCC series) D = AI training and inference optimized G = cloud gaming and remote desktop optimized V = visualization and graphics optimized X = extra memory
# of vCPUs	Denotes the number of vCPUs of the VM
*Constrained vCPUs	Used for certain VM sizes only. Denotes the number of vCPUs for the constrained vCPU capable size
Additive Features	Lower case letters denote additive features, such as: a = AMD-based processor b = remote storage bandwidth optimized d = includes local temporary disks e = encrypted; contains confidential TDX capabilities f = flat ratio (1:1) of vCPU to memory size i = isolated size l = low memory; decreased vCPU to memory ratio m = memory intensive; highest vCPU to memory ratio in a particular series n = network optimized; increased vCPU to network bandwidth ratio o = increased vCPU to local SSD storage capacity ratio p = ARM-based processor r = includes RDMA (InfiniBand) secondary network

Value	Explanation
	s = compatible with any premium SSD type t = tiny memory; smallest vCPU to memory ratio in a particular size
*Accelerator Type	Denotes the type of hardware accelerator in the specialized/GPU SKUs. Only new specialized/GPU SKUs launched from Q3 2020 have the hardware accelerator in the name.
*Memory Capacity	Denotes M-series memory capacity, rounded to the nearest TiB.
Version	Denotes the version number of the VM Family Series

Example breakdown

[Family] + *[Sub-family]** + **[# of vCPUs]** + *[Constrained vCPUs]** + **[Additive Features]** + *[Accelerator Type]** + *[Memory Capacity]** + **[Version]**

Example 1: M48ds_1_v3

Expand table

Value	Explanation
Family	M
# of vCPUs	48
Additive Features	d = local temp disk s = Premium Storage capable
Memory Capacity	974 GiB ≈ 1 TiB
Version	v3

Example 2: NV16as_v4

Expand table

Value	Explanation
Family	N
Subfamily	V

Value	Explanation
# of vCPUs	16
Additive Features	a = AMD-based processor s = Premium Storage capable
Version	v4

Example 3: NC4as_T4_v3

Expand table

Value	Explanation
Family	N
Subfamily	C
# of vCPUs	4
Additive Features	a = AMD-based processor s = Premium Storage capable
Accelerator Type	T4
Version	v3

Example 4: M8-2ms_v2 (Constrained vCPU)

Expand table

Value	Explanation
Family	M
# of vCPUs	8
Constrained vCPUs	2
Additive Features	m = memory intensive s = Premium Storage capable
Version	v2

Example 5: E96bds_v5 (a SQL Server preferred option)

Expand table

Value	Explanation
Family	Eb
# of vCPUs	96
Additive Features	b = memory d = includes local temporary disks s = Premium Storage capable
Version	v5

Next steps

Learn more about available VM Sizes in Azure.

Last updated on 12/01/2025

Change the size of a virtual machine

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This article shows you how to change an existing virtual machine's VM size.

After you create a VM, you can scale it up or down by changing the VM size. This process works whether the VM is currently running or has already been deallocated.

In some cases, you must deallocate the VM before resizing. Deallocation may be required if the new size isn't available on the hardware cluster currently hosting the VM. Even when deallocation isn't required, changing the size of a running VM will cause it to restart. For this reason, resizing a VM should be considered a disruptive operation, especially for stateful workloads.

> ⚠ **Warning**
>
> If the virtual machine is currently running, changing its size will cause it to restart.
>
> Deallocating the VM also releases any dynamic IP addresses assigned to the VM. The OS and data disks are not affected.
>
> If a resize operation fails, the VM model will still display the requested size, but the VM will continue running on its previous size until the resize is successfully allocated. This behavior applies to both the Azure portal, and all GET operations, as the displayed size always reflects the latest applied configuration, not the actual running size.

Choose the right SKU

When resizing a VM, it's important to choose the right SKU based on the signals from the VM to determine whether you need more CPU, memory, or storage capacity:

- If the VM is running a CPU-intensive workload, such as a database server or a web server with high traffic, you may need to choose a SKU with more CPU cores. For more information, see Compute optimized sizes.
- If the VM is running a memory-intensive workload, such as a machine learning model or a big data application, you may need to choose a SKU with more memory. For more information, see Memory optimized sizes.
- If the VM is running out of storage capacity, you may need to choose a SKU with more storage. For more information, see Storage optimized sizes.

- If your VM uses Premium Storage, make sure that you choose an **s** version of the size to get Premium Storage support. For example, choose Standard_E4**s**_v3 instead of Standard_E4_v3.

For more information on choosing the right SKU, you can use the following resources:

- Sizes for VMs in Azure: This article lists all the VM sizes available in Azure.
- Azure VM Selector : This tool helps you find the right VM SKU based on your workload type, OS and software, and deployment region.

Limitations

1. You can't resize a VM size that has a local temp disk to a VM size with no local temp disk and vice versa.

   The only combinations allowed for resizing are:

- VM (with local temp disk) -> VM (with local temp disk); and
- VM (with no local temp disk) -> VM (with no local temp disk).

  For a work-around, see How do I migrate from a VM size with local temp disk to a VM size with no local temp disk? . The work-around can be used to resize a VM with no local temp disk to VM with a local temp disk. You create a snapshot of the VM with no local temp disk > create a disk from the snapshot > create VM from the disk with appropriate VM size that supports VMs with a local temp disk.

2. You can't resize a VM size that has a SCSI-based VM to a VM size that has a remote NVMe-enabled VM.

   For a workaround, see How can I resize a SCSI-based VM to a remote NVMe-enabled VM of a different size?

Change the VM size

Select an option for changing the VM size:

Portal

To change the VM size using the Azure portal:

1. Open the Azure portal .

2. Type *virtual machines* in the search. Under **Services**, select **Virtual machines**.
3. In the **Virtual machines** page, select the virtual machine you want to resize.
4. In the left menu, in the **Availability + scale** section, select **size**.
5. Pick a new compatible size from the list of available sizes and then select **Resize**.

If your VM is still running and you don't see the size you want in the list, stopping the virtual machine may reveal more sizes.

Next steps

- For more scalability, run multiple VM instances and scale out.
- For more SKU selection information, see Sizes for virtual machines in Azure.
- To determine VM sizes by workload type, OS and software, or deployment region, see Azure VM Selector.
- For more information on Virtual Machine Scale Sets sizes, see Automatically scale machines in a VMSS.
- For more cost management planning information, see the Plan and manage your Azure costs module.

Last updated on 02/05/2026

Azure Cobalt processor-based Virtual Machines

Article • 03/25/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Cobalt 100-based Virtual Machines (VMs) are powered by Azure's first-generation Cobalt 100 processor. These VMs offer an attractive option for a wide range of scale-out and cloud-native Linux-based workloads, including data analytics, web and application servers, open-source databases, caches, and more. These VMs utilize Microsoft's first 64-bit Arm-based CPU, which was fully designed by Microsoft. The Cobalt 100 processor, based on the Arm's Neoverse N2 design, aims to enhance performance and power efficiency for broad range of workloads. Operating at 3.4 GHz, the Azure Cobalt 100 processor provides an entire physical core for each virtual machine vCPU.

List of VM series by family

The following section lists the VM series supported by Cobalt 100 processors. To find detailed information on a series, including specifications and recommended workloads, check out the links in the *Series* column.

Cobalt 100-based VMs

Expand table

Family	Memory Ratio	Series
General Purpose	2 GiB: 1 vCPU	Dplsv6-series, Dpldsv6-series
General Purpose	4 GiB: 1 vCPU	Dpsv6-series, Dpdsv6-series
Memory Optimized	8 GiB: 1 vCPU	Epsv6-series, Epdsv6-series

To learn how to deploy these VMs for cloud-native applications on AKS, check out Azure Kubernetes Service (AKS) demo on Cobalt 100 VMs.

To find the most up-to-date regional availability, visit the Virtual Machines section at the bottom of the Product Availability by Region page.

Guest OS Support for Cobalt 100 VMs

The following section lists the Guest OS support for Cobalt 100 VMs.

Guest OS Support

Expand table

Guest OS	Version
AlmaLinux	8+
Azure Linux	2
Debian	11+
Flatcar	N/A
Kali Linux	N/A
QNX Neutrino RTOS	7.1
Red Hat Enterprise Linux (RHEL)	8.6+
SUSE Linux Enterprise Server (SLES)	15 SP4+
Ubuntu	20.04+
Windows 11 Client	N/A

To find the most up-to-date Guest OS support, visit Azure portal. Under "Instance Details," select "See all images," then click the "Image Type" filter to select "Arm64."

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

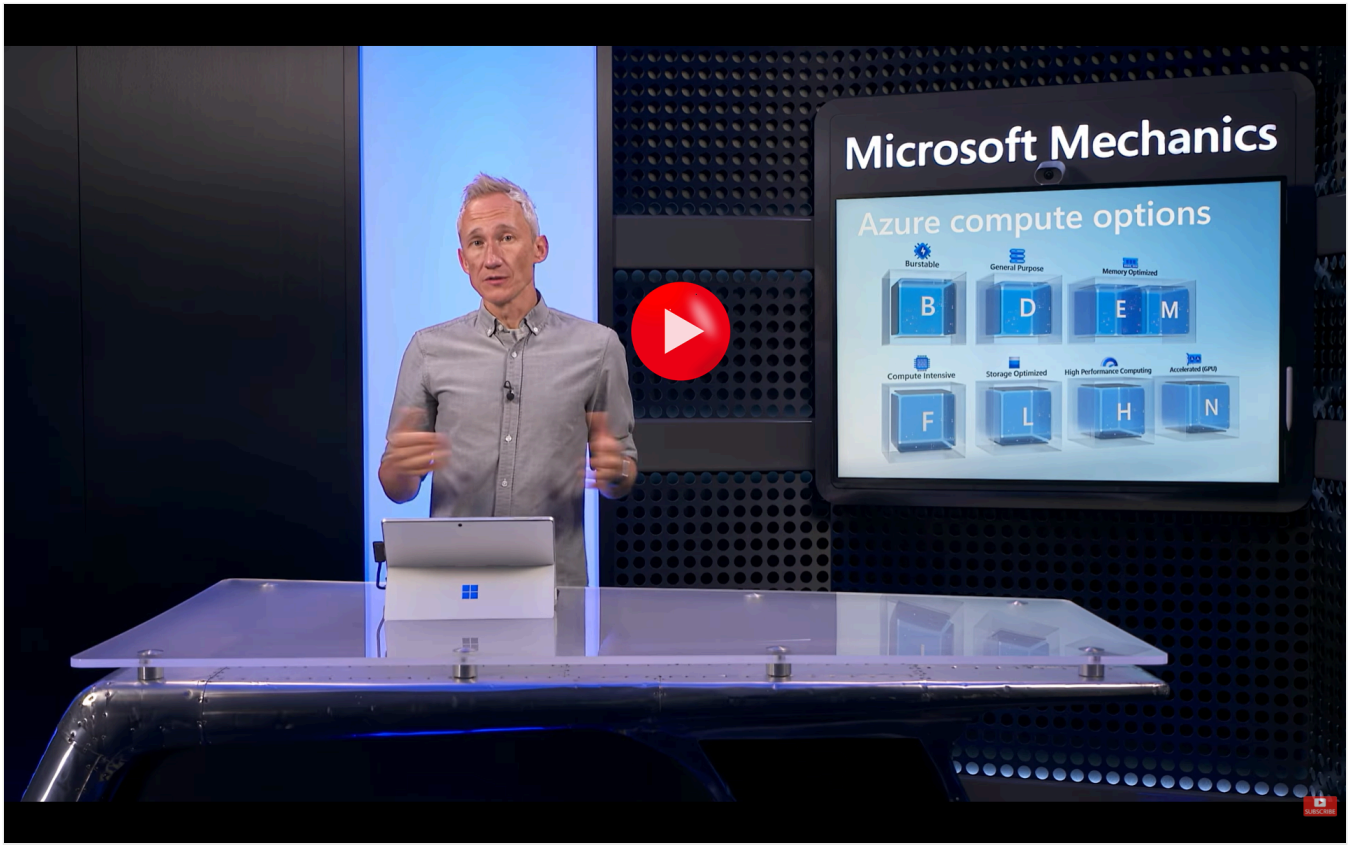
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

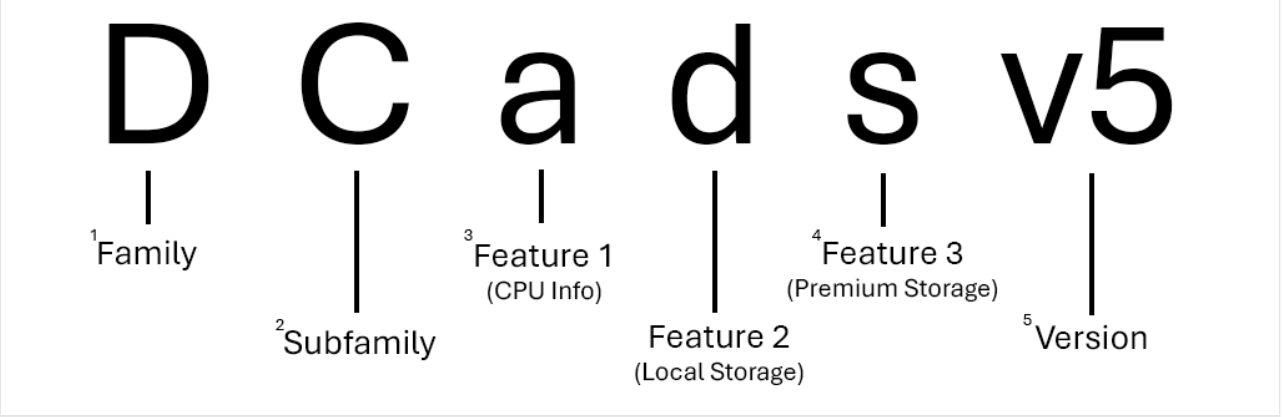
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

'A' family general purpose VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'A' family of VM size series are one of Azure's general purpose VM instances. They're designed for entry-level workloads, such as development and test environments, small to medium databases, and low-traffic web servers.

Workloads and use cases

Cost Efficiency: A-series VMs are some of the most budget-friendly options available on Azure, making them a good choice for projects with limited financial resources or those that do not require high-performance compute capabilities.

General Workloads: A-series VMs are well-suited for handling basic applications, light web servers, and small databases that do not demand extensive CPU, memory, or I/O performance.

Entry-Level Applications: A-series VMs can serve as a good starting point for deploying applications that are not expected to scale significantly. They provide a platform for applications and services that require less processing power.

Series in family

Av2-series

The Av2-series VMs can be deployed on a variety of hardware types and processors. Av2-series run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), the Intel® Xeon® Platinum 8272CL (Cascade Lake), the Intel® Xeon® 8171M 2.1 GHz (Skylake), the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors. Av2-series VMs have CPU performance and memory configurations best suited for entry level workloads like development and test. The size is throttled to offer consistent processor performance for the running instance, regardless of the hardware it is deployed on. To determine the physical hardware on which this size is deployed, query the virtual hardware from within the Virtual Machine. Some example use cases include development and test servers, low traffic web servers, small to medium databases, proof-of-concepts, and code repositories.

View the full Av2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 8 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M 2.1 GHz (Skylake) [x86-64] Intel Xeon E5-2673 v4 2.3 GHz (Broadwell) [x86-64] Intel Xeon E5-2673 v3 2.4 GHz (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	2 - 64 GiB	
Local Storage	1 Disk	10 - 80 GiB 2x500 - 16x500 IOPS 10 - 80 MBps
Remote Storage	2 - 16 Disks	
Network	2 - 8 NICs	250 - 2000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Previous-generation A family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 07/03/2025

Av2 sizes series

Summarize this article for me

The Av2-series VMs can be deployed on a variety of hardware types and processors. Av2-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1 GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors. Av2-series VMs have CPU performance and memory configurations best suited for entry level workloads like development and test. The size is throttled to offer consistent processor performance for the running instance, regardless of the hardware it is deployed on. To determine the physical hardware on which this size is deployed, query the virtual hardware from within the Virtual Machine. Some example use cases include development and test servers, low traffic web servers, small to medium databases, proof-of-concepts, and code repositories.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 8 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M 2.1 GHz (Skylake) [x86-64] Intel Xeon E5-2673 v4 2.3 GHz (Broadwell) [x86-64] Intel Xeon E5-2673 v3 2.4 GHz (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	2 - 64 GiB	
Local Storage	1 Disk	10 - 80 GiB 2x500 - 16x500 IOPS 10 - 80 MBps
Remote Storage	2 - 16 Disks	Disk Types: Standard SDD/HDD
Network	2 - 8 NICs	250 - 2,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_A1_v2	1	2
Standard_A2_v2	2	4
Standard_A4_v2	4	8
Standard_A8_v2	8	16
Standard_A2m_v2	2	16
Standard_A4m_v2	4	32
Standard_A8m_v2	8	64

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1

Feature name	Support status
Accelerated Networking	Not Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

'B' family general purpose VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'B' family of VM size series are one of Azure's general purpose VM instances. While traditional Azure virtual machines provide fixed CPU performance, B-series virtual machines are the only VM type that use credits for CPU performance provisioning. B-series VMs utilize a CPU credit model to track how much CPU is consumed - the virtual machine accumulates CPU credits when a workload is operating below the base CPU performance threshold and uses credits when running above the base CPU performance threshold until all of its credits are consumed. Upon consuming all the CPU credits, a B-series virtual machine is throttled back to its base CPU performance until it accumulates the credits to CPU burst again.

Read more about the B-series CPU credit model.

Workloads and use cases

Usage Flexibility: B-family VMs are best suited for workloads that do not require constant full CPU performance.

Ideal Applications: B-family VMs are ideal applications include web servers, proof of concepts, small databases, and development build environments.

Performance Needs: Some workloads often have burstable performance requirements, meaning they only need high performance sporadically. B-family VMs are perfect for this use case.

Series in family

Basv2-series

Basv2-series virtual machines run on the AMD's third Generation EPYC™ 7763v processor in a multi-threaded configuration with up to 256 MB L3 cache configuration. This series provides low cost CPU burstable general purpose virtual machines. Basv2-series virtual machines utilize a CPU credit model to track how much CPU is consumed. The virtual machine accumulates CPU credits when a workload is operating below the base CPU performance threshold and uses credits when running above the base CPU performance threshold, until all of its credits are consumed. Upon consuming all the CPU credits, a Basv2-series virtual machine is throttled back to its base CPU performance until it accumulates the credits to CPU burst again.

Basv2-series virtual machines offer a balance of compute, memory, and network resources, and are a cost effective way to run a broad spectrum of general purpose workloads. These include large scale micro-services, small and medium databases, virtual desktops, and business-critical applications. This series also offers an affordable option to run your code repositories and dev/test environments. Basv2-Series offers virtual machines of up-to 32 vCPU and 128 Gib of RAM, with max network bandwidth of upto 6,250 Mbps and max uncached disk throughput of 600 Mbps. Basv2-series virtual machines also support attachments of Standard SSD, Standard HDD, Premium SSD disk types with a default Remote-SSD support. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

View the full Basv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	AMD EPYC 7763v (Milan) [x86-64]
Memory	1 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 25,600 IOPS 85 - 600 MBps
Network	2 - 4 NICs	6250 Mbps
Accelerators	None	

Bpsv2-series

The Bpsv2-series virtual machines are based on the Arm architecture, featuring the Ampere® Altra® Arm-based processor operating at 3.0 GHz, delivering outstanding price-performance for general-purpose workloads. These virtual machines offer a range of VM sizes, from 0.5 GiB to up to 4 GiB of memory per vCPU, to meet the needs of applications that don't need the full performance of the CPU continuously, such as development and test servers, low traffic web servers, small databases, micro services, servers for proof-of-concepts, build servers, and code repositories. These workloads typically have burstable performance requirements. The Bpsv2-series VMs provides you with the ability to purchase a VM size with baseline performance that can build up credits when it's using less than its baseline performance. When the VM has accumulated credits, the VM can burst above the baseline using up to 100% of the vCPU when your application requires higher CPU performance.

View the full Bpsv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 16 vCPUs	Ampere Altra [Arm64]
Memory	1 - 64 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 25,600 IOPS 85 - 600 MBps
Network	2 - 4 NICs	6250 Mbps
Accelerators	None	

Bsv2-series

Bsv2-series virtual machines run on Intel® Xeon® Platinum 8473C (Sapphire Rapids), or Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing low-cost CPU burstable general purpose virtual machines. Bsv2-series virtual machines utilize a CPU credit model to track how much CPU is consumed. The virtual machine accumulates CPU credits when a workload is operating below the base CPU performance threshold and, uses credits when running above the base CPU performance threshold until all of its credits are consumed. Upon consuming all the CPU credits, a Bsv2-series virtual machine is throttled back to its base CPU performance until it accumulates the credits to CPU burst again.

Bsv2-series virtual machines offer a balance of compute, memory, and network resources and are a cost-effective way to run a broad spectrum of general-purpose workloads. These include large-scale micro-services, small and medium databases, virtual desktops, and business-critical applications; and are also an affordable option to run your code repositories and dev/test environments. Bsv2-Series offers virtual machines of up to 32 vCPU and 128 Gib of RAM, with max network bandwidth of up to 6,250 Mbps and max uncached disk throughput of 600 Mbps. Bsv2-series virtual machines also support attachments of Standard SSD, Standard HDD, and Premium SSD disk types with a default Remote-SSD support, you can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

View the full Bsv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64]
Memory	1 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 51,200 IOPS 85 - 600 MBps
Network	2 - 4 NICs	6250 Mbps
Accelerators	None	

Previous-generation B family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 11/06/2025

B Series CPU Credit Model

Article • 10/31/2024

While traditional Azure virtual machines provide fixed CPU performance, B-series virtual machines are the only VM type that use credits for CPU performance provisioning. B-series VMs utilize a CPU credit model to track how much CPU is consumed - the virtual machine accumulates CPU credits when a workload is operating below the base CPU performance threshold and uses credits when running above the base CPU performance threshold until all of its credits are consumed. Upon consuming all the CPU credits, a B-series virtual machine is throttled back to its base CPU performance until it accumulates the credits to CPU burst again.

Credit concepts and definitions

- Base CPU performance = The minimum CPU performance threshold a VM will have available always. This level sets the bar for net credit accumulation when the CPU utilization is below the base CPU performance level and net credit consumption when the CPU utilization is above the base CPU performance.
- Initial Credits = The number of credits allocated to a B-series virtual machine when a VM is deployed.
- Credits banked/hour = The number of credits a B-series virtual machine accumulates per hour if the VM is idle (no CPU performance consumption).
- Max Banked Credits = The maximum number/upper limit of credits a B-series virtual machine can accumulate. Upon reaching this upper limit, a B-series VM can no longer accumulate more credits.
- CPU Credits Consumed = The number of CPU credits spent during the measurement time-period.
- CPU Credits Remaining = The number of CPU credits available to consume for a given B-series VM.
- Percentage CPU = CPU performance of a given VM during a measurement period.

Credits accumulation and consumption

The credit accumulation and consumption rates are set such that a VM running at exactly its base performance level will have neither a net accumulation or consumption

of bursting credits. A VM has a net credit increase whenever it's running below its base CPU performance level and will have a net decrease in credits whenever the VM is utilizing the CPU more than its base CPU performance level.

To conduct calculations on credit accumulations and consumptions, customers can utilize the holistic 'credits banked per minute' formula => `((Base CPU performance * number of vCPU) - (Percentage CPU * number of vCPU))/100`.

Putting this calculation into action, let's say that a customer deploys the Standard_B2ts_v2 VM size and their workload demands 10% of the 'Percentage CPU' or CPU performance, then the 'credits banked per minute' calculation will be as follows: `((20*2) - (10*2))/100 = 0.2 credits/minute`. In such a scenario, a B-series VM is accumulating credits given the 'Percentage CPU' per CPU performance requirement is below the 'Base CPU performance' of the Standard_B2ts_v2.

Similarly, utilizing the example of a Standard_B32as_v2 VM size, if the workload demands 60% of the CPU performance for a measurement of time - then the 'credits banked per minute' calculation will be as follows: `((40*32) - (60*32))/100 = -6.4 credits/minute`. Here the negative result implies the B-series VM is consuming credits given the 'Percentage CPU'/CPU performance requirement is above the 'Base CPU performance' of the Standard_B32as_v2.

Credit monitoring

To monitor B-series specific credit metrics, customers can utilize the Azure monitor data platform, see Overview of metrics in Microsoft Azure. Azure monitor data platform can be accessed via the Azure portal and other orchestration paths, and via programmatic API calls to Azure monitor. Via Azure monitor data platform, customers can access B-series credit model-specific metrics such as 'CPU Credits Consumed', 'CPU Credits Remaining', and 'Percentage CPU' for their given B-series size in real time.

Other sizes and information

- General purpose
- Compute optimized
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute

Pricing Calculator: Pricing Calculator

More information on Disks Types: Disk Types

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Bsv2-series sizes

Summarize this article for me

Bsv2-series virtual machines run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), or Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing low-cost CPU burstable general purpose virtual machines. Bsv2-series virtual machines utilize a CPU credit model to track how much CPU is consumed. The virtual machine accumulates CPU credits when a workload is operating below the base CPU performance threshold and, uses credits when running above the base CPU performance threshold until all of its credits are consumed. Upon consuming all the CPU credits, a Bsv2-series virtual machine is throttled back to its base CPU performance until it accumulates the credits to CPU burst again.

Bsv2-series virtual machines offer a balance of compute, memory, and network resources and are a cost-effective way to run a broad spectrum of general-purpose workloads. These include large-scale micro-services, small and medium databases, virtual desktops, and business-critical applications; and are also an affordable option to run your code repositories and dev/test environments. Bsv2-series offers virtual machines of up to 32 vCPU and 128 GiB of RAM, with max network bandwidth of up to 6,250 Mbps and max uncached disk throughput of 600 Mbps. Bsv2-series virtual machines also support attachments of Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Read more about the B-series CPU credit model.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	1 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 51,000 IOPS 85 - 600 MBps Disk Types: Standard SDD/HDD, Premium SSD

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - 4 NICs	6,250 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_B2ts_v2	2	1
Standard_B2ls_v2	2	4
Standard_B2s_v2	2	8
Standard_B4ls_v2	4	8
Standard_B4s_v2	4	16
Standard_B8ls_v2	8	16
Standard_B8s_v2	8	32
Standard_B16ls_v2	16	32
Standard_B16s_v2	16	64
Standard_B32ls_v2	32	64
Standard_B32s_v2	32	128

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Basv2 sizes series

Summarize this article for me

Basv2-series virtual machines run on the AMD's third Generation EPYC™ 7763v processor in a multi-threaded configuration with up to 256 MB L3 cache configuration. This series provides low cost CPU burstable general purpose virtual machines. Basv2-series virtual machines utilize a CPU credit model to track how much CPU is consumed. The virtual machine accumulates CPU credits when a workload is operating below the base CPU performance threshold and uses credits when running above the base CPU performance threshold, until all of its credits are consumed. Upon consuming all the CPU credits, a Basv2-series virtual machine is throttled back to its base CPU performance until it accumulates the credits to CPU burst again.

Basv2-series virtual machines offer a balance of compute, memory, and network resources, and are a cost effective way to run a broad spectrum of general purpose workloads. These include large scale micro-services, small and medium databases, virtual desktops, and business-critical applications. This series also offers an affordable option to run your code repositories and dev/test environments. Basv2-Series offers virtual machines of up-to 32 vCPU and 128 Gib of RAM, with max network bandwidth of upto 6,250 Mbps and max uncached disk throughput of 600 Mbps. Basv2-series virtual machines also support attachments of Standard SSD, Standard HDD, Premium SSD types with a default Remote-SSD support. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Read more about the B-series CPU credit model.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	AMD EPYC 7763v (Milan) [x86-64]
Memory	1 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 26,000 IOPS 85 - 600 MBps Disk Types: Premium SSD
Network	2 - 4 NICs	6,250 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
		Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_B2ats_v2	2	1
Standard_B2als_v2	2	4
Standard_B2as_v2	2	8
Standard_B4als_v2	4	8
Standard_B4as_v2	4	16
Standard_B8als_v2	8	16
Standard_B8as_v2	8	32
Standard_B16als_v2	16	32
Standard_B16as_v2	16	64
Standard_B32als_v2	32	64
Standard_B32as_v2	32	128

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Bpsv2 sizes series

Summarize this article for me

The Bpsv2-series virtual machines are based on the Arm architecture, featuring the Ampere® Altra® Arm-based processor operating at 3.0 GHz, delivering outstanding price-performance for general-purpose workloads. These virtual machines offer a range of VM sizes, from 0.5 GiB to up to 4 GiB of memory per vCPU, to meet the needs of applications that don't need the full performance of the CPU continuously, such as development and test servers, low traffic web servers, small databases, micro services, servers for proof-of-concepts, build servers, and code repositories. These workloads typically have burstable performance requirements. The Bpsv2-series VMs provides you with the ability to purchase a VM size with baseline performance that can build up credits when it's using less than its baseline performance. When the VM has accumulated credits, the VM can burst above the baseline using up to 100% of the vCPU when your application requires higher CPU performance.

Read more about the B-series CPU credit model.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 16 vCPUs	Ampere Altra [Arm64]
Memory	1 - 64 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 26,000 IOPS 85 - 600 MBps Disk Types: Premium SSD
Network	2 - 4 NICs	6,250 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_B2pts_v2	2	1
Standard_B2pls_v2	2	4
Standard_B2ps_v2	2	8
Standard_B4pls_v2	4	8
Standard_B4ps_v2	4	16
Standard_B8pls_v2	8	16
Standard_B8ps_v2	8	32
Standard_B16pls_v2	16	32
Standard_B16ps_v2	16	64

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

'D' family general purpose VM size series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'D' family of VM sizes are one of Azure's general purpose VM sizes. They're designed for a variety of demanding workloads, such as enterprise applications, web and application servers, development and test environments, and batch processing tasks. Equipped with faster processors and more memory per core than the A-series, D-series VMs offer a strong performance balance, making them suitable for applications that require both high computational power and substantial memory resources. They are particularly favored for running enterprise-grade applications, supporting moderate to high-traffic web servers, and performing data-intensive batch processing.

Workloads and use cases

Balanced Performance: D-series VMs provide a solid balance between CPU capabilities and memory size, which makes them suitable for most production workloads. They are equipped with faster processors compared to the A-series and provide more memory per core.

Enterprise Applications: They are well-suited for running enterprise applications like SAP, Microsoft Dynamics, or large relational databases that require both high computational power and substantial memory.

Development and Test Environments: With their balanced resources, D-series VMs are ideal for development and testing environments where developers need to simulate production conditions closely.

Web and Application Servers: They provide the necessary resources to host web servers and application servers that experience moderate to heavy traffic, ensuring smooth and responsive user experiences.

Batch Processing: D-series VMs are efficient for handling batch processing tasks that require processing large amounts of data quickly, thanks to their fast processors and ample memory.

Gaming Servers: The high-performance capabilities of D-series VMs make them suitable for gaming servers where latency and speed are critical for a good user experience.

Series in family

Dsv7 series and Ddsv7 series

Dsv7 series

Dsv7-series virtual machines run on the Intel® Xeon® 6 6973PC (Granite Rapids) processor, reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. Dsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation for example increased networking, increased remote storage performance, upgraded CPU, and elevated memory bandwidth. Dsv7-series virtual machines are ideal for many general computing workloads such as e-commerce systems, web applications, desktop virtualization solutions, application servers, and more.

Dsv7-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD V2 disk storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

View the full Dsv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 372 vCPUs	Intel Xeon 6 6973PC (Granite Rapids) [x86-64]
Memory	8 - 1,488 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4,000 - 500,000 IOPS 115 - 16,000 MBps
Network	3 - 15 NICs	16,000 - 200,000 Mbps
Accelerators	None	

Dlsv7 series and Dldsv7 series

Dlsv7 series

Dlsv7-series virtual machines run on Intel® Xeon® 6 6973PC (Granite Rapids) processor reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2

GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 248 vCPUs and 496 GiB of RAM. Dlsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation, for example increased networking, increased remote storage performance, upgraded CPU, and elevated memory bandwidth. These virtual machine sizes can reduce costs when running non-memory intensive applications. Dlsv7-series virtual machines are ideal for many general computing workloads such as e-commerce systems, web applications, desktop virtualization solutions, application servers, and more.

Dlsv7-series virtual machines don't have any temporary storage, thus lowering the price of entry. You can attach Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines

View the full Dlsv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 248 vCPUs	Intel Xeon 6 6973PC (Granite Rapids) [x86-64]
Memory	4 - 496 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4,000 - 396,800 IOPS 115 - 13,145 MBps
Network	3 - 15 NICs	16,000 - 150,000 Mbps
Accelerators	None	

Dasv7 series and Dadsv7 series

Dasv7 series

Dasv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 640 GiB of RAM. The Dasv7-series sizes offer a combination of vCPU and memory for most production workloads. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

View the full Dasv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	8 - 640 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4000 – 212000 IOPS 118 – 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 – 310000 IOPS 136 – 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

Dalsv7 series and Daldsv7 series

Dalsv7 series

Dalsv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 320 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

View the full Dalsv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	4 - 320 GiB	
Local Storage	None	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

Dpsv6 and Dplsv6-series

Dpsv6-series

The Dpsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for general-purpose workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 4 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs meet the requirements of most scale-out enterprise workloads such as web and application servers, small to medium databases, caches, and more. Dpsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Dpsv6-series virtual machines offer up to 96 vCPU and 384 GiB of RAM. These VMs come without local disk, providing a better value proposition for workloads that don't require local temporary storage.

View the full Dpsv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [Arm64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	3750 - 153600 IOPS

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Premium SSD		106 - 5000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4163 - 199987 IOPS 124 - 5850 MBps
Network	2 - 8 NICs	12500 - 60000 Mbps
Accelerators	None	

Dpdsv6 and Dpldsv6-series

Dpdsv6-series

The Dpdsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for general-purpose workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 4 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs meet the requirements of most scale-out enterprise workloads such as web and application servers, small to medium databases, caches, and more. Dpdsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Dpdsv6-series virtual machines offer up to 96 vCPU, up to 384 GiB of RAM, and up to 5,280 GiB of fast local NVMe temporary storage.

View the full Dpdsv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [Arm64]
Memory	8 - 384 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB/Disk 15000 - 1800000 IOPS 90 - 8640 MBps
Remote Storage Premium SSD	8 - 64 Disks	3750 - 153600 IOPS 106 - 5000 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4163 - 199987 IOPS 124 - 5850 MBps
Network	2 - 8 NICs	12500 - 60000 Mbps
Accelerators	None	

Dasv6 and Dadsv6-series

Dasv6-series

Dasv6-series VMs utilize AMD's 4th Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. These virtual machines offer up to 96 vCPU and 384 GiB of RAM. The Dasv6-series sizes offer a combination of vCPU and memory for most production workloads. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage.

View the full Dasv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GB	
Local Storage	None	
Remote Storage Premium SSD	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Remote Storage Premium SSD v2 / Ultra Disks	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

Dalsv6 and Daldsv6-series

Dalsv6-series

Dalsv6-series VMs utilize AMD's 4th Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. These virtual machines offer up to 96 vCPU and 192 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage.

View the full Dalsv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	4 - 192 GiB	
Local Storage	None	
Remote Storage Premium SSD	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Remote Storage Premium SSD v2 / Ultra Disks	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

Dv5 and Dsv5-series

Dv5-series

The Dv5 and Dsv5-series virtual machines run on Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing a better value proposition for most general-purpose workloads. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer a combination of vCPUs and memory to meet the requirements associated with most enterprise workloads, such as small-to-medium databases, low-to-medium traffic web servers, application servers and more. The Dv5 and Dsv5-series provide a better value proposition for workloads that don't

require local temp disk. For information about similar virtual machines with local disk, see Ddv5 and Ddsv5-series VMs.

View the full Dv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 80000 IOPS 85 - 2600 MBps
Network	2 - 8 NICs	12500 - 35000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Ddv5 and Ddsv5-series

Ddv5-series

The Ddv5 and Ddsv5-series Virtual Machines run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing a better value proposition for most general-purpose workloads. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer a combination of vCPUs, memory and temporary storage able to meet the requirements associated with most enterprise workloads, such as small-to-medium databases, low-to-medium traffic web servers, application servers and more.

View the full Ddv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	75 - 3600 GiB 9000 - 450000 IOPS (RR) 125 - 4000 MBps (RR)
Remote Storage	4 - 32 Disks	3750 - 80000 IOPS 85 - 2600 MBps
Network	2 - 8 NICs	12500 - 35000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Dasv5 and Dadsv5-series

Dasv5-series

Dasv5-series VMs utilize AMD's 3rd Generation EPYC™ 7763v and AMD's 4th Generation EPYC™ 9004 processors. The Dasv5-series sizes offer a combination of vCPU and memory for most production workloads. The new VMs with no local disk provide a better value proposition for workloads that do not require local temp disk.

View the full Dasv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7763v (Milan) [x86-64] , AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 80000 IOPS 82 - 1600 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	None	

Dpsv5 and Dpdsv5-series

Dpsv5-series

The Dpsv5-series and Dpdsv5-series virtual machines are based on the Arm architecture, delivering outstanding price-performance for general-purpose workloads. These virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer a range of vCPU sizes, up to 4 GiB of memory per vCPU, and temporary storage options able to meet the requirements of scale-out and most enterprise workloads such as web and application servers, small to medium databases, caches, and more.

View the full Dpsv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Ampere Altra [Arm64]
Memory	8 - 208 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 80000 IOPS 85 - 1735 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

Dplsv5 and Dpldsv5-series

Dplsv5-series

The Dplsv5-series and Dpldsv5-series virtual machines are based on the Arm architecture, delivering outstanding price-performance for general-purpose workloads. These virtual

machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer a range of vCPU sizes, up to 2 GiB of memory per vCPU, and temporary storage options able to meet the requirements of most non-memory-intensive and scale-out workloads such as microservices, small databases, caches, gaming servers, and more.

View the full Dplsv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Ampere Altra [Arm64]
Memory	4 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 80000 IOPS 85 - 1735 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

Dlsv5 and Dldsv5-series

Dlsv5-series

Dlsv5-series virtual machines run on Intel® Xeon® Platinum 8473C (Sapphire Rapids), or Intel® Xeon® Platinum 8370C (Ice Lake) processor reaching an all core turbo clock speed of up to 3.5 GHz. These virtual machines offer up to 96 vCPU and 192 GiB of RAM. These VM sizes can reduce cost when not running memory intensive applications.

Dlsv5-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. See pricing for disks.

View the full Dlsv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 192 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 80000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 35000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Dv4 and Dsv4-series

Dv4-series

The Dv4 and Dsv4-series run on Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), or Intel® Xeon® Platinum 8272CL (Cascade Lake) processors in a hyper-threaded configuration, providing a better value proposition for most general-purpose workloads. It features an all core Turbo clock speed of 3.4 GHz, Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and Intel® Advanced Vector Extensions 512 (Intel® AVX-512). They also support Intel® Deep Learning Boost.

View the full Dv4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	None	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 48 - 1200 MBps
Network	2 - 8 NICs	5000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Dav4 and Dasv4-series

Dav4-series

The Dav4-series run on 2nd Generation AMD EPYC™ 7452 (up to 3.35GHz) or 3rd Generation EPYC™ 7763v processors (up to 3.5GHz). The Dav4-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads. Data disk storage is billed separately from virtual machines. To use premium SSD, use the Dasv4 sizes. The pricing and billing meters for Dasv4 sizes are the same as the Dav4-series.

View the full Dav4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7452 (Rome) [x86-64] AMD EPYC 7763v (Milan) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	50 - 2400 GiB 3000 - 96000 IOPS (RR) 46 - 1000 MBps (RR)
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 48 - 1200 MBps
Network	2 - 8 NICs	2000 - 40000 Mbps
Accelerators	None	

Ddv4 and Ddsv4-series

Ddv4-series

The Ddv4 and Ddsv4-series run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake) or the Intel® Xeon® Platinum 8272CL (Cascade Lake) processors in a hyper-threaded configuration, providing a better value proposition for most general-purpose workloads. It features an all core Turbo clock speed of 3.4 GHz, Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and Intel® Advanced Vector Extensions 512 (Intel® AVX-512). They also support Intel® Deep Learning Boost. These new VM sizes will have 50% larger local storage, as well as better local disk IOPS for both read and write compared to the Dv3/Dsv3 sizes with Gen2 VMs. D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

View the full Ddv4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	1 Disk	75 - 2400 GiB 9000 - 300000 IOPS (RR) 125 - 4000 MBps (RR)
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 48 - 1200 MBps
Network	2 - 8 NICs	5000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Dv3 and Dsv3-series

Dv3-series

The Dv3-series run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads. Memory has been expanded (from ~3.5 GiB/vCPU to 4 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Dv3-series no longer has the high memory VM sizes of the D/Dv2-series, those have been moved to the memory optimized Ev3 and Esv3-series. Example D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

View the full Dv3-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	1 Disk	50 - 1600 GiB 3000 - 96000 IOPS (RR) 46 - 1000 MBps (RR)
Remote Storage	4 - 32 Disks	
Network	2 - 8 NICs	1000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Dv2 and Dsv2-series

Dv2-series

The Dv2 and DSv2-series, a follow-on to the original D-series, feature a more powerful CPU and optimal CPU-to-memory configuration making them suitable for most

production workloads. The Dv2-series is about 35% faster than the D-series. Dv2-series run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with the Intel Turbo Boost Technology 2.0. The Dv2-series has the same memory and disk configurations as the D-series.

View the full Dv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 16 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	3.5 - 56 GiB	
Local Storage	1 Disk	50 - 800 GiB 3000 - 48000 IOPS (RR) 46 - 750 MBps (RR)
Remote Storage	4 - 64 Disks	4x500 - 64x500 IOPS
Network	2 - NICs	750 - 12000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Previous-generation D family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 02/03/2026

Dsv7 sizes series

Summarize this article for me

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Dsv7-series virtual machines run on the Intel® Xeon® 6 6973PC (Granite Rapids) processor, reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. Dsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation for example increased networking, increased remote storage performance, upgraded CPU, and elevated memory bandwidth. Dsv7-series virtual machines are ideal for many general computing workloads such as e-commerce systems, web applications, desktop virtualization solutions, application servers, and more.

Dsv7-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 372 vCPUs	Intel Xeon 6 6973PC (Granite Rapids) [x86-64]
Memory	8 - 1,488 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4,000 - 500,000 IOPS 115 - 16,000 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	3 - 15 NICs	16,000 - 200,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2s_v7	2	8
Standard_D4s_v7	4	16
Standard_D8s_v7	8	32
Standard_D16s_v7	16	64
Standard_D32s_v7	32	128
Standard_D48s_v7	48	192
Standard_D64s_v7	64	256
Standard_D96s_v7	96	384
Standard_D128s_v7	128	512
Standard_D192s_v7	192	768
Standard_D248s_v7	248	992
Standard_D372s_v7	372	1,488

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dlsv7 sizes series

Summarize this article for me

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Dlsv7-series virtual machines run on Intel® Xeon® 6 6973PC (Granite Rapids) processor reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 248 vCPUs and 496 GiB of RAM. Dlsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation, for example increased networking, increased remote storage performance, upgraded CPU, and elevated memory bandwidth. These virtual machine sizes can reduce costs when running non-memory intensive applications. Dlsv7-series virtual machines are ideal for many general computing workloads such as e-commerce systems, web applications, desktop virtualization solutions, application servers, and more.

Dlsv7-series virtual machines don't have any temporary storage, thus lowering the price of entry. You can attach Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 248 vCPUs	Intel Xeon 6 6973PC (Granite Rapids) [x86-64]
Memory	4 - 496 GiB	
Local Storage	None	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage	10 - 64 Disks	4,000 - 397,000 IOPS 115 - 13,145 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	3 - 15 NICs	16,000 - 150,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2ls_v7	2	4
Standard_D4ls_v7	4	8
Standard_D8ls_v7	8	16
Standard_D16ls_v7	16	32
Standard_D32ls_v7	32	64
Standard_D48ls_v7	48	96
Standard_D64ls_v7	64	126
Standard_D96ls_v7	96	384
Standard_D128ls_v7	128	256
Standard_D192ls_v7	192	384
Standard_D248ls_v7	248	496

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dldsv7 sizes series

Summarize this article for me

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Dldsv7-series virtual machines run on the Intel® Xeon® 6 6973PC (Granite Rapids) processor reaching an all-core turbo clock speed of 3.6 GHz with a max turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 248 vCPU and 496 GiB of RAM. Dldsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation with increased networking, increased remote storage performance, upgraded CPU performance, elevated memory bandwidth, faster NVMe local storage access, increased scalability, and larger local storage capacity. These virtual machine sizes can reduce cost when running non-memory intensive applications. Dldsv7-series virtual machines are ideal for many general computing workloads such as e-commerce systems, web applications, desktop virtualization solutions, application servers, and more.

Dldsv7-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 248 vCPUs	Intel Xeon 6 6973PC (Granite Rapids) [x86-64]
Memory	4 - 496 GiB	
Local Storage	1 - 6 Disks	110 - 1,760 GiB 50,000 - 6,400,000 IOPS 280 - 35,840 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage	10 - 64 Disks	4,000 - 397,000 IOPS 115 - 13,145 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	3 - 15 NICs	16,000 - 150,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2lds_v7	2	4
Standard_D4lds_v7	4	8
Standard_D8lds_v7	8	16
Standard_D16lds_v7	16	32
Standard_D32lds_v7	32	64
Standard_D48lds_v7	48	96
Standard_D64lds_v7	64	126
Standard_D96lds_v7	96	384
Standard_D128lds_v7	128	256
Standard_D192lds_v7	192	384
Standard_D248lds_v7	248	496

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Ddsv7 sizes series

Summarize this article for me

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Ddsv7-series virtual machines run on the Intel® Xeon® 6 6973PC (Granite Rapids) processor reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 372 vCPUs and 1,488 GiB of RAM. Ddsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation for example increased networking, increased remote storage performance, upgraded CPU, elevated memory bandwidth, faster NVMe local storage access, increased scalability, and larger local storage capacity. Ddsv7-series virtual machines are ideal for many general computing workloads such as e-commerce systems, web applications, desktop virtualization solutions, application servers, and more.

Ddsv7-series virtual machines offer local storage disks. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 372 vCPUs	Intel Xeon 6 6973PC (Granite Rapids) [x86-64]
Memory	8 - 1,488 GiB	
Local Storage	1 - 6 Disks	110 - 3,520 GiB 50,000 - 9,600,000 IOPS 280 - 53,760 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage	10 - 64 Disks	4,000 - 500,000 IOPS 115 - 16,000 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	3 - 15 NICs	16,000 - 200,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2ds_v7	2	8
Standard_D4ds_v7	4	16
Standard_D8ds_v7	8	32
Standard_D16ds_v7	16	64
Standard_D32ds_v7	32	128
Standard_D48ds_v7	48	192
Standard_D64ds_v7	64	256
Standard_D96ds_v7	96	384
Standard_D128ds_v7	128	512
Standard_D192ds_v7	192	768
Standard_D248ds_v7	248	992
Standard_D372ds_v7	372	1,488

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dasv7 sizes series

Summarize this article for me

Dasv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 640 GiB of RAM. The Dasv7-series sizes offer a combination of vCPU and memory for most production workloads. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	8 - 640 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4,000 - 310,000 IOPS 118 - 10,356 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 15 NICs	16,000 - 80,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2as_v7	2	8
Standard_D4as_v7	4	16
Standard_D8as_v7	8	32
Standard_D16as_v7	16	64
Standard_D32as_v7	32	128
Standard_D48as_v7	48	192
Standard_D64as_v7	64	256
Standard_D96as_v7	96	384
Standard_D128as_v7	128	512
Standard_D160as_v7	160	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported

Feature name	Support status
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dadsv7 sizes series

Summarize this article for me

Dadsv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 640 GiB of RAM. The Dadsv7-series sizes offer a combination of vCPU, memory and fast local NVMe temporary storage for most production workloads. Dadsv7-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	8 - 640 GiB	
Local Storage	1 - 6 Disks	110 - 2,200 GiB 38,000 - 3,000,000 IOPS 280 - 22,400 MBps
Remote Storage	10 - 64 Disks	4,000 - 310,000 IOPS 118 - 10,356 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 15 NICs	16,000 - 80,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ads_v7	2	8
Standard_D4ads_v7	4	16
Standard_D8ads_v7	8	32
Standard_D16ads_v7	16	64
Standard_D32ads_v7	32	128
Standard_D48ads_v7	48	192
Standard_D64ads_v7	64	256
Standard_D96ads_v7	96	384
Standard_D128ads_v7	128	512
Standard_D160ads_v7	160	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dalsv7 sizes series

Summarize this article for me

Dalsv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 320 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	4 - 320 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4,000 - 310,000 IOPS 118 - 10,356 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 15 NICs	16,000 - 80,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2als_v7	2	4
Standard_D4als_v7	4	8
Standard_D8als_v7	8	16
Standard_D16als_v7	16	32
Standard_D32als_v7	32	64
Standard_D48als_v7	48	96
Standard_D64als_v7	64	128
Standard_D96als_v7	96	192
Standard_D128als_v7	128	256
Standard_D160als_v7	160	320

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported

Feature name	Support status
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Daldsv7 sizes series

Summarize this article for me

Daldsv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPUs, 320 GiB of RAM. The Daldsv7-series sizes offer a combination of vCPU, memory and fast local NVMe temporary storage for most production workloads. These VM sizes can reduce cost when running non-memory intensive applications. Daldsv7-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	4 - 320 GiB	
Local Storage	1 - 6 Disks	110 - 2,200 GiB 38,000 - 3,000,000 IOPS 280 - 22,400 MBps
Remote Storage	10 - 64 Disks	4,000 - 310,000 IOPS 118 - 10,356 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 15 NICs	16,000 - 80,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2alds_v7	2	4
Standard_D4alds_v7	4	8
Standard_D8alds_v7	8	16
Standard_D16alds_v7	16	32
Standard_D32alds_v7	32	64
Standard_D48alds_v7	48	96
Standard_D64alds_v7	64	128
Standard_D96alds_v7	96	192
Standard_D128alds_v7	128	256
Standard_D160alds_v7	160	320

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dsv6 sizes series

Summarize this article for me

Dsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all-core turbo clock speed of 3.0 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 192 vCPU and 768 GiB of RAM. Dsv6-series virtual machines provide a better performance for most general-purpose workloads compared to the prior generation (for example, increased scalability, upgraded CPU, elevated memory bandwidth, faster remote storage access).

Dsv6-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 768 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 400,000 IOPS 106 - 12,000 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Remote Storage	8 - 64 Disks	4,000 - 400,000 IOPS 124 - 12,000 MBps
Network	2 - 8 NICs	12,500 - 82,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2s_v6	2	8
Standard_D4s_v6	4	16
Standard_D8s_v6	8	32
Standard_D16s_v6	16	64
Standard_D32s_v6	32	128
Standard_D48s_v6	48	192
Standard_D64s_v6	64	256
Standard_D96s_v6	96	384
Standard_D128s_v6	128	512
Standard_D192s_v6	192	768

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported

Feature name	Support status
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Ddsv6 sizes series

Summarize this article for me

Ddsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all-core turbo clock speed of 3.0 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 192 vCPU and 768 GiB of RAM. Ddsv6-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation with increased scalability, upgraded CPU performance, elevated memory bandwidth, faster NVMe local and remote storage access, and larger local storage capacity.

Ddsv6-series virtual machines offer local storage disks. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 768 GiB	
Local Storage	1 - 6 Disks	110 - 1,760 GiB 38,000 - 3,600,000 IOPS 180 - 17,280 MBps
Remote Storage	8 - 64 Disks	4,000 - 400,000 IOPS 106 - 12,000 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 82,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ds_v6	2	8
Standard_D4ds_v6	4	16
Standard_D8ds_v6	8	32
Standard_D16ds_v6	16	64
Standard_D32ds_v6	32	128
Standard_D48ds_v6	48	192
Standard_D64ds_v6	64	256
Standard_D96ds_v6	96	384
Standard_D128ds_v6	128	512
Standard_D192ds_v6	192	768

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported

Feature name	Support status
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dlsv6 sizes series

Summarize this article for me

Dlsv6-series virtual machines run on 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) CPU processor reaching an all-core turbo clock speed of 3.0 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 128 vCPU and 256 GiB of RAM. These virtual machine sizes can reduce cost when running non-memory intensive applications.

Dlsv6-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 256 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 267,000 IOPS 106 - 7,935 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 54,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ls_v6	2	4
Standard_D4ls_v6	4	8
Standard_D8ls_v6	8	16
Standard_D16ls_v6	16	32
Standard_D32ls_v6	32	64
Standard_D48ls_v6	48	96
Standard_D64ls_v6	64	128
Standard_D96ls_v6	96	192
Standard_D128ls_v6	128	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dldsv6 sizes series

Summarize this article for me

Dldsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all-core turbo clock speed of 3.0 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 128 vCPU and 256 GiB of RAM as well as fast, local SSD storage up to 4x1760 GiB. These virtual machine sizes can reduce cost when running non-memory intensive applications.

Dldsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 256 GiB	
Local Storage	1 - 4 Disks	110 - 1,760 GiB 38,000 - 2,400,000 IOPS 180 - 11,520 MBps
Remote Storage	8 - 64 Disks	4,000 - 267,000 IOPS 106 - 7,935 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 54,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2lds_v6	2	4
Standard_D4lds_v6	4	8
Standard_D8lds_v6	8	16
Standard_D16lds_v6	16	32
Standard_D32lds_v6	32	64
Standard_D48lds_v6	48	96
Standard_D64lds_v6	64	128
Standard_D96lds_v6	96	192
Standard_D128lds_v6	128	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dasv6-series sizes

Summarize this article for me

Dasv6-series VMs utilize AMD's 4th Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. These virtual machines offer up to 96 vCPU and 384 GiB of RAM. The Dasv6-series sizes offer a combination of vCPU and memory for most production workloads. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in the Dasv6-series

Basics

vCPUs (Qty.) and Memory for each size in the Dasv6-series.

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2as_v6	2	8
Standard_D4as_v6	4	16
Standard_D8as_v6	8	32
Standard_D16as_v6	16	64
Standard_D32as_v6	32	128
Standard_D48as_v6	48	192
Standard_D64as_v6	64	256
Standard_D96as_v6	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. **NVMe** support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dadsv6 sizes series

Summarize this article for me

Dadsv6-series VMs utilize AMD's 4th Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. These virtual machines offer up to 96 vCPU and 384 GiB of RAM. The Dadsv6-series sizes offer a combination of vCPU, memory and fast local NVMe temporary storage for most production workloads. Dadsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB 38,000 - 1,800,000 IOPS 180 - 8,640 MBps
Remote Storage	4 - 32 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ads_v6	2	8
Standard_D4ads_v6	4	16
Standard_D8ads_v6	8	32
Standard_D16ads_v6	16	64
Standard_D32ads_v6	32	128
Standard_D48ads_v6	48	192
Standard_D64ads_v6	64	256
Standard_D96ads_v6	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported

Feature name	Support status
Nested Virtualization	Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dalsv6 sizes series

Dalsv6-series VMs utilize AMD's 4th Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. These virtual machines offer up to 96 vCPU and 192 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	4 - 192 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2als_v6	2	4
Standard_D4als_v6	4	8
Standard_D8als_v6	8	16
Standard_D16als_v6	16	32
Standard_D32als_v6	32	64
Standard_D48als_v6	48	96
Standard_D64als_v6	64	128
Standard_D96als_v6	96	192

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Note

This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Daldsv6 sizes series

Summarize this article for me

Daldsv6-series VMs utilize AMD's 4th Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. These virtual machines offer up to 96 vCPUs, 192 GiB of RAM, and up to 5,280 GiB of fast local NVMe temporary storage. These VM sizes can reduce cost when running non-memory intensive applications. Daldsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	4 - 192 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB 38,000 - 1,800,000 IOPS 180 - 8,640 MBps
Remote Storage	4 - 32 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2alds_v6	2	4
Standard_D4alds_v6	4	8
Standard_D8alds_v6	8	16
Standard_D16alds_v6	16	32
Standard_D32alds_v6	32	64
Standard_D48alds_v6	48	96
Standard_D64alds_v6	64	128
Standard_D96alds_v6	96	192

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

Feature name	Support status
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dpsv6 sizes series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows Client VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Dpsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for general-purpose workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 4 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs meet the requirements of most scale-out enterprise workloads such as web and application servers, small to medium databases, caches, and more. Dpsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Dpsv6-series virtual machines offer up to 96 vCPU and 384 GiB of RAM. These VMs come without local disk, providing a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [Arm64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 200,000 IOPS 106 - 5,850 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 60,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2ps_v6	2	8
Standard_D4ps_v6	4	16
Standard_D8ps_v6	8	32
Standard_D16ps_v6	16	64
Standard_D32ps_v6	32	128
Standard_D48ps_v6	48	192
Standard_D64ps_v6	64	256
Standard_D96ps_v6	96	384

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature Support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported

Feature name	Support status
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dpdsv6 sizes series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows Client VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Dpdsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for general-purpose workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 4 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs meet the requirements of most scale-out enterprise workloads such as web and application servers, small to medium databases, caches, and more. Dpdsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Dpdsv6-series virtual machines offer up to 96 vCPU, up to 384 GiB of RAM, and up to 5,280 GiB of fast local NVMe temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [Arm64]
Memory	8 - 384 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB 15,000 - 1,800,000 IOPS 90 - 8,640 MBps
Remote Storage	8 - 64 Disks	4,000 - 200,000 IOPS 106 - 5,850 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 60,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2pds_v6	2	8
Standard_D4pds_v6	4	16
Standard_D8pds_v6	8	32
Standard_D16pds_v6	16	64
Standard_D32pds_v6	32	128
Standard_D48pds_v6	48	192
Standard_D64pds_v6	64	256
Standard_D96pds_v6	96	384

> **Note**
>
> The Dpdsv6 VM series will only work on OS images that support NVMe (i.e. NVMe drivers required for the local storage). If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature Support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dplsv6 sizes series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows Client VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Dplsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for general-purpose workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 2 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs meet the requirements of most non-memory-intensive and scale-out workloads such as microservices, small databases, caches, gaming servers, and more. Dplsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Dplsv6-series virtual machines offer up to 96 vCPU and 192 GiB of RAM. These VMs come without local disk, providing a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [Arm64]
Memory	4 - 192 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 200,000 IOPS 106 - 5,850 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 60,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2pls_v6	2	4
Standard_D4pls_v6	4	8
Standard_D8pls_v6	8	16
Standard_D16pls_v6	16	32
Standard_D32pls_v6	32	64
Standard_D48pls_v6	48	96
Standard_D64pls_v6	64	128
Standard_D96pls_v6	96	192

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature Support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported

Feature name	Support status
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dpldsv6 sizes series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows Client VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Dpldsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for general-purpose workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 2 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs meet the requirements of most non-memory-intensive and scale-out workloads such as microservices, small databases, caches, gaming servers, and more. Dpldsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Dpldsv6-series virtual machines offer up to 96 vCPU, up to 192 GiB of RAM, and up to 5,280 GiB of fast local NVMe temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [Arm64]
Memory	4 - 192 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB 15,000 - 1,800,000 IOPS 90 - 8,640 MBps
Remote Storage	8 - 64 Disks	4,000 - 200,000 IOPS 106 - 5,850 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 60,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_D2plds_v6	2	4
Standard_D4plds_v6	4	8
Standard_D8plds_v6	8	16
Standard_D16plds_v6	16	32
Standard_D32plds_v6	32	64
Standard_D48plds_v6	48	96
Standard_D64plds_v6	64	128
Standard_D96plds_v6	96	192

Note

The Dpldsv6 VM series will only work on OS images that support NVMe (i.e. NVMe drivers required for the local storage). If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature Support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dv5 sizes series

The Dv5 and Dsv5-series virtual machines run on Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing a better value proposition for most general-purpose workloads. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer a combination of vCPUs and memory to meet the requirements associated with most enterprise workloads, such as small-to-medium databases, low-to-medium traffic web servers, application servers and more. The Dv5 and Dsv5-series provide a better value proposition for workloads that don't require local temp disk. For information about similar virtual machines with local disk, see Ddv5 and Ddsv5-series VMs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 2,600 MBps Disk Types: Standard SDD/HDD
Network	2 - 8 NICs	12,500 - 35,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2_v5	2	8
Standard_D4_v5	4	16
Standard_D8_v5	8	32
Standard_D16_v5	16	64
Standard_D32_v5	32	128
Standard_D48_v5	48	192
Standard_D64_v5	64	256
Standard_D96_v5	96	384

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking1	Required

Feature name	Support status
Ephemeral OS Disks	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Dsv5 sizes series

Summarize this article for me

The Dv5 and Dsv5-series virtual machines run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing a better value proposition for most general-purpose workloads. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer a combination of vCPUs and memory to meet the requirements associated with most enterprise workloads, such as small-to-medium databases, low-to-medium traffic web servers, application servers and more. The Dv5 and Dsv5-series provide a better value proposition for workloads that don't require local temp disk. For information about similar virtual machines with local disk, see Ddv5 and Ddsv5-series VMs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 2,600 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 35,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2s_v5	2	8
Standard_D4s_v5	4	16
Standard_D8s_v5	8	32
Standard_D16s_v5	16	64
Standard_D32s_v5	32	128
Standard_D48s_v5	48	192
Standard_D64s_v5	64	256
Standard_D96s_v5	96	384

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Supported
Accelerated Networking	Required
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Ddv5 sizes series

Summarize this article for me

The Ddv5 and Ddsv5-series Virtual Machines run on the Intel® Xeon® Platinum 8573C (Emerald Rapids) or the Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing a better value proposition for most general-purpose workloads. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer a combination of vCPUs, memory and temporary storage able to meet the requirements associated with most enterprise workloads, such as small-to-medium databases, low-to-medium traffic web servers, application servers and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	75 - 3,600 GiB 9,000 - 450,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 2,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	12,500 - 35,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2d_v5	2	8
Standard_D4d_v5	4	16
Standard_D8d_v5	8	32
Standard_D16d_v5	16	64
Standard_D32d_v5	32	128
Standard_D48d_v5	48	192
Standard_D64d_v5	64	256
Standard_D96d_v5	96	384

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Required
Ephemeral OS Disks	Supported

Feature name	Support status
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Ddsv5 sizes series

The Ddv5 and Ddsv5-series Virtual Machines run on the Intel® Xeon® Platinum 8573C (Emerald Rapids) or the Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration, providing a better value proposition for most general-purpose workloads. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer a combination of vCPUs, memory and temporary storage able to meet the requirements associated with most enterprise workloads, such as small-to-medium databases, low-to-medium traffic web servers, application servers and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	75 - 3,600 GiB 9,000 - 450,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 2,600 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 35,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ds_v5	2	8
Standard_D4ds_v5	4	16
Standard_D8ds_v5	8	32
Standard_D16ds_v5	16	64
Standard_D32ds_v5	32	128
Standard_D48ds_v5	48	192
Standard_D64ds_v5	64	256
Standard_D96ds_v5	96	384

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking1	Required

Feature name	Support status
Ephemeral OS Disks	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Dasv5 sizes series

Dasv5-series VMs utilize AMD's 3rd Generation EPYC™ 7763v and AMD's 4th Generation EPYC™ 9004 processors. The Dasv5-series sizes offer a combination of vCPU and memory for most production workloads. The new VMs with no local disk provide a better value proposition for workloads that do not require local temp disk.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7763v (Milan) [x86-64] AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 82 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD, Ultra Disk
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2as_v5	2	8

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D4as_v5	4	16
Standard_D8as_v5	8	32
Standard_D16as_v5	16	64
Standard_D32as_v5	32	128
Standard_D48as_v5	48	192
Standard_D64as_v5	64	256
Standard_D96as_v5	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Dadsv5 sizes series

Summarize this article for me

Dadsv5-series utilize AMD's third Generation EPYC™ 7763v processors and AMD's 4th Generation EPYC™ 9004 processors. The Dadsv5-series sizes offer a combination of vCPU, memory, and temporary storage for most production workloads. The new VMs have 50% larger local storage and better local disk IOPS for both read and write compared to the Dav4/Dasv4 sizes with Gen2 VMs.

Dadsv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7763v (Milan) [x86-64] , AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	75 - 3,600 GiB 9,000 - 450,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 82 - 1,600 MBps Disk Types: Premium SSD
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ads_v5	2	8
Standard_D4ads_v5	4	16
Standard_D8ads_v5	8	32
Standard_D16ads_v5	16	64
Standard_D32ads_v5	32	128
Standard_D48ads_v5	48	192
Standard_D64ads_v5	64	256
Standard_D96ads_v5	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported

Feature name	Support status
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dpsv5 sizes series

Summarize this article for me

The Dpsv5-series and Dpdsv5-series virtual machines are based on the Arm architecture, delivering outstanding price-performance for general-purpose workloads. These virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer a range of vCPU sizes, up to 4 GiB of memory per vCPU, and temporary storage options able to meet the requirements of scale-out and most enterprise workloads such as web and application servers, small to medium databases, caches, and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Ampere Altra [Arm64]
Memory	8 - 208 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 1,735 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ps_v5	2	8
Standard_D4ps_v5	4	16
Standard_D8ps_v5	8	32
Standard_D16ps_v5	16	64
Standard_D32ps_v5	32	128
Standard_D48ps_v5	48	192
Standard_D64ps_v5	64	208

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dpdsv5 sizes series

Summarize this article for me

The Dpsv5-series and Dpdsv5-series virtual machines are based on the Arm architecture, delivering outstanding price-performance for general-purpose workloads. These virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer a range of vCPU sizes, up to 4 GiB of memory per vCPU, and temporary storage options able to meet the requirements of scale-out and most enterprise workloads such as web and application servers, small to medium databases, caches, and more.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	2 - 64 vCPUs	Ampere Altra [Arm64]
Memory	8 - 208 GiB	
Local Storage	1 Disk	75 - 2,400 GiB 9,000 - 300,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 1,735 MBps Disk Types: Premium SSD
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2pds_v5	2	8
Standard_D4pds_v5	4	16
Standard_D8pds_v5	8	32
Standard_D16pds_v5	16	64
Standard_D32pds_v5	32	128
Standard_D48pds_v5	48	192
Standard_D64pds_v5	64	208

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Supported
Local temporary storage	Supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dplsv5 sizes series

Summarize this article for me

The Dplsv5-series and Dpldsv5-series virtual machines are based on the Arm architecture, delivering outstanding price-performance for general-purpose workloads. These virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer a range of vCPU sizes, up to 2 GiB of memory per vCPU, and temporary storage options able to meet the requirements of most non-memory-intensive and scale-out workloads such as microservices, small databases, caches, gaming servers, and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Ampere Altra [Arm64]
Memory	4 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 1,735 MBps Disk Types: Premium SSD
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2pls_v5	2	4
Standard_D4pls_v5	4	8
Standard_D8pls_v5	8	16
Standard_D16pls_v5	16	32
Standard_D32pls_v5	32	64
Standard_D48pls_v5	48	96
Standard_D64pls_v5	64	128

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dpldsv5 sizes series

Summarize this article for me

The Dplsv5-series and Dpldsv5-series virtual machines are based on the Arm architecture, delivering outstanding price-performance for general-purpose workloads. These virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer a range of vCPU sizes, up to 2 GiB of memory per vCPU, and temporary storage options able to meet the requirements of most non-memory-intensive and scale-out workloads such as microservices, small databases, caches, gaming servers, and more.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	2 - 64 vCPUs	Ampere Altra [Arm64]
Memory	4 - 128 GiB	
Local Storage	1 Disk	75 - 2,400 GiB 9,000 - 300,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 1,735 MBps Disk Types: Premium SSD
Network	2 - 8 NICs	12,500 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2plds_v5	2	4
Standard_D4plds_v5	4	8
Standard_D8plds_v5	8	16
Standard_D16plds_v5	16	32
Standard_D32plds_v5	32	64
Standard_D48plds_v5	48	96
Standard_D64plds_v5	64	128

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Supported
Local temporary storage	Supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dlsv5 sizes series

Summarize this article for me

Dlsv5-series virtual machines run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), or Intel® Xeon® Platinum 8370C (Ice Lake) processor reaching an all core turbo clock speed of up to 3.5 GHz. These virtual machines offer up to 96 vCPU and 192 GiB of RAM. These VM sizes can reduce cost when not running memory intensive applications.

Dlsv5-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSD, Standard HDD, Premium SSD, and Premium SSD v2 disk storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. See pricing for disks.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 192 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 4,000 MBps Disk Types: Standard HDD/SSD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 - 35,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ls_v5	2	4
Standard_D4ls_v5	4	8
Standard_D8ls_v5	8	16
Standard_D16ls_v5	16	32
Standard_D32ls_v5	32	64
Standard_D48ls_v5	48	96
Standard_D64ls_v5	64	128
Standard_D96ls_v5	96	192

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2

Feature name	Support status
Accelerated Networking	Required
Ephemeral OS Disks	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dldsv5 sizes series

Summarize this article for me

Dldsv5-series virtual machines run on the Intel® Xeon® Platinum 8573C (Emerald Rapids) or the Intel® Xeon® Platinum 8370C (Ice Lake) processor reaching an all core turbo clock speed of up to 3.5 GHz. These virtual machines offer up to 96 vCPU and 192 GiB of RAM and fast, local SSD storage up to 3,600 GiB. These VM sizes can reduce cost when not running memory intensive applications.

Dldsv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. See pricing for disks.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 192 GiB	
Local Storage	1 Disk	75 - 3,600 9,000 - 450,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 85 - 4,000 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 35,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2lds_v5	2	4
Standard_D4lds_v5	4	8
Standard_D8lds_v5	8	16
Standard_D16lds_v5	16	32
Standard_D32lds_v5	32	64
Standard_D48lds_v5	48	96
Standard_D64lds_v5	64	128
Standard_D96lds_v5	96	192

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2

Feature name	Support status
Accelerated Networking	Required
Ephemeral OS Disks	Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dnsv6 sizes series (Preview)

Summarize this article for me

Dnsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all-core turbo clock speed of 3.0 GHz. These virtual machines offer up to 128 vCPU and 512 GiB of RAM. Dnsv6-series virtual machines, an extension of our standard Dsv6-series provides better networking performance for most general-purpose workloads. It also provides increased scalability, upgraded CPU, elevated memory bandwidth, and faster remote storage access compared to the prior generation.

Network Optimized VMs enhance accelerated networking by providing hardware acceleration of initial connection setup for certain traffic types, a task previously performed in software. These enhancements reduce the end-to-end latency for initially establishing a connection or initial packet flow, and allow a VM to scale up the number of connections it manages more quickly, subject to application constraints.

The new Network Optimized sizes make use of enhancements provided by Azure Boost to deliver increased network bandwidth per vCPU, a greater number of vNICs, and improved connection setup performance. The Dnsv6 offers up to 200 Gbps network bandwidth, 15 vNICs and up to 400k Connection per Second (CPS).

These VM sizes are relevant for workloads such as network virtual appliances, large-scale e-commerce applications that require the ability to handle a high number of user connections, and data transfers and media processing tasks that involve transferring large amounts of data quickly. Other applications will also benefit from increased network connection setup performance.

Dnsv6-series virtual machines don't have any temporary storage thus lowering the price of entry. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 512 GiB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	None	
Remote Storage Premium SSD	8 - 64 Disks	3750 - 204800 IOPS 106 - 6782 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4167 - 266667 IOPS 124 - 7935 MBps
Network	4 - 15 NICs	25000 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ns_v6	2	8
Standard_D4ns_v6	4	16
Standard_D8ns_v6	8	32
Standard_D16ns_v6	16	64
Standard_D32ns_v6	32	128
Standard_D48ns_v6	48	192
Standard_D64ns_v6	64	256
Standard_D96ns_v6	96	384
Standard_D128ns_v6	128	512

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dndsv6 sizes series (Preview)

Summarize this article for me

Dndsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all- core turbo clock speed of 3.0 GHz. These virtual machines offer up to 128 vCPU and 512 GiB of RAM. Dndsv6-series virtual machines, an extension of our standard Ddsv6-series, provide better networking performance for most general-purpose workloads. It also provides increased scalability, upgraded CPU, elevated memory bandwidth, and faster remote storage access compared to the prior generation.

Network Optimized VMs enhance accelerated networking by providing hardware acceleration of initial connection setup for certain traffic types, a task previously performed in software. These enhancements reduce the end-to-end latency for initially establishing a connection or initial packet flow and allow a VM to scale up the number of connections it manages more quickly, subject to application constraints.

The new Network Optimized sizes make use of enhancements provided by Azure Boost to deliver increased network bandwidth per vCPU, a greater number of vNICs, and improved connection setup performance. The Dndsv6 offers up to 200 Gbps network bandwidth, 15 vNICs and up to 400k Connection per Second (CPS).

These VM sizes are relevant for workloads such as network virtual appliances, large-scale e-commerce applications that require the ability to handle a high number of user connections, and data transfers and media processing tasks that involve transferring large amounts of data quickly. Other applications will also benefit from increased network connection setup performance.

Dndsv6-series virtual machines offer local storage disks. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 512 GiB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	1 - 4 Disks	110 - 1760 GiB 37500 - 2400000 (RR) 180-11520 MBps (RR)
Remote Storage Premium SSD	8 - 64 Disks	3750 - 204800 IOPS 106 - 6782 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4167 - 266667 IOPS 124 - 7935 MBps
Network	4 - 15 NICs	25000 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2nds_v6	2	8
Standard_D4nds_v6	4	16
Standard_D8nds_v6	8	32
Standard_D16nds_v6	16	64
Standard_D32nds_v6	32	128
Standard_D48nds_v6	48	192
Standard_D64nds_v6	64	256
Standard_D96nds_v6	96	384
Standard_D128nds_v6	128	512

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dnlsv6 sizes series (Preview)

Summarize this article for me

Dnlsv6-series virtual machines run on 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) CPU processor reaching an all- core turbo clock speed of up to 3.0 GHz. These virtual machines offer up to 128 vCPU and 256 GiB of RAM. These VM sizes can reduce costs when running non-memory intensive applications. Dnlsv6-series virtual machines, an extension of our standard Dlsv6-series, provide better networking performance for most general-purpose workloads.

Network Optimized VMs enhance accelerated networking by providing hardware acceleration of initial connection setup for certain traffic types, a task previously performed in software. These enhancements reduce the end-to-end latency for initially establishing a connection or initial packet flow, and allow a VM to scale up the number of connections it manages more quickly, subject to application constraints.

The new Network Optimized sizes make use of enhancements provided by Azure Boost to deliver increased network bandwidth per vCPU, a greater number of vNICs, and improved connection setup performance. The Dnlsv6 offers up to 200 Gbps network bandwidth, 15 vNICs and up to 400k Connection per Second (CPS).

These VM sizes are relevant for workloads such as network virtual appliances, large-scale e-commerce applications that require the ability to handle a high number of user connections, and data transfers and media processing tasks that involve transferring large amounts of data quickly. Other applications will also benefit from increased network connection setup performance.

Dnlsv6-series virtual machines don't have any temporary storage, thus lowering the price of entry. You can attach Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 256 GiB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	None	
Remote Storage Premium SSD	8 - 64 Disks	3750 - 204800 IOPS 106 - 6782 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4167 - 266667 IOPS 124 - 7935 MBps
Network	4 - 15 NICs	25000 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2nls_v6	2	4
Standard_D4nls_v6	4	8
Standard_D8nls_v6	8	16
Standard_D16nls_v6	16	32
Standard_D32nls_v6	32	64
Standard_D48nls_v6	48	96
Standard_D64nls_v6	64	128
Standard_D96nls_v6	96	192
Standard_D128nls_v6	128	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dnldsv6 sizes series (Preview)

Summarize this article for me

Dnldsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all- core turbo clock speed of up to 3.0 GHz. These virtual machines offer up to 128 vCPU and 256 GiB of RAM and fast, local SSD storage up to 4x1760 GiB. These VM sizes can reduce cost when running non-memory intensive applications. Dnldsv6-series virtual machines, an extension of our standard Dldsv6-series, provide better networking performance for most general-purpose workloads.

Network Optimized VMs enhance accelerated networking by providing hardware acceleration of initial connection setup for certain traffic types, a task previously performed in software. These enhancements reduce the end-to-end latency for initially establishing a connection or initial packet flow, and allow a VM to scale up the number of connections it manages more quickly, subject to application constraints.

The new Network Optimized sizes make use of enhancements provided by Azure Boost to deliver increased network bandwidth per vCPU, a greater number of vNICs, and improved connection setup performance. The Dnldsv6 offers up to 200 Gbps network bandwidth, 15 vNICs and up to 400k Connection per Second (CPS).

These VM sizes are relevant for workloads such as network virtual appliances, large-scale e-commerce applications that require the ability to handle a high number of user connections, and data transfers and media processing tasks that involve transferring large amounts of data quickly. Other applications will also benefit from increased network connection setup performance.

Dnldsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 256 GiB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	1 - 4 Disks	110 - 1760 GiB 37500 - 2400000 IOPS (RR) 180-11520 MBps (RR)
Remote Storage Premium SSD	8 - 64 Disks	3750 - 204800 IOPS 106 - 6782 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4167 - 266667 IOPS 124 - 7935 MBps
Network	4 - 15 NICs	25000 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2nlds_v6	2	4
Standard_D4nlds_v6	4	8
Standard_D8nlds_v6	8	16
Standard_D16nlds_v6	16	32
Standard_D32nlds_v6	32	64
Standard_D48nlds_v6	48	96
Standard_D64nlds_v6	64	128
Standard_D96nlds_v6	96	192
Standard_D128nlds_v6	128	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Dv4 sizes series

Summarize this article for me

The Dv4 and Dsv4-series run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), or Intel® Xeon® Platinum 8272CL (Cascade Lake) processors in a hyper-threaded configuration, providing a better value proposition for most general-purpose workloads. It features an all core Turbo clock speed of 3.4 GHz, Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and Intel® Advanced Vector Extensions 512 (Intel® AVX-512). They also support Intel® Deep Learning Boost.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD
Network	2 - 8 NICs	5,000 - 30,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2_v4	2	8
Standard_D4_v4	4	16
Standard_D8_v4	8	32
Standard_D16_v4	16	64
Standard_D32_v4	32	128
Standard_D48_v4	48	192
Standard_D64_v4	64	256

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dsv4 sizes series

Summarize this article for me

The Dv4 and Dsv4-series run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), or Intel® Xeon® Platinum 8272CL (Cascade Lake) processors in a hyper-threaded configuration, providing a better value proposition for most general-purpose workloads. It features an all core Turbo clock speed of 3.4 GHz, Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and Intel® Advanced Vector Extensions 512 (Intel® AVX-512). They also support Intel® Deep Learning Boost.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	5,000 - 30,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2s_v4	2	8
Standard_D4s_v4	4	16
Standard_D8s_v4	8	32
Standard_D16s_v4	16	64
Standard_D32s_v4	32	128
Standard_D48s_v4	48	192
Standard_D64s_v4	64	256

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dav4 sizes series

Summarize this article for me

The Dav4-series run on 2nd Generation AMD EPYC™ 7452 (up to 3.35GHz) or 3rd Generation EPYC™ 7763v processors (up to 3.5GHz). The Dav4-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads. Data disk storage is billed separately from virtual machines. To use Premium SSD, use the Dasv4 sizes. The pricing and billing meters for Dasv4 sizes are the same as the Dav4-series.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7452 (Rome) [x86-64] AMD EPYC 7763v (Milan) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	50 - 2,400 GiB 3,000 - 96,000 IOPS 46 - 1,000 MBps
Remote Storage	4 - 32 Disks	3,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD, Premium SSD, Ultra Disk
Network	2 - 8 NICs	2,000 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2a_v4	2	8
Standard_D4a_v4	4	16
Standard_D8a_v4	8	32
Standard_D16a_v4	16	64
Standard_D32a_v4	32	128
Standard_D48a_v4	48	192
Standard_D64a_v4	64	256
Standard_D96a_v4	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dasv4 sizes series

Summarize this article for me

The Dasv4-series run on 2nd Generation AMD EPYC™ 7452 (up to 3.35GHz) or 3rd Generation EPYC™ 7763v processors (up to 3.5GHz) and use Premium SSD. The Dasv4-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7452 (Rome) [x86-64] AMD EPYC 7763v (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	4 - 32 Disks	16 - 768 GiB 4,000 - 192,000 IOPS 32 - 1,020 MBps
Remote Storage	4 - 32 Disks	3,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD, Premium SSD, Ultra Disk
Network	2 - 8 NICs	2,000 - 40,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2as_v4	2	8
Standard_D4as_v4	4	16
Standard_D8as_v4	8	32
Standard_D16as_v4	16	64
Standard_D32as_v4	32	128
Standard_D48as_v4	48	192
Standard_D64as_v4	64	256
Standard_D96as_v4	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Ddv4 sizes series

Summarize this article for me

The Ddv4 and Ddsv4-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) or the Intel® Xeon® Platinum 8272CL (Cascade Lake) processors in a hyper-threaded configuration, providing a better value proposition for most general-purpose workloads. It features an all core Turbo clock speed of 3.4 GHz, Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and Intel® Advanced Vector Extensions 512 (Intel® AVX-512). They also support Intel® Deep Learning Boost. These new VM sizes will have 50% larger local storage, as well as better local disk IOPS for both read and write compared to the Dv3/Dsv3 sizes with Gen2 VMs. D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	1 Disk	75 - 2,400 GiB 9,000 - 300,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	3,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	5,000 - 30,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2d_v4	2	8
Standard_D4d_v4	4	16
Standard_D8d_v4	8	32
Standard_D16d_v4	16	64
Standard_D32d_v4	32	128
Standard_D48d_v4	48	192
Standard_D64d_v4	64	256

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disks	Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Ddsv4 sizes series

Summarize this article for me

The Ddv4 and Ddsv4-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) or the Intel® Xeon® Platinum 8272CL (Cascade Lake) processors in a hyper-threaded configuration, providing a better value proposition for most general-purpose workloads. It features an all core Turbo clock speed of 3.4 GHz, Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology and Intel® Advanced Vector Extensions 512 (Intel® AVX-512). They also support Intel® Deep Learning Boost. These new VM sizes will have 50% larger local storage, as well as better local disk IOPS for both read and write compared to the Dv3/Dsv3 sizes with Gen2 VMs. D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	1 Disk	75 - 2,400 GiB 9,000 - 300,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	3,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD, Premium SSD, Ultra Disk
Network	2 - 8 NICs	5,000 - 30,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2ds_v4	2	8
Standard_D4ds_v4	4	16
Standard_D8ds_v4	8	32
Standard_D16ds_v4	16	64
Standard_D32ds_v4	32	128
Standard_D48ds_v4	48	192
Standard_D64ds_v4	64	256

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disks	Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

'DC' family general purpose VM size series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'DC' series family are one of Azure's security focused general purpose VM instances. They're designed for confidential computing offering enhanced data protection and integrity with various hardware-based Trusted Execution Environments (TEEs). These VMs work well for many general computing workloads, e-commerce systems, web front ends, desktop virtualization solutions, sensitive databases, other enterprise applications and more.

Workloads and use cases

Data Protection: DC-series VMs are ideal for applications that manage, store, and process sensitive data, such as personal identifiable information (PII), financial data, health records, and other types of confidential information. The hardware-based encryption ensures that data is protected at rest and during processing.

Regulatory Compliance: For businesses that need to comply with stringent regulatory requirements for data privacy and security (like GDPR, HIPAA, or financial industry regulations), DC-series VMs provide a hardware-assured environment that can help meet these compliance demands.

Series in family

DCasv6 and DCadsv6-series

DCasv6-series

Azure's DCasv6-series virtual machines provide robust confidential computing capabilities, ensuring code and data remain secure during processing. Companies can migrate sensitive workloads to cloud infrastructure without modifying existing applications. The series runs on AMD's fourth-generation EPYC™ processors configured for multi-threaded operations with enhanced L3 cache architecture.

AMD's SEV-SNP technology creates hardware-isolated environments that prevent unauthorized access from hypervisors, host management systems, and administrative users. This isolation protects virtual machines from neighboring VMs, hypervisor vulnerabilities, and both hardware and software-based security threats.

The platform includes hardware-based VM memory encryption alongside native support for confidential disk encryption feature. Organizations can implement OS disk encryption at boot through customer-managed keys (CMK) or platform-managed keys (PMK). This feature is fully integrated with Azure KeyVault and Azure Managed HSM ensuring FIPS 140-2 Level 3 compliance.

DCasv6 instances provide balanced memory-to-vCPU ratios suitable for production environments. Configurations scale up to 96 vCPUs paired with 384 GiB RAM and remote storage connectivity. These specifications support diverse workloads including e-commerce platforms, web services, desktop virtualization, secure databases, enterprise applications and more.

View the full DCasv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 34,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

DCasv5 and DCadsv5-series

DCasv5-series

DCasv5-series VMs offer a combination of vCPU and memory for most production workloads. These confidential VMs use AMD's third-Generation EPYC™ 7763v processor in a multi-threaded configuration with up to 256 MB L3 cache. These processors can achieve a boosted maximum frequency of 3.5 GHz. Both series offer Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP). SEV-SNP provides hardware-isolated VMs that protect data from other VMs, the hypervisor, and host management code.

Confidential VMs offer hardware-based VM memory encryption. These series also offer OS disk pre-encryption before VM provisioning with different key management solutions. These VMs with no local disk provide a better value proposition for workloads where you don't need a local temporary disk. For more information, see the FAQ for Azure VM sizes with no local temporary disk.

View the full DCasv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC (Milan) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 82 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	3,000 - 20,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

DCesv6-series

The DCesv6-series are Azure confidential VMs that protect the confidentiality and integrity of code and data while it's being processed. Organizations can use these VMs to seamlessly bring confidential workloads to the cloud without any code changes to the application. These machines are powered by Intel® 5th Generation Xeon® Scalable processors reaching an all-core turbo clock speed of 3.0 GHz and Intel® AMX for AI acceleration.

With the support of Intel® Trust Domain Extensions (TDX), these VMs are hardened from the cloud virtualized environment by denying the hypervisor, other host management code, and administrators access to the VM memory and state. It helps to protect VMs against a broad range of sophisticated hardware and software attacks.

These VMs have native support for confidential disk encryption meaning organizations can encrypt their VM disks at boot with either a customer-managed key (CMK), or platform-managed key (PMK). This feature is fully integrated with Azure Key Vault or Azure Managed HSM with validation for FIPS 140-2 Level 3.

The DCesv6-series offers a balance of memory to vCPU performance. With up to 128 vCPUs, 512 GB of RAM, and support for remote disk storage, this size series works well for many general computing workloads, e-commerce systems, web front ends, desktop virtualization solutions, sensitive databases, other enterprise applications.

View the full DCesv6-series page.

DCas_cc_v5 and DCads_cc_v5-series

DCas_cc_v5-series

Confidential child capable VMs allow you to borrow resources from the parent VM you deploy, to create AMD SEV-SNP protected child VMs. The parent VM has almost complete feature parity with any other general purpose Azure VM (for example, D-series VMs). This parent-child deployment model can help you achieve higher levels of isolation from the Azure host and parent VM. These confidential child capable VMs are built on the same hardware that powers our Azure confidential VMs. Azure confidential VMs are now generally available. The DCas_cc_v5-series sizes offer a combination of vCPU and memory for most production workloads. These new VMs with no local disk provide a better value proposition for workloads that do not require local temp disk.

View the full DCas_cc_v5-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	4 - 96 vCPUs	AMD EPYC (Milan) [x86-64]
Memory	16 - 384 GiB	
Local Storage	None	
Remote Storage	8 - 32 Disks	6,000 - 80,000 IOPS 144 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

DCsv3 and DCdsv3-series

DCsv3-series

The DCsv3-series Azure Virtual Machines help protect the confidentiality and integrity of your code and data while they're being processed in the public cloud. By using Intel® Software Guard Extensions and Intel® Total Memory Encryption - Multi Key, customers can ensure their data is always encrypted and protected in use. These machines are powered by the latest 3rd Generation Intel® Xeon Scalable processors, and use Intel® Turbo Boost Max Technology 3.0 to reach 3.5 GHz. With this generation, CPU Cores have increased 6x (up to a maximum of 48 physical cores). Encrypted Memory (EPC) has increased 1500x to 256 GB. Regular Memory has increased 12x to 384 GB. All these changes substantially improve the performance and unlock entirely new scenarios.

View the full DCsv3-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	1 - 48 vCPUs	Intel Xeon (Ice Lake) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	
Network	2 - 8 NICs	
Accelerators	None	

DCsv2-series

The DCsv2-series virtual machines help protect the confidentiality and integrity of your data and code while it's processed in the public cloud. DCsv2-series leverage Intel® Software Guard Extensions (SGX), which enable customers to use secure enclaves for protection. These machines are backed by 3.7 GHz Intel® Xeon E-2288G (Coffee Lake) with SGX technology. With Intel® Turbo Boost Max Technology 3.0 these machines can go up to 5.0 GHz. Example confidential use cases include: databases, blockchain, multiparty data analytics, fraud detection, anti-money laundering, usage analytics, intelligence analysis and machine learning.

> **Note**
>
> DCsv2-series virtual machines (VMs) are scheduled for retirement on June 30, 2026.

View the full DCsv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 8 vCPUs	Intel Xeon E-2288G (Coffee Lake) [x86-64]
Memory	4 - 32 GiB	
Local Storage	1 Disk	50 - 400 GiB
Remote Storage	1 - 8 Disks	
Network	1 NICs	Interfaces: NetVSC, ConnectX
Accelerators	None	

Previous-generation DC family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 02/13/2026

DCasv5 sizes series

DCasv5-series VMs offer a combination of vCPU and memory for most production workloads. These confidential VMs use AMD's third-Generation EPYC™ 7763v processor in a multi-threaded configuration with up to 256 MB L3 cache. These processors can achieve a boosted maximum frequency of 3.5 GHz. Both series offer Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP). SEV-SNP provides hardware-isolated VMs that protect data from other VMs, the hypervisor, and host management code. Confidential VMs offer hardware-based VM memory encryption. These series also offer OS disk pre-encryption before VM provisioning with different key management solutions. These VMs with no local disk provide a better value proposition for workloads where you don't need a local temporary disk. For more information, see the FAQ for Azure VM sizes with no local temporary disk.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC (Milan) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 82 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	3,000 - 20,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC2as_v5	2	8
Standard_DC4as_v5	4	16
Standard_DC8as_v5	8	32
Standard_DC16as_v5	16	64
Standard_DC32as_v5	32	128
Standard_DC48as_v5	48	192
Standard_DC64as_v5	64	256
Standard_DC96as_v5	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCadsv5 sizes series

Summarize this article for me

DCadsv5-series VMs offer a combination of vCPU and memory for most production workloads. These confidential VMs use AMD's third-Generation EPYC™ 7763v processor in a multi-threaded configuration with up to 256 MB L3 cache. These processors can achieve a boosted maximum frequency of 3.5 GHz. Both series offer Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP). SEV-SNP provides hardware-isolated VMs that protect data from other VMs, the hypervisor, and host management code. Confidential VMs offer hardware-based VM memory encryption. These series also offer OS disk pre-encryption before VM provisioning with different key management solutions. DCadsv5-series offer a combination of vCPU, memory, and temporary storage for most production workloads.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC (Milan) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	75 - 3,600 GiB 9,000 - 450,000 IOPS 125 - 4,000 MBps
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 82 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	3,000 - 20,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC2ads_v5	2	8
Standard_DC4ads_v5	4	16
Standard_DC8ads_v5	8	32
Standard_DC16ads_v5	16	64
Standard_DC32ads_v5	32	128
Standard_DC48ads_v5	48	192
Standard_DC64ads_v5	64	256
Standard_DC96ads_v5	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCasv6 sizes series

Summarize this article for me

Azure's DCasv6-series virtual machines provide robust confidential computing capabilities, ensuring code and data remain secure during processing. Companies can migrate sensitive workloads to cloud infrastructure without modifying existing applications. The series runs on AMD's fourth-generation EPYC™ processors configured for multi-threaded operations with enhanced L3 cache architecture.

AMD's SEV-SNP technology creates hardware-isolated environments that prevent unauthorized access from hypervisors, host management systems, and administrative users. This isolation protects virtual machines from neighboring VMs, hypervisor vulnerabilities, and both hardware and software-based security threats.

The platform includes hardware-based VM memory encryption alongside native support for confidential disk encryption feature. Organizations can implement OS disk encryption at boot through customer-managed keys (CMK) or platform-managed keys (PMK). This feature is fully integrated with Azure KeyVault and Azure Managed HSM ensuring FIPS 140-2 Level 3 compliance.

DCasv6 instances provide balanced memory-to-vCPU ratios suitable for production environments. Configurations scale up to 96 vCPUs paired with 384 GiB RAM and remote storage connectivity. These specifications support diverse workloads including e-commerce platforms, web services, desktop virtualization, secure databases, enterprise applications and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - 8 NICs	12,500 - 34,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC2as_v6	2	8
Standard_DC4as_v6	4	16
Standard_DC8as_v6	8	32
Standard_DC16as_v6	16	64
Standard_DC32as_v6	32	128
Standard_DC48as_v6	48	192
Standard_DC64as_v6	64	256
Standard_DC96as_v6	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCadsv6 sizes series

Summarize this article for me

Azure's DCadsv6-series virtual machines provide robust confidential computing capabilities, ensuring code and data remain secure during processing. Companies can migrate sensitive workloads to cloud infrastructure without modifying existing applications. The series runs on AMD's fourth-generation EPYC™ processors configured for multi-threaded operations with enhanced L3 cache architecture.

AMD's SEV-SNP technology creates hardware-isolated environments that prevent unauthorized access from hypervisors, host management systems, and administrative users. This isolation protects virtual machines from neighboring VMs, hypervisor vulnerabilities, and both hardware and software-based security threats.

The platform includes hardware-based VM memory encryption alongside native support for confidential disk encryption feature. Organizations can implement OS disk encryption at boot through customer-managed keys (CMK) or platform-managed keys (PMK). This feature is fully integrated with Azure KeyVault and Azure Managed HSM ensuring FIPS 140-2 Level 3 compliance.

DCadsv6 instances provide balanced memory-to-vCPU ratios suitable for production environments. Configurations scale up to 96 vCPUs paired with 384 GiB RAM and remote storage connectivity. These specifications support diverse workloads including e-commerce platforms, web services, desktop virtualization, secure databases, enterprise applications and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC™ 9004 (Genoa) [x86-64]
Memory	8 - 384 GiB	
Local Storage		75 - 3,600 GiB 9,000 - 450,000 IOPS 125 - 4,000 MBps
Remote Storage	8 - 64 Disks	4,000 - 175,000 IOPS 90 - 4,320 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
		Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2
Network	2 - 8 NICs	12,500 - 34,000 Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC2ads_v6	2	8
Standard_DC4ads_v6	4	16
Standard_DC8ads_v6	8	32
Standard_DC16ads_v6	16	64
Standard_DC32ads_v6	32	128
Standard_DC48ads_v6	48	192
Standard_DC64ads_v6	64	256
Standard_DC96ads_v6	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCesv6-series sizes

The DCesv6-series are Azure confidential VMs that protect the confidentiality and integrity of code and data while it's being processed. Organizations can use these VMs to seamlessly bring confidential workloads to the cloud without any code changes to the application. These machines are powered by Intel® 5th Generation Xeon® Scalable processors reaching an all-core turbo clock speed of 3.0 GHz and Intel® AMX for AI acceleration.

With the support of Intel® Trust Domain Extensions (TDX), these VMs are hardened from the cloud virtualized environment by denying the hypervisor, other host management code, and administrators access to the VM memory and state. It helps to protect VMs against a broad range of sophisticated hardware and software attacks.

These VMs have native support for confidential disk encryption meaning organizations can encrypt their VM disks at boot with either a customer-managed key (CMK), or platform-managed key (PMK). This feature is fully integrated with Azure Key Vault or Azure Managed HSM with validation for FIPS 140-2 Level 3.

The DCesv6-series offers a balance of memory to vCPU performance. With up to 128 vCPUs, 512 GB of RAM, and support for remote disk storage, this size series works well for many general computing workloads, e-commerce systems, web front ends, desktop virtualization solutions, sensitive databases, other enterprise applications.

Host specifications

Expand table

Part	Quantity min - max	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 512 GB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4,000 - 205,000 IOPS 80 - 4,000 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	12,500 Mbps to 40,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

This table shows the number of vCPUs and amount of memory for each DCesv6-series size.

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC2es_v6	2	8
Standard_DC4es_v6	4	16
Standard_DC8es_v6	8	32
Standard_DC16es_v6	16	64
Standard_DC32es_v6	32	128
Standard_DC48es_v6	48	192
Standard_DC64es_v6	64	256
Standard_DC96es_v6	96	384
Standard_DC128es_v6	128	512

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported

Feature name	Support status
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCedsv6-series sizes

Summarize this article for me

The DCedsv6-series are Azure confidential VMs that protect the confidentiality and integrity of code and data while it's being processed. Organizations can use these VMs to seamlessly bring confidential workloads to the cloud without any code changes to the application. These machines are powered by Intel® 5th Generation Xeon® Scalable processors reaching an all-core turbo clock speed of 3.0 GHz and Intel® AMX for AI acceleration.

Featuring Intel® Trust Domain Extensions (TDX), these VMs are hardened from the cloud virtualized environment by denying the hypervisor, other host management code, and administrators access to the VM memory and state. It helps to protect VMs against a broad range of sophisticated hardware and software attacks.

These VMs have native support for confidential disk encryption meaning organizations can encrypt their VM disks at boot with either a customer-managed key (CMK), or platform-managed key (PMK). This feature is fully integrated with Azure Key Vault or Azure Managed HSM with validation for FIPS 140-2 Level 3.

The DCedsv6 offer a balance of memory to vCPU performance that's suitable most production workloads. With up to 128 vCPUs, 512 GiB of RAM, and support for up to 7 TB of local disk storage. These VMs work well for many general computing workloads, e-commerce systems, web front ends, desktop virtualization solutions, sensitive databases, other enterprise applications and more.

Host specifications

Expand table

Part	Quantity min - max	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8,573C (Emerald Rapids) [x86-64]
Memory	8 - 512 GB	
Local Storage	110 - 7,040 GiB	
Remote Storage	8 - 64 Disks	3,750 to 204,800 IOPS and 80 to 4,000 MBps
Network	2 - 8 NICs	12,500 to 40,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

This table shows the number of vCPUs and amount of memory for each DCesv6-series size.

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC2eds_v6	2	8
Standard_DC4eds_v6	4	16
Standard_DC8eds_v6	8	32
Standard_DC16eds_v6	16	64
Standard_DC32eds_v6	32	128
Standard_DC48eds_v6	48	192
Standard_DC64eds_v6	64	256
Standard_DC96eds_v6	96	384
Standard_DC128eds_v6	128	512

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported

Feature name	Support status
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCas_cc_v5 sizes series

Summarize this article for me

> **Note**
>
> This VM series is currently in **Preview**. See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Confidential child capable VMs allow you to borrow resources from the parent VM you deploy, to create AMD SEV-SNP protected child VMs. The parent VM has almost complete feature parity with any other general purpose Azure VM (for example, D-series VMs). This parent-child deployment model can help you achieve higher levels of isolation from the Azure host and parent VM. These confidential child capable VMs are built on the same hardware that powers our Azure confidential VMs. Azure confidential VMs are now generally available. The DCas_cc_v5-series sizes offer a combination of vCPU and memory for most production workloads. These new VMs with no local disk provide a better value proposition for workloads that do not require local temp disk.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 96 vCPUs	AMD EPYC (Milan) [x86-64]
Memory	16 - 384 GiB	
Local Storage	None	
Remote Storage	8 - 32 Disks	6,000 - 80,000 IOPS 144 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC4as_cc_v5	4	16
Standard_DC8as_cc_v5	8	32
Standard_DC16as_cc_v5	16	64
Standard_DC32as_cc_v5	32	128
Standard_DC48as_cc_v5	48	192
Standard_DC64as_cc_v5	64	256
Standard_DC96as_cc_v5	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Trusted Launch	Not Supported
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCads_cc_v5 sizes series

Summarize this article for me

> **Note**
>
> This VM series is currently in **Preview**. See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Confidential child capable VMs allow you to borrow resources from the parent VM you deploy, to create AMD SEV-SNP protected child VMs. The parent VM has almost complete feature parity with any other general purpose Azure VM (for example, D-series VMs). This parent-child deployment model can help you achieve higher levels of isolation from the Azure host and parent VM. These confidential child capable VMs are built on the same hardware that powers our Azure confidential VMs. Azure confidential VMs are now generally available. The DCads_cc_v5-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 96 vCPUs	AMD EPYC (Milan) [x86-64]
Memory	16 - 384 GiB	
Local Storage	1 Disk	150 - 3,600 GiB
Remote Storage	8 - 32 Disks	6,000 - 80,000 IOPS 144 - 1,600 MBps Disk Types: Standard SDD/HDD, Premium SSD
Network	2 - 8 NICs	Mbps Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DC4ads_cc_v5	4	16
Standard_DC8ads_cc_v5	8	32
Standard_DC16ads_cc_v5	16	64
Standard_DC32ads_cc_v5	32	128
Standard_DC48ads_cc_v5	48	192
Standard_DC64ads_cc_v5	64	256
Standard_DC96ads_cc_v5	96	384

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Trusted Launch	Not Supported
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

DCsv3 sizes series

The DCsv3-series Azure Virtual Machines help protect the confidentiality and integrity of your code and data while they're being processed in the public cloud. By using Intel® Software Guard Extensions and Intel® Total Memory Encryption - Multi Key, customers can ensure their data is always encrypted and protected in use. These machines are powered by the latest 3rd Generation Intel® Xeon Scalable processors, and use Intel® Turbo Boost Max Technology 3.0 to reach 3.5 GHz. With this generation, CPU Cores have increased 6x (up to a maximum of 48 physical cores). Encrypted Memory (EPC) has increased 1500x to 256 GB. Regular Memory has increased 12x to 384 GB. All these changes substantially improve the performance and unlock entirely new scenarios.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 48 vCPUs	Intel Xeon (Ice Lake) [x86-64]
Memory	8 - 384 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	
Network	2 - 8 NICs	
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

Cores (Qty.) and Memory for each size

Expand table

Size Name	Cores (Qty.)	Memory (GB)	EPC Memory (GiB)
Standard_DC1s_v3	1	8	4
Standard_DC2s_v3	2	16	8
Standard_DC4s_v3	4	32	16
Standard_DC8s_v3	8	64	32
Standard_DC16s_v3	16	128	64
Standard_DC24s_v3	24	192	128
Standard_DC32s_v3	32	256	192
Standard_DC48s_v3	48	384	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Not Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

DCdsv3 sizes series

The DCdsv3-series Azure Virtual Machines help protect the confidentiality and integrity of your code and data while they're being processed in the public cloud. By using Intel® Software Guard Extensions and Intel® Total Memory Encryption - Multi Key, customers can ensure their data is always encrypted and protected in use. These machines are powered by the latest 3rd Generation Intel® Xeon Scalable processors, and use Intel® Turbo Boost Max Technology 3.0 to reach 3.5 GHz. With this generation, CPU Cores have increased 6x (up to a maximum of 48 physical cores). Encrypted Memory (EPC) has increased 1500x to 256 GB. Regular Memory has increased 12x to 384 GB. All these changes substantially improve the performance and unlock entirely new scenarios.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 48 vCPUs	Intel Xeon (Ice Lake) [x86-64]
Memory	8 - 384 GiB	
Local Storage	1 Disk	75 - 2,400 GiB
Remote Storage	4 - 32 Disks	
Network	2 - 8 NICs	Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

Cores (Qty.) and Memory for each size

Expand table

Size Name	Cores (Qty.)	Memory (GB)	EPC Memory (GiB)
Standard_DC1ds_v3	1	8	4
Standard_DC2ds_v3	2	16	8
Standard_DC4ds_v3	4	32	16
Standard_DC8ds_v3	8	64	32
Standard_DC16ds_v3	16	128	64
Standard_DC24ds_v3	24	192	128
Standard_DC32ds_v3	32	256	192
Standard_DC48ds_v3	48	384	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Not Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

DCsv2 sizes series

The DCsv2-series virtual machines help protect the confidentiality and integrity of your data and code while it's processed in the public cloud. DCsv2-series leverage Intel® Software Guard Extensions (SGX), which enable customers to use secure enclaves for protection. These machines are backed by 3.7 GHz Intel® Xeon E-2288G (Coffee Lake) with SGX technology. With Intel® Turbo Boost Max Technology 3.0 these machines can go up to 5.0 GHz. Example confidential use cases include: databases, blockchain, multiparty data analytics, fraud detection, anti-money laundering, usage analytics, intelligence analysis and machine learning.

> **Note**
>
> DCsv2-series virtual machines (VMs) are scheduled for retirement on June 30, 2026.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 8 vCPUs	Intel Xeon E-2288G (Coffee Lake) [x86-64]
Memory	4 - 32 GiB	
Local Storage	1 Disk	50 - 400 GiB
Remote Storage	1 - 8 Disks	
Network	1 NICs	Interfaces: NetVSC, ConnectX
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

Cores (Qty.) and Memory for each size

Expand table

Size Name	Cores (Qty.)	Memory (GB)	EPC Memory (MiB)
Standard_DC1s_v2	1	4	28
Standard_DC2s_v2	2	8	56
Standard_DC4s_v2	4	16	112
Standard_DC8_v2	8	32	168

VM Basics resources

- Check Core quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

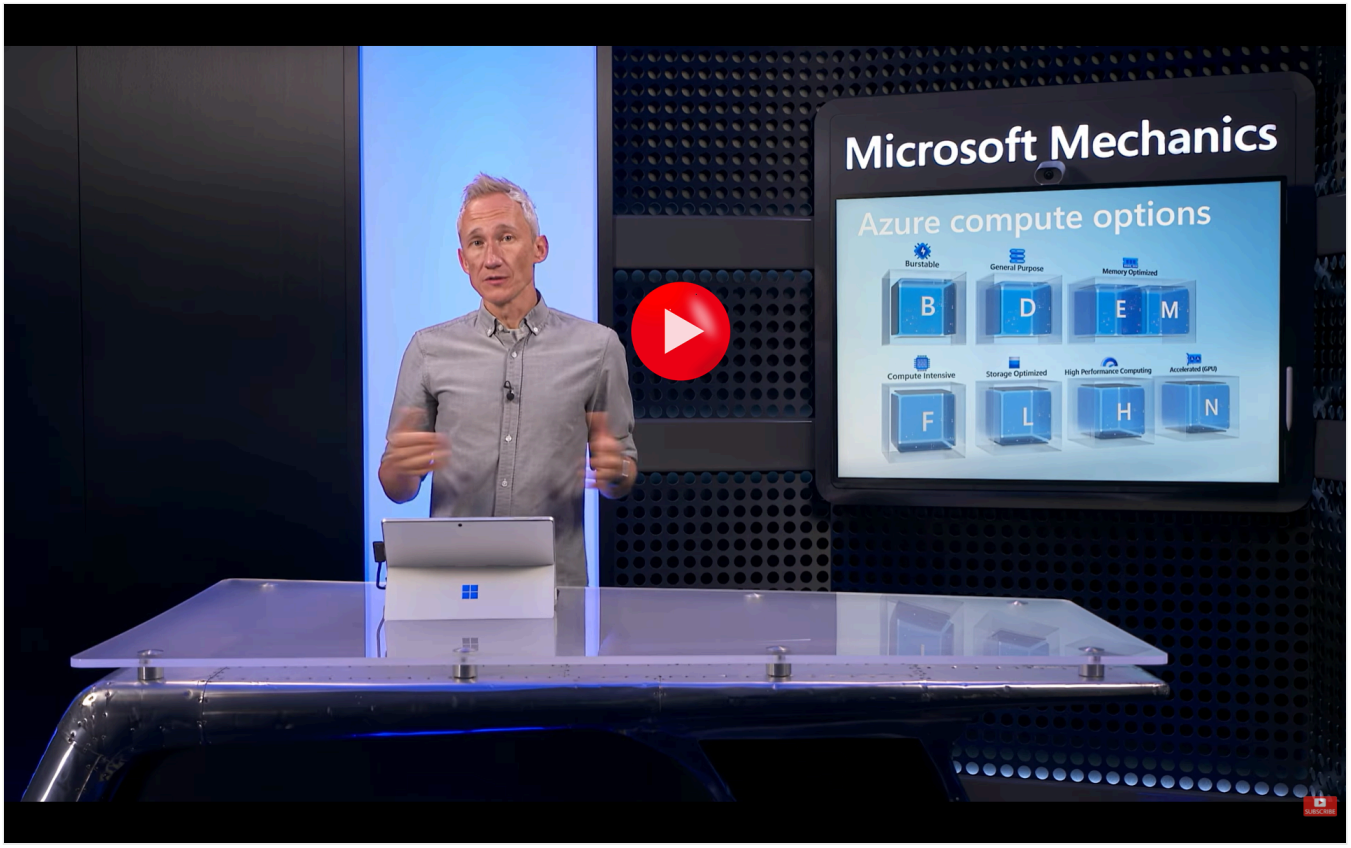
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

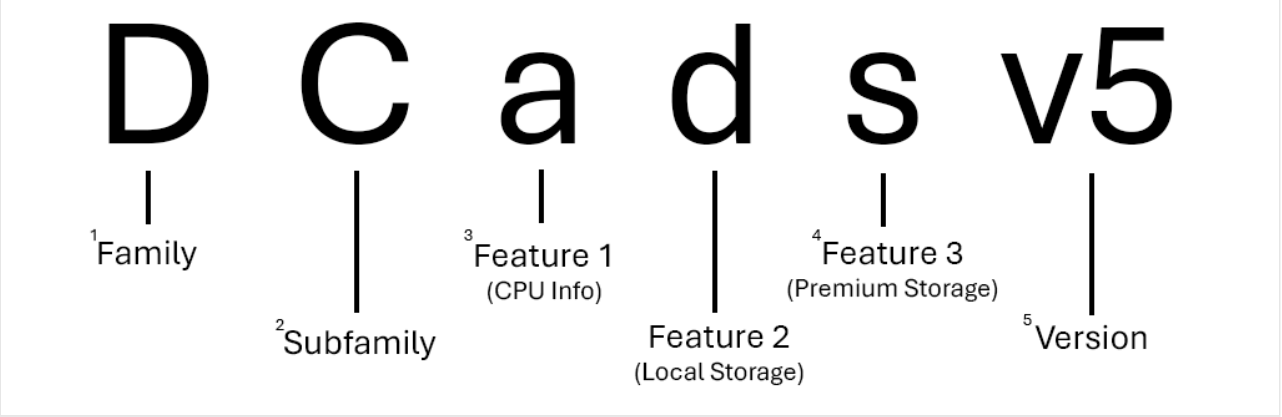
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

'F' family compute optimized VM size series

09/09/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'F' family of VM size series are one of Azure's compute-optimized VM instances. They're designed for workloads that require high CPU performance, such as batch processing, web servers, analytics, and gaming. Featuring a high CPU-to-memory ratio, F-series VMs are equipped with powerful processors to handle applications that demand more CPU capacity relative to memory. This makes them particularly effective for scenarios where fast and efficient processing is critical, allowing businesses to run their compute-bound applications efficiently and cost-effectively.

Workloads and use cases

Web Servers: F-series VMs are excellent for hosting web servers and applications that require significant compute capability to handle web traffic efficiently without necessarily needing large amounts of memory.

Batch Processing: F-series VMs are ideal for batch jobs and other processing tasks that involve handling large volumes of data or tasks in a queue but are more CPU-intensive than memory-intensive.

Application Servers: Applications that require quick processing and do not have high memory demands can benefit from F-series VMs. These can include medium traffic application servers, back-end servers for enterprise applications, and other similar tasks. Gaming Servers: Due to their high CPU performance, F-series VMs are also suitable for gaming servers where fast processing is critical for a good gaming experience.

Analytics: F-series VMs can be used for data analytics applications that require processing speed to crunch numbers and perform calculations more than they require a large amount of memory.

Series in family

Fasv7 series and Fadsv7 series

Fasv7 series

The Fasv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Fasv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Fasv7-series offers up to 80 full core and 320 GiB of RAM. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

View the full Fasv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	4 - 320 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4000 - 310000 IOPS 118 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

Famsv7 series and Famdsv7 series

Famsv7 series

The Famsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Famsv7-series VM comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Famsv7-series offers up to 80 full core vCPUs and 640 GiB of RAM. These VMs offer a combination of core and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage. This series is optimized for scientific simulations, financial and risk

analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

View the full Famsv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	8 - 640 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4000 - 310000 IOPS 118 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

Falsv7 series and Faldsv7 series

Falsv7 series

The Falsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Falsv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Falsv7-series offers up to 80 full core and 160 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

View the full Falsv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	2 - 160 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4000 - 310000 IOPS 118 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

Fasv6, Falsv6, and Famsv6-series

Fasv6-series

The Fasv6-series utilizes AMD's fourth Generation EPYC™ 9004 processor that can achieve a boosted maximum frequency of 3.7 GHz with up to 320 MB L3 cache. The Fasv6 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Fasv6-series offers up to 64 full core vCPUs and 256 GiB of RAM. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance. Customers running software licensed on per-vCPU basis can use these VMs to optimize compute costs within their infrastructure.

View the full Fasv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 256 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4000 - 115200 IOPS

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
		90 - 2880 MBps
Network	2 - 8 NICs	12500 - 36000 Mbps
Accelerators	None	

Fsv2-series

The Fsv2-series run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), the Intel® Xeon® Platinum 8272CL (Cascade Lake) processors, or the Intel® Xeon® Platinum 8168 (Skylake) processors. It features a sustained all core Turbo clock speed of 3.4 GHz and a maximum single-core turbo frequency of 3.7 GHz. Intel® AVX-512 instructions are new on Intel Scalable Processors. These instructions provide up to a 2X performance boost to vector processing workloads on both single and double precision floating point operations. In other words, they're really fast for any computational workload. Fsv2-series VMs feature Intel® Hyper-Threading Technology.

View the full Fsv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 72 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8168 (Skylake) [x86-64]
Memory	4 - 144 GiB	
Local Storage	1 Disk	16 - 576 GiB 4000 - 144000 IOPS (RR) 31 - 1152 MBps (RR)
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 47 - 1100 MBps
Network	2 - 8 NICs	5000 - 30000 Mbps
Accelerators	None	

Previous-generation F family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Fasv7 sizes series

The Fasv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Fasv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Fasv7-series offers up to 80 full core and 320 GiB of RAM. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	4 - 320 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F1as_v7	1	4
Standard_F2as_v7	2	8
Standard_F4as_v7	4	16
Standard_F8as_v7	8	32
Standard_F16as_v7	16	64
Standard_F32as_v7	32	128
Standard_F48as_v7	48	192
Standard_F64as_v7	64	256
Standard_F80as_v7	80	320

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. **NVMe** support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Fadsv7 sizes series

Summarize this article for me

The Fadsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Fadsv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Fadsv7-series offers up to 80 full core and 320 GiB of RAM. The new VMs with local disk are ideal for workloads that require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 cores	AMD EPYC 9005 (Turin) [x86-64]
Memory	4 - 320 GiB	
Local Storage	1 - 6 Disks	110 - 2200 GiB 37500 - 3000000 IOPS (RR) 280 - 22400 MBps (RR)
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F1ads_v7	1	4
Standard_F2ads_v7	2	8
Standard_F4ads_v7	4	16
Standard_F8ads_v7	8	32
Standard_F16ads_v7	16	64
Standard_F32ads_v7	32	128
Standard_F48ads_v7	48	192
Standard_F64ads_v7	64	256
Standard_F80ads_v7	80	320

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 02/26/2026

Famsv7 sizes series

Summarize this article for me

The Famsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Famsv7-series VM comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Famsv7-series offers up to 80 full core vCPUs and 640 GiB of RAM. These VMs offer a combination of core and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	8 - 640 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F1ams_v7	1	8
Standard_F2ams_v7	2	16
Standard_F4ams_v7	4	32
Standard_F8ams_v7	8	64
Standard_F16ams_v7	16	128
Standard_F32ams_v7	32	256
Standard_F48ams_v7	48	384
Standard_F64ams_v7	64	512
Standard_F80ams_v7	80	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Famdsv7 sizes series

Summarize this article for me

The Famdsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Famdsv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Famdsv7-series offers up to 80 full core and 640 GiB of RAM. These VMs offer a combination of core and memory that is ideal for memory-intensive enterprise applications. The new VMs with local disk are ideal for workloads that require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of exceptional performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 cores	AMD EPYC 9005 (Turin) [x86-64]
Memory	8 - 640 GiB	
Local Storage	1 - 6 Disks	110 - 2200 GiB 37500 - 3000000 IOPS (RR) 280 - 22400 MBps (RR)
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F1amds_v7	1	8
Standard_F2amds_v7	2	16
Standard_F4amds_v7	4	32
Standard_F8amds_v7	8	64
Standard_F16amds_v7	16	128
Standard_F32amds_v7	32	256
Standard_F48amds_v7	48	384
Standard_F64amds_v7	64	512
Standard_F80amds_v7	80	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Falsv7 sizes series

Summarize this article for me

The Falsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Falsv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Falsv7-series offers up to 80 full core and 160 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with no local disk provide a better value proposition for workloads that do not require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 80 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	2 - 160 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F1als_v7	1	2
Standard_F2als_v7	2	4
Standard_F4als_v7	4	8
Standard_F8als_v7	8	16
Standard_F16als_v7	16	32
Standard_F32als_v7	32	64
Standard_F48als_v7	48	96
Standard_F64als_v7	64	128
Standard_F80als_v7	80	160

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Faldsv7 sizes series

Summarize this article for me

The Faldsv7-series utilizes AMD's 5th Generation EPYC™ 9005 processor that can achieve a boosted maximum frequency of up to 4.5 GHz. The Faldsv7 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Faldsv7-series offers up to 80 full core and 160 GiB of RAM. These VM sizes can reduce cost when running non-memory intensive applications. The new VMs with local disk are ideal for workloads that require local temporary storage. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	1 - 80 cores	AMD EPYC 9005 (Turin) [x86-64]
Memory	2 - 160 GiB	
Local Storage	1 - 6 Disks	110 - 2200 GiB 37500 - 3000000 IOPS (RR) 280 - 22400 MBps (RR)
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F1alds_v7	1	2
Standard_F2alds_v7	2	4
Standard_F4alds_v7	4	8
Standard_F8alds_v7	8	16
Standard_F16alds_v7	16	32
Standard_F32alds_v7	32	64
Standard_F48alds_v7	48	96
Standard_F64alds_v7	64	128
Standard_F80alds_v7	80	160

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Fasv6 sizes series

Summarize this article for me

The Fasv6-series utilizes AMD's fourth Generation EPYC™ 9004 processor that can achieve a boosted maximum frequency of 3.7 GHz with up to 320 MB L3 cache. The Fasv6 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Fasv6-series offers up to 64 full core vCPUs and 256 GiB of RAM. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance. Customers running software licensed on per-vCPU basis can use these VMs to optimize compute costs within their infrastructure.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	8 - 256 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4000 - 115200 IOPS 90 - 2880 MBps
Network	2 - 8 NICs	12500 - 36000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F2as_v6	2	8
Standard_F4as_v6	4	16
Standard_F8as_v6	8	32
Standard_F16as_v6	16	64
Standard_F32as_v6	32	128
Standard_F48as_v6	48	192
Standard_F64as_v6	64	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Note

This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Falsv6 sizes series

Summarize this article for me

The Falsv6-series utilizes AMD's fourth Generation EPYC™ 9004 processor that can achieve a boosted maximum frequency of 3.7 GHz with up to 320 MB L3 cache. The Falsv6 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Falsv6-series offers up to 64 full core vCPUs and 128 GiB of RAM. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance. Customers running software licensed on per-vCPU basis can use these VMs to optimize compute costs within their infrastructure.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	4 - 128 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4000 - 115200 IOPS 90 - 2880 MBps
Network	2 - 8 NICs	12500 - 36,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F2als_v6	2	4
Standard_F4als_v6	4	8
Standard_F8als_v6	8	16
Standard_F16als_v6	16	32
Standard_F32als_v6	32	64
Standard_F48als_v6	48	96
Standard_F64als_v6	64	128

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Note

This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Famsv6 sizes series

Summarize this article for me

The Famsv6-series utilizes AMD's fourth Generation EPYC™ 9004 processor that can achieve a boosted maximum frequency of 3.7 GHz with up to 320 MB L3 cache. The Famsv6 VM series comes without Simultaneous Multithreading (SMT), meaning a vCPU is now mapped to a full physical core, allowing software processes to run on dedicated and uncontested resources. These new full core VMs suit workloads demanding the highest CPU performance. Famsv6-series offers up to 64 full core vCPUs and 512 GiB of RAM. This series is optimized for scientific simulations, financial and risk analysis, gaming, rendering and other workloads able to take advantage of the exceptional performance. Customers running software licensed on per-vCPU basis can use these VMs to optimize compute costs within their infrastructure.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 512 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	4000 - 115200 IOPS 90 - 2880 MBps
Network	2 - 8 NICs	12500 - 36000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F2ams_v6	2	16
Standard_F4ams_v6	4	32
Standard_F8ams_v6	8	64
Standard_F16ams_v6	16	128
Standard_F32ams_v6	32	256
Standard_F48ams_v6	48	384
Standard_F64ams_v6	64	512

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Note

This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Fsv2-series

The Fsv2-series sizes run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake) processors, or the Intel® Xeon® Platinum 8168 (Skylake) processors. Processor availability varies by region; sizes may run on any of the listed processors.

The Fsv2-series features a sustained all core Turbo clock speed of 3.4 GHz and a maximum single-core turbo frequency of 3.7 GHz. Intel® AVX-512 instructions are new on Intel Scalable Processors. These instructions provide up to a 2X performance boost to vector processing workloads on both single and double precision floating point operations. In other words, they're really fast for any computational workload. Fsv2-series VMs feature Intel® Hyper-Threading Technology.

Host specifications for the Fsv2-series

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 72 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8168 (Skylake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	4 - 144 GB	
Local Storage	1 Disk	16 - 576 GiB 4000 to 144000 IOPS (RR) and 31 to 1152 MBps (RR)
Remote Storage	4 - 32 Disks	3200 to 80000 IOPS and 47 to 1100 MBps
Network	2 - 8 NICs	5000 to 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in the Fsv2-series

Basics

vCPUs (Qty.) and Memory for each size in the Fsv2-series.

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_F2s_v2	2	4
Standard_F4s_v2	4	8
Standard_F8s_v2	8	16
Standard_F16s_v2	16	32
Standard_F32s_v2	32	64
Standard_F48s_v2	48	96
Standard_F64s_v2	64	128
Standard_F72s_v2	72	144

Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

'FX' family general purpose VM size series

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'FX' family of VM size series are one of Azure's specialized compute-optimized VM instances, designed primarily workloads that require significant CPU capabilities. These VMs use the latest Intel Ice Lake processors and are optimized for compute-intensive tasks such as financial modeling, scientific simulations, and heavy calculations. With a high frequency and a large cache per core, FX-series VMs provide exceptional computational power, making them ideal for scenarios demanding extensive processing resources and rapid execution of complex operations.

Workloads and use cases

Electronic Design Automation (EDA): FX-series VMs are well-suited for EDA workloads, which require high CPU clock speeds and high memory-to-CPU ratios. These workloads benefit from the high single-core performance and large memory capacity of FX-series VMs.

Batch Processing: FX-series VMs are excellent for high-throughput batch processing jobs, such as those involving large-scale data analysis and transformation, where rapid processing is critical.

Data Analytics: FX-series VMs are suitable for intensive data analytics applications, especially those that require quick iteration and processing of large data sets.

Series in family

FX-series

The FX-series runs on the Intel® Xeon® Gold 6246R (Cascade Lake) processors. It features an all-core-turbo frequency of 4.0 GHz, 21 GB RAM per vCPU, up to 1 TB total RAM, and local temporary storage. The FX-series will benefit workloads that require a high CPU clock speed and high memory to CPU ratio, workloads with high per-core licensing costs, and applications requiring a high single-core performance. A typical use case for FX-series is the Electronic Design Automation (EDA) workload. FX-series VMs

feature Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology, and Intel® Advanced Vector Extensions 512 (Intel® AVX-512).

View the full FX-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 48 vCPUs	Intel Xeon Gold 6246R (Cascade Lake) [x86-64]
Memory	84 - 1008 GiB	
Local Storage	1 Disk	168 - 2016 GiB 40000 - 400000 IOPS (RR) 343 - 3871 MBps (RR)
Remote Storage	8 - 32 Disks	6700 - 80000 IOPS 104 - 1258 MBps
Network	2 - 8 NICs	4000 - 32000 Mbps
Accelerators	None	

Previous-generation FX family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Check out Azure Dedicated Hosts for physical servers able to host one or more virtual machines assigned to one Azure subscription.

Learn how to Monitor Azure virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

FXmsv2 sizes series

Summarize this article for me

The FXmsv2-series Virtual Machine (VM) runs on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor in a hyper threaded configuration. FXv2-series VMs features an all-core-turbo frequency up to 4.0 GHz. These virtual machines offer up to 96 vCPU and up to 1832 GiB of RAM. The FXmsv2-series benefits workloads that are ideal for memory-intensive enterprise applications requiring a high memory to CPU ratio, high performance, high CPU clock speed, higher remote storage IOPs and throughput. Typical use-cases for FX-series are the Electronic Design Automation (EDA) and SQL workloads

FXmsv2-series virtual machines don't support local storage, thus offering a lower price point for entry. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	42 - 1832 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	8,000 - 400,000 IOPS 273 - 10,000 MBps
Network	2 - 8 NICs	12,500 - 70,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_FX2ms_v2	2	42
Standard_FX4ms_v2	4	84
Standard_FX8ms_v2	8	168
Standard_FX12ms_v2	12	252
Standard_FX16ms_v2	16	336
Standard_FX24ms_v2	24	504
Standard_FX32ms_v2	32	672
Standard_FX48ms_v2	48	1008
Standard_FX64ms_v2	64	1344
Standard_FX96ms_v2	96	1832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported
NVMe storage	Supported
Constrained core	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

FXmdsv2 sizes series

Summarize this article for me

The FXmdsv2-series Virtual Machine (VM) runs on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor in a hyper threaded configuration. FXv2-series VMs features an all-core-turbo frequency up to 4.0 GHz. These virtual machines offer up to 96 vCPU, up to 1832 GiB of RAM and up to 6x880 GiB fast local SSD storage. The FXmdsv2-series benefits workloads that are ideal for memory-intensive enterprise applications requiring a high memory to CPU ratio, high performance, high CPU clock speed, higher remote storage IOPs and throughput. Typical use-cases for FX-series are the Electronic Design Automation (EDA) and SQL workloads.

FXmdsv2-series virtual machines support local storage. You can attach Standard SSDs, Standard HDDs, Premium SSDs, and Premium SSD v2 storage to these virtual machines. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	42 - 1832 GiB	
Local Storage	1 - 6 Disk	110 - 880 GiB 37,500 - 1,800,000 IOPS (RR) 180 - 8,640 MBps (RR)
Remote Storage	8 - 64 Disks	8,000 - 400,000 IOPS 273 - 10,000 MBps
Network	2 - 8 NICs	12,500 - 70,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_FX2mds_v2	2	42
Standard_FX4mds_v2	4	84
Standard_FX8mds_v2	8	168
Standard_FX12mds_v2	12	252
Standard_FX16mds_v2	16	336
Standard_FX24mds_v2	24	504
Standard_FX32mds_v2	32	672
Standard_FX48mds_v2	48	1008
Standard_FX64mds_v2	64	1344
Standard_FX96mds_v2	96	1832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported

Feature name	Support status
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported
NVMe storage	Supported
Constrained core	Supported

> **Note**
>
> FXmdsv2-series VMs provision the ephemeral storage in a RAW state, which must be manually formatted and configured before installing SQL Server, when using the ephemeral storage for the `tempdb` database as dictated by best practices. The ephemeral storage resets to RAW when the VM is deallocated or resized, which impacts some automation by preventing SQL Server from starting. For more information, see How to format and initialize temp NVMe disks and the SQL Server support guidance.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

FX sizes series

The FX-series runs on the Intel® Xeon® Gold 6246R (Cascade Lake) processors. It features an all-core-turbo frequency of 4.0 GHz, 21 GB RAM per vCPU, up to 1 TB total RAM, and local temporary storage. The FX-series will benefit workloads that require a high CPU clock speed and high memory to CPU ratio, workloads with high per-core licensing costs, and applications requiring a high single-core performance. A typical use case for FX-series is the Electronic Design Automation (EDA) workload. FX-series VMs feature Intel® Turbo Boost Technology 2.0, Intel® Hyper-Threading Technology, and Intel® Advanced Vector Extensions 512 (Intel® AVX-512).

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 48 vCPUs	Intel Xeon Gold 6246R (Cascade Lake) [x86-64]
Memory	84 - 1008 GiB	
Local Storage	1 Disk	168 - 2016 GiB 40000 - 400000 IOPS (RR) 343 - 3871 MBps (RR)
Remote Storage	8 - 32 Disks	6700 - 80000 IOPS 104 - 1258 MBps
Network	2 - 8 NICs	4000 - 32000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_FX4mds	4	84
Standard_FX12mds	12	252
Standard_FX24mds	24	504
Standard_FX36mds	36	756
Standard_FX48mds	48	1008

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

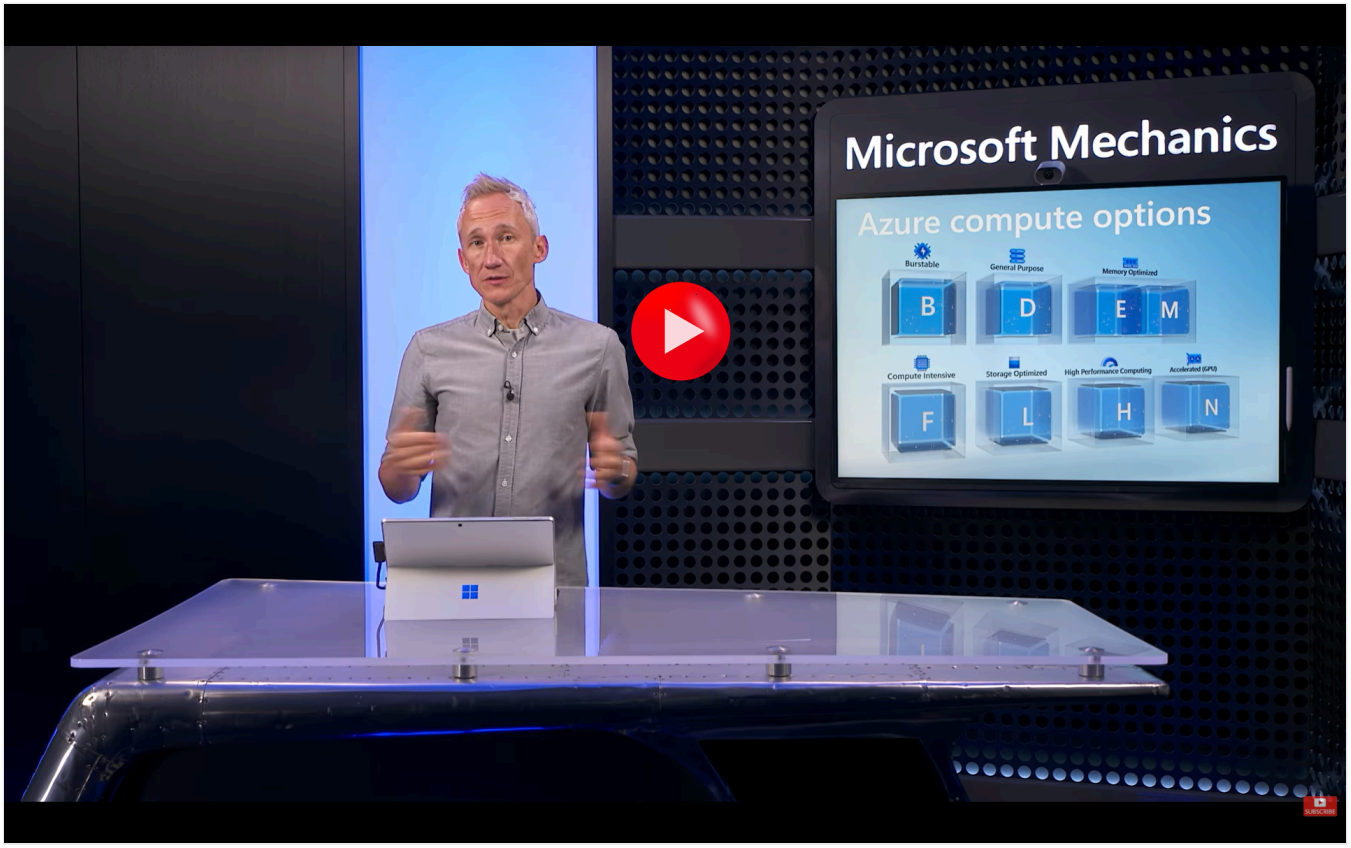
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

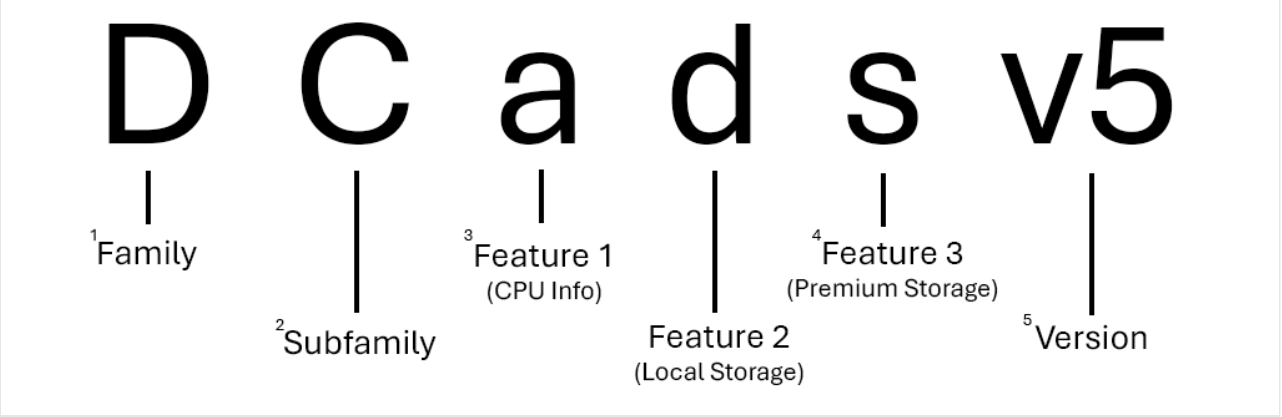
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

'E' family memory optimized VM size series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'E' family of VM size series are one of Azure's memory-optimized VM instances. They're designed for memory-intensive workloads, such as large databases, big data analytics, and enterprise applications that require significant amounts of RAM to maintain high performance. Equipped with high memory-to-core ratios, E-series VMs support applications and services that benefit from faster data access and more efficient data processing capabilities. This makes them particularly well-suited for scenarios involving in-memory databases and extensive data processing tasks where ample memory is crucial for optimal performance.

Workloads and use cases

Memory-Intensive Workloads: E-family VMs are for workloads that demands a large memory footprint to efficiently handle tasks, such as simulations, large-scale computations in scientific research, or financial risk modeling.

Large Databases and SQL Servers: E-family VMs are ideal for hosting large relational databases like SQL Server and NoSQL databases that benefit from high memory capacities for improved performance in data processing and transaction handling.

Enterprise Applications: E-family VMs are suitable for resource-intensive enterprise applications, including large-scale ERP and CRM systems, where the availability of ample memory is crucial for managing complex transactions and user loads.

Big Data Applications: E-family VMs are effective for big data analytics applications that need to process vast amounts of data in memory to speed up analysis and insights generation.

In-Memory Computing: E-family VMs are great for in-memory databases (e.g., SAP HANA) that require large amounts of RAM to keep the entire dataset in memory, allowing for ultra-fast data processing and query responses.

Data Warehousing: E-family VMs provide the necessary resources for data warehousing solutions that handle and analyze large datasets, improving query performance and reducing response times.

Series in family

Esv7 and Edsv7-series

Esv7 series

Esv7-series virtual machines run on the Intel® Xeon® 6973PC (Granite Rapids) processor, reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 372 vCPUs and 2,832 GiB of RAM. Esv7-series virtual machines are ideal for memory-intensive workloads, such as relational database servers, large databases, data analytics engines, in-memory caches, and more.Esv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation, for example increased networking, increased remote storage performance, upgraded CPU, and elevated memory bandwidth.

Esv7-series virtual machines are ideal for memory-intensive workloads, such as SQL and NoSQL database servers, data analytics engines, data warehousing workloads, business intelligence applications, in-memory databases such as SAP and Redis, and in-memory analytics and more.

View the full Esv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 372 vCPUs	Intel@ Xeon® 6 processor 6973P-C(Granite Rapids) [x86-64]
Memory	16 - 2,832 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4,000 - 500,000 IOPS 115 - 16,000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	5,000 - 800,000 IOPS 135 - 20,000 MBps
Network	3 - 15 NICs	16,000 - 400,000 Mbps
Accelerators	None	

Easv7 series and Eadsv7 series

Easv7 series

Easv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 1280 GiB of RAM. The Easv7-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

View the full Easv7-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	16 - 1280 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

Ebsv6 and Ebdsv6-series

Ebsv6-series

The Ebsv6-series virtual machines run on the fifth Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor and deliver higher remote storage performance compared to the previous generation. The Ebsv6-series is ideal for storage throughput intensive workloads. These virtual machines offer up to 800,000 IOPS and 14,000 MBps of remote disk storage performance. Additionally, the Ebsv6 VM series offer up to 192vCPU and 1,832 GiB of RAM providing larger processing power to consolidate workloads. Refer to the Premium SSD, Premium SSD v2, and Ultra disks to attain published disk performance.

View the full Ebsv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1832 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	8800 - 800000 IOPS 253 - 14000 MBps
Network	2 - 8 NICs	12500 - 200000 Mbps
Accelerators	None	

Epsv6 and Epdsv6-series

Epsv6-series

The Epsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for memory optimized workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 8 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs are best suited for memory-intensive scale-out and enterprise workloads, such as relational database servers, large databases, data analytics engines, in-memory caches, and more. Epsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Epsv6-series virtual machines offer up to 96 vCPU and 672 GiB of RAM. These VMs come without local disk, providing a better value proposition for workloads that don't require local temporary storage.

View the full Epsv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [ARM-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage Premium SSD	8 - 64 Disks	3750 - 153600 IOPS 106 - 5000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4163 - 199987 IOPS 124 - 5850 MBps
Network	2 - 8 NICs	12500 - 60000 Mbps
Accelerators	None	

Easv6 and Eadsv6-series

Easv6-series

Easv6-series VMs utilize AMD's fourth Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7 GHz. These virtual machines offer up to 96 vCPU and 672 GiB of RAM. The Easv6-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

View the full Easv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage Premium SSD	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage Premium SSD v2 / Ultra Disks	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

Ev5 and Esv5-series

Ev5-series

Esv5-series virtual machines run on Intel® Xeon® Platinum 8473C (Sapphire Rapids), or Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer up to 104 vCPU and 672 GiB of RAM. Esv5-series virtual machines don't have temporary storage thus lowering the price of entry.

Esv5-series supports Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

View the full Ev5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 104 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 100000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
		Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

Edv5 and Edsv5-series

Edv5-series

Edv5-series virtual machines run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer up to 104 vCPU and 672 GiB of RAM and fast, local SSD storage up to 3800 GiB. Edv5-series virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage.

Edv5-series virtual machines support Standard SSD and Standard HDD disk types. To use Premium SSD or Ultra Disk storage, select Edsv5-series virtual machines. Disk storage is billed separately from virtual machines.

View the full Edv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 104 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 672 GiB	
Local Storage	1 Disk	75 - 3800 GiB 9000 - 450000 IOPS 125 - 4000 MBps
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 100000 Mbps Interfaces: NetVSC, ConnectX, MANA

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	None	

Easv5 and Eadsv5-series

Easv5-series

Easv5-series utilize AMD's 3rd Generation EPYC™ 7763v and AMD's 4th Generation EPYC™ 9004 processors. The Easv5-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temp disk. Easv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD disk types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

View the full Easv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 112 vCPUs	AMD EPYC 7763v (Milan) [x86-64] , AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 82 - 2000 MBps
Network	2 - 8 NICs	12500 - 50000 Mbps
Accelerators	None	

Epsv5 and Epdsv5-series

Epsv5-series

Epsv5-series virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer up to 32 vCPU and 208 GiB of RAM and are ideal for memory-intensive scale-out and most Enterprise workloads. Epsv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD disk types with no local-SSD support. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

View the full Epsv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	Ampere Altra [ARM-64]
Memory	16 - 208 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 51200 IOPS 85 - 865 MBps
Network	2 - 8 NICs	12500 - 16000 Mbps
Accelerators	None	

Previous-generation E family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 02/03/2026

Esv7 sizes series

Summarize this article for me

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Esv7-series virtual machines run on the Intel® Xeon® 6973PC (Granite Rapids) processor, reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 372 vCPUs and 2,832 GiB of RAM. Esv7-series virtual machines are ideal for memory-intensive workloads, such as relational database servers, large databases, data analytics engines, in-memory caches, and more.Esv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation, for example increased networking, increased remote storage performance, upgraded CPU, and elevated memory bandwidth.

Esv7-series virtual machines are ideal for memory-intensive workloads, such as SQL and NoSQL database servers, data analytics engines, data warehousing workloads, business intelligence applications, in-memory databases such as SAP and Redis, and in-memory analytics and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 372 vCPUs	Intel@ Xeon® 6 processor 6973P-C(Granite Rapids) [x86-64]
Memory	16 - 2,832 GiB	
Local Storage	None	
Remote Storage	10 - 64 Disks	4,000 - 500,000 IOPS

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Premium SSD		115 - 16,000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	5,000 - 800,000 IOPS 135 - 20,000 MBps
Network	3 - 15 NICs	16,000 - 400,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2s_v7	2	16
Standard_E4s_v7	4	32
Standard_E8s_v7	8	64
Standard_E16s_v7	16	128
Standard_E20s_v7	20	160
Standard_E32s_v7	32	256
Standard_E48s_v7	48	384
Standard_E64s_v7	64	512
Standard_E96s_v7	96	768
Standard_E128s_v7	128	1,024
Standard_E192s_v7	192	1,536
Standard_E248s_v7	248	1,984

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E372is_v7	372	2,832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Edsv7 sizes series

Summarize this article for me

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Edsv7-series virtual machines run on the Intel® Xeon® 6 6973PC (Granite Rapids) processor reaching an all-core turbo clock speed of 3.6 GHz with a maximum turbo frequency is 4.2 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 372 vCPUs and 2,832 GiB of RAM. Edsv7-series virtual machines provide better performance for most general-purpose workloads compared to the prior generation, for example increased networking, increased remote storage performance, upgraded CPU, elevated memory bandwidth, faster NVMe local storage access, increased scalability, and larger local storage capacity.

Edsv7-series virtual machines are ideal for memory-intensive workloads, such as SQL and NoSQL database servers, data analytics engines, data warehousing workloads, business intelligence applications, in-memory databases such as SAP and Redis, and in-memory analytics and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 372 vCPUs	Intel@ Xeon® 6 6973P-C (Granite Rapids) [x86-64]
Memory	16 - 2,832 GiB	
Local Storage	1 - 6 Disks	110 - 7,040 GiB 50,000 - 6,900,000 IOPS 280 - 36,000 MBps
Remote Storage	10 - 64 Disks	4,000 - 500,000 IOPS

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Premium SSD		115 - 16,000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	5,000 - 800,000 IOPS 135 - 20,000 MBps
Network	3 - 15 NICs	16,000 - 400,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_E2ds_v7	2	16
Standard_E4ds_v7	4	32
Standard_E8ds_v7	8	64
Standard_E16ds_v7	16	128
Standard_E20ds_v7	20	160
Standard_E32ds_v7	32	256
Standard_E48ds_v7	48	384
Standard_E64ds_v7	64	512
Standard_E96ds_v7	96	768
Standard_E128ds_v7	128	1,024
Standard_E192ds_v7	192	1,536
Standard_E248ds_v7	248	1,984
Standard_E372ids_v7	372	2,832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Easv7 sizes series

Summarize this article for me

Easv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 1280 GiB of RAM. The Easv7-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	16 - 1280 GiB	
Local Storage	None	
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2as_v7	2	16
Standard_E4as_v7	4	32
Standard_E8as_v7	8	64
Standard_E16as_v7	16	128
Standard_E32as_v7	32	256
Standard_E48as_v7	48	384
Standard_E64as_v7	64	512
Standard_E96as_v7	96	768
Standard_E128as_v7	128	1024
Standard_E160as_v7	160	1280

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported

Feature name	Support status
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Eadsv7 sizes series

Summarize this article for me

Eadsv7-series VMs utilize AMD's 5th Generation EPYC™ 9005 processors that can achieve a boosted maximum frequency of up to 4.5 GHz. These virtual machines offer up to 160 vCPU and 1280 GiB of RAM. The Eadsv7-series sizes offer a combination of vCPU, memory, and fast local NVMe temporary storage that is ideal for memory-intensive enterprise applications. Eadsv7-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 160 vCPUs	AMD EPYC 9005 (Turin) [x86-64]
Memory	16 - 1280 GiB	
Local Storage	1 - 6 Disks	110 - 2200 GiB 37500 - 3000000 IOPS 280 - 22400 MBps
Remote Storage Premium SSD	10 - 64 Disks	4000 - 212000 IOPS 118 - 10344 MBps
Remote Storage Premium SSD v2 / Ultra Disks	10 - 64 Disks	4400 - 310000 IOPS 136 - 10356 MBps
Network	2 - 15 NICs	16000 - 80000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ads_v7	2	16
Standard_E4ads_v7	4	32
Standard_E8ads_v7	8	64
Standard_E16ads_v7	16	128
Standard_E32ads_v7	32	256
Standard_E48ads_v7	48	384
Standard_E64ads_v7	64	512
Standard_E96ads_v7	96	768
Standard_E128ads_v7	128	1024
Standard_E160ads_v7	160	1280

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Esv6 sizes series

Summarize this article for me

Esv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all-core turbo clock speed of 3.0 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 128 vCPU and 1,024 GiB of RAM. Esv6-series virtual machines are ideal for memory-intensive workloads, such as relational database servers, large databases, data analytics engines, in-memory caches, and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1024 GiB	
Local Storage	None	
Remote Storage Premium SSD	8 - 64 Disks	3750 - 260000 IOPS 106 - 12000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4167 - 400000 IOPS 124 - 12000 MBps
Network	2 - 8 NICs	12500 - 54000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2s_v6	2	16
Standard_E4s_v6	4	32
Standard_E8s_v6	8	64
Standard_E16s_v6	16	128
Standard_E20s_v6	20	160
Standard_E32s_v6	32	256
Standard_E48s_v6	48	384
Standard_E64s_v6	64	512
Standard_E96s_v6	96	768
Standard_E128s_v6	128	1024
Standard_E192is_v6	192	1832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Edsv6 sizes series

Edsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all-core turbo clock speed of 3.0 GHz. This new processor features Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512), and Intel® Advanced Matrix Extensions. These virtual machines offer up to 192 vCPU and 1,832 GiB of RAM and fast, local SSD storage up to 6x1760 GiB. Edsv6-series virtual machines are ideal for memory-intensive workloads such as relational database servers, large databases, data analytics engines, in-memory caches, and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1024 GiB	
Local Storage	1 - 4 Disks	110 - 1760 GiB 37500 - 2400000 IOPS 180 - 11520 MBps
Remote Storage Premium SSD	8 - 64 Disks	3750 - 360000 IOPS 106 - 17280 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4167 - 400000 IOPS 124 - 12000 MBps
Network	2 - 8 NICs	12500 - 54000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ds_v6	2	16
Standard_E4ds_v6	4	32
Standard_E8ds_v6	8	64
Standard_E16ds_v6	16	128
Standard_E20ds_v6	20	160
Standard_E32ds_v6	32	256
Standard_E48ds_v6	48	384
Standard_E64ds_v6	64	512
Standard_E96ds_v6	96	768
Standard_E128ds_v6	128	1024
Standard_E192ids_v6	192	1832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Epsv6 sizes series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows Client VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Epsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for memory optimized workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 8 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs are best suited for memory-intensive scale-out and enterprise workloads, such as relational database servers, large databases, data analytics engines, in-memory caches, and more. Epsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Epsv6-series virtual machines offer up to 96 vCPU and 672 GiB of RAM. These VMs come without local disk, providing a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Azure Cobalt 100 [ARM-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage Premium SSD	8 - 64 Disks	3750 - 153600 IOPS 106 - 5000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4163 - 199987 IOPS 124 - 5850 MBps
Network	2 - 8 NICs	12500 - 60000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size:

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_E2ps_v6	2	16
Standard_E4ps_v6	4	32
Standard_E8ps_v6	8	64
Standard_E16ps_v6	16	128
Standard_E32ps_v6	32	256
Standard_E48ps_v6	48	384
Standard_E64ps_v6	64	512
Standard_E96ps_v6	96	672

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported

Feature name	Support status
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Not supported
Nested Virtualization	Not supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Epdsv6 sizes series

Applies to: ✔ Linux VMs ✔ Windows Client VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Epdsv6-series virtual machines are powered by Azure's first-generation Cobalt 100 processor, delivering outstanding performance for memory optimized workloads. The Azure Cobalt 100 processor operates at 3.4 GHz and provides an entire physical core for each virtual machine vCPU. These virtual machines offer 8 GiB of memory per vCPU and a wide range of vCPU sizes. These VMs are best suited for memory-intensive scale-out and enterprise workloads, such as relational database servers, large databases, data analytics engines, in-memory caches, and more. Epdsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. For more information, see pricing for disks. Epdsv6-series virtual machines offer up to 96 vCPU, up to 672 GiB of RAM, and up to 5,280 GiB of fast local NVMe temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCores	Azure Cobalt 100 [ARM-64]
Memory	16 - 672 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB/Disk 15000 - 1800000 IOPS 90 - 8640 MBps
Remote Storage Premium SSD	8 - 64 Disks	3750 - 153600 IOPS 106 - 5000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	8 - 64 Disks	4163 - 199987 IOPS 124 - 5850 MBps
Network	2 - 8 NICs	12500 - 60000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_E2pds_v6	2	16
Standard_E4pds_v6	4	32
Standard_E8pds_v6	8	64
Standard_E16pds_v6	16	128
Standard_E32pds_v6	32	256
Standard_E48pds_v6	48	384
Standard_E64pds_v6	64	512
Standard_E96pds_v6	96	672

> **Note**
>
> The Epdsv6 VM series will only work on OS images that support NVMe (i.e. NVMe drivers required for the local storage). If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

VM Basics resources

- What are vCPUs (Qty.)
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 2
Accelerated Networking	Supported
Ephemeral OS Disks	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Easv6 sizes series

Summarize this article for me

Easv6-series VMs utilize AMD's fourth Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7 GHz. These virtual machines offer up to 96 vCPU and 672 GiB of RAM. The Easv6-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temporary storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage Premium SSD	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Remote Storage Premium SSD v2 / Ultra Disks	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2as_v6	2	16
Standard_E4as_v6	4	32
Standard_E8as_v6	8	64
Standard_E16as_v6	16	128
Standard_E20as_v6	20	160
Standard_E32as_v6	32	256
Standard_E48as_v6	48	384
Standard_E64as_v6	64	512
Standard_E96as_v6	96	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. **NVMe** support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Eadsv6 sizes series

Summarize this article for me

Eadsv6-series VMs utilize AMD's fourth Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7 GHz. These virtual machines offer up to 96 vCPU and 672 GiB of RAM. The Eadsv6-series sizes offer a combination of vCPU, memory, and fast local NVMe temporary storage that is ideal for memory-intensive enterprise applications. Eadsv6-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	1 - 6 Disks	110 - 880 GiB 37500 - 1800000 IOPS 180 - 8640 MBps
Remote Storage Premium SSD	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Remote Storage Premium SSD v2 / Ultra Disks	4 - 32 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ads_v6	2	16
Standard_E4ads_v6	4	32
Standard_E8ads_v6	8	64
Standard_E16ads_v6	16	128
Standard_E20ads_v6	20	160
Standard_E32ads_v6	32	256
Standard_E48ads_v6	48	384
Standard_E64ads_v6	64	512
Standard_E96ads_v6	96	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported

Feature name	Support status
Temporary local NVMe disks	Supported
Nested Virtualization	Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Ev5 sizes series

Summarize this article for me

Ev5-series virtual machines run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids) or Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer up to 104 vCPU and 672 GiB of RAM. Esv5-series virtual machines don't have temporary storage thus lowering the price of entry.

Ev5-series supports Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 104 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 100000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2_v5	2	16
Standard_E4_v5	4	32
Standard_E8_v5	8	64
Standard_E16_v5	16	128
Standard_E20_v5	20	160
Standard_E32_v5	32	256
Standard_E48_v5	48	384
Standard_E64_v5	64	512
Standard_E96_v5	96	672
Standard_E104i_v5	104	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Esv5 sizes series

Esv5-series virtual machines run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids) or Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer up to 104 vCPU and 672 GiB of RAM. Esv5-series virtual machines don't have temporary storage thus lowering the price of entry.

Esv5-series supports Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 104 vCPUs	Intel Xeon Platinum 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 100000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2s_v5	2	16
Standard_E4s_v5	4	32
Standard_E8s_v5	8	64
Standard_E16s_v5	16	128
Standard_E20s_v5	20	160
Standard_E32s_v5	32	256
Standard_E48s_v5	48	384
Standard_E64s_v5	64	512
Standard_E96s_v5	96	672
Standard_E104is_v5	104	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Edv5 sizes series

Summarize this article for me

Edv5-series virtual machines run on the Intel® Xeon® Platinum 8573C (Emerald Rapids) or the Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer up to 104 vCPU and 672 GiB of RAM and fast, local SSD storage up to 3800 GiB. Edv5-series virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage.

Edv5-series virtual machines support Standard SSD and Standard HDD types. To use Premium SSD or Ultra Disk storage, select Edsv5-series virtual machines. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 104 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 672 GiB	
Local Storage	1 Disk	75 - 3800 GiB 9000 - 450000 IOPS 125 - 4000 MBps
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 100000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2d_v5	2	16
Standard_E4d_v5	4	32
Standard_E8d_v5	8	64
Standard_E16d_v5	16	128
Standard_E20d_v5	20	160
Standard_E32d_v5	32	256
Standard_E48d_v5	48	384
Standard_E64d_v5	64	512
Standard_E96d_v5	96	672
Standard_E104id_v52	104	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported

Feature name	Support status
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Edsv5 sizes series

Edsv5-series virtual machines run on the Intel® Xeon® Platinum 8573C (Emerald Rapids) or the Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper threaded configuration. This new processor features an all core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. These virtual machines offer up to 104 vCPU and 672 GiB of RAM and fast, local SSD storage up to 3800 GiB. Edsv5-series virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage.

Edsv5-series virtual machines support Standard SSD and Standard HDD types. You can attach Standard SSD, Standard HDD, and Premium SSD storage to these VMs. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines. See pricing for disks. The Edsv5-series virtual machines can burst their disk performance and get up to their bursting max for up to 30 minutes at a time.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 104 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 672 GiB	
Local Storage	1 Disk	75 - 3800 GiB 9000 - 450000 IOPS 125 - 4000 MBps
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 85 - 4000 MBps
Network	2 - 8 NICs	12500 - 100000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ds_v5	2	16
Standard_E4ds_v5	4	32
Standard_E8ds_v5	8	64
Standard_E16ds_v5	16	128
Standard_E20ds_v5	20	160
Standard_E32ds_v5	32	256
Standard_E48ds_v5	48	384
Standard_E64ds_v5	64	512
Standard_E96ds_v5	96	672
Standard_E104ids_v5	104	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported

Feature name	Support status
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Easv5 sizes series

Summarize this article for me

Easv5-series utilize AMD's 3rd Generation EPYC™ 7763v and AMD's 4th Generation EPYC™ 9004 processors. The Easv5-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. The new VMs with no local disk provide a better value proposition for workloads that don't require local temp disk. Easv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 112 vCPUs	AMD EPYC 7763v (Milan) [x86-64] , AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 82 - 2000 MBps
Network	2 - 8 NICs	12500 - 50000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2as_v5	2	16
Standard_E4as_v5	4	32
Standard_E8as_v5	8	64
Standard_E16as_v5	16	128
Standard_E20as_v5	20	160
Standard_E32as_v5	32	256
Standard_E48as_v5	48	384
Standard_E64as_v5	64	512
Standard_E96as_v5	96	672
Standard_E112ias_v5	112	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported

Feature name	Support status
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Eadsv5 sizes series

Summarize this article for me

Eadsv5-series utilize AMD's 3rd Generation EPYC™ 7763v and AMD's 4th Generation EPYC™ 9004 processors. The Eadsv5-series sizes offer a combination of vCPU, memory, and temporary storage that is ideal for memory-intensive enterprise applications. The new VMs have 50% larger local storage, and better local disk IOPS for both read and write compared to the Eav4/Easv4 sizes with Gen2 VMs. Eadsv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 112 vCPUs	AMD EPYC 7763v (Milan) [x86-64] , AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	1 Disk	75 - 3800 GiB 9000 - 450000 IOPS 125 - 4000 MBps
Remote Storage	4 - 64 Disks	3750 - 120000 IOPS 82 - 2000 MBps
Network	2 - 8 NICs	12500 - 50000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ads_v5	2	16
Standard_E4ads_v5	4	32
Standard_E8ads_v5	8	64
Standard_E16ads_v5	16	128
Standard_E20ads_v5	20	160
Standard_E32ads_v5	32	256
Standard_E48ads_v5	48	384
Standard_E64ads_v5	64	512
Standard_E96ads_v5	96	672
Standard_E112iads_v5	112	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported

Feature name	Support status
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Ensv6 sizes series (Preview)

Summarize this article for me

Ensv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all- core turbo clock speed of up to 3.0 GHz. These virtual machines offer up to 128 vCPU and 1024 GiB of RAM. Ensv6-series virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage.

Network Optimized VMs enhance accelerated networking by providing hardware acceleration of initial connection setup for certain traffic types, a task previously performed in software. These enhancements reduce the end-to-end latency for initially establishing a connection or initial packet flow, and allow a VM to scale up the number of connections it manages more quickly, subject to application constraints.

The Ensv6 VMs offer up to 200 Gbps network bandwidth, 15 vNICs and up to 400k Connection per Second (CPS). The series also includes up to 128 vCPU and 1024 GiB of RAM.

Both series provide up to 3x improvement in NW BW/vCPU than the current generation Intel Ev6 VMs. This series will be relevant for workloads such as network virtual appliances, large-scale e-commerce applications, express route, application gateway, and more that require the ability to handle a high number of user connections and data transfers. They're also suited for media processing tasks that involve transferring large amounts of data quickly, and other applications that benefit from high NW BW/vCPU and CPS & flow limit configurations.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1024 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	3750 - 204800 IOPS 106 - 6782 MBps
Network	4 - 15 NICs	25000 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ns_v6	2	16
Standard_E4ns_v6	4	32
Standard_E8ns_v6	8	64
Standard_E16ns_v6	16	128
Standard_E32ns_v6	32	256
Standard_E48ns_v6	48	384
Standard_E64ns_v6	64	512
Standard_E96ns_v6	96	768
Standard_E128ns_v6	128	1024

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported

Feature name	Support status
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Endsv6 sizes series (Preview)

Summarize this article for me

Endsv6-series virtual machines run on the 5th Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all- core turbo clock speed of up to 3.0 GHz. These virtual machines offer up to 128 vCPU and 1024 GiB of RAM. Endsv6-series virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage.

Network Optimized VMs enhance accelerated networking by providing hardware acceleration of initial connection setup for certain traffic types, a task previously performed in software. These enhancements reduce the end-to-end latency for initially establishing a connection or initial packet flow, and allow a VM to scale up the number of connections it manages more quickly, subject to application constraints.

Endsv6 VMs offer up to 200 Gbps network bandwidth, 15 vNICs and up to 400k Connection per Second (CPS). Both series also include up to 128 vCPU and 1024 GiB of RAM.

Both series provide up to 3x improvement in NW BW/vCPU than the current generation Intel Ev6 VMs. This series will be relevant for workloads such as network virtual appliances, large-scale e-commerce applications, express route, application gateway, and more that requires the ability to handle a high number of user connections and data transfers. They're also suited for media processing tasks that involve transferring large amounts of data quickly, and other applications that benefit from high NW BW/vCPU and CPS & flow limit configurations.

The Endsv6 series comes with a local disk. We recommend choosing Premium SSD, Premium SSD v2, or Ultra Disks to attain the published disk performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 128 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1024 GiB	
Local Storage	1 - 4 Disks	110 - 1760 GiB 37500 - 2400000 180 - 11520 MBps
Remote Storage	8 - 64 Disks	3750 - 204800 IOPS

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
		106 - 6782 MBps
Network	4 - 15 NICs	25000 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2nds_v6	2	16
Standard_E4nds_v6	4	32
Standard_E8nds_v6	8	64
Standard_E16nds_v6	16	128
Standard_E32nds_v6	32	256
Standard_E48nds_v6	48	384
Standard_E64nds_v6	64	512
Standard_E96nds_v6	96	768
Standard_E128nds_v6	128	1024

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Epsv5 sizes series

Summarize this article for me

Epsv5-series virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer up to 32 vCPU and 208 GiB of RAM and are ideal for memory-intensive scale-out and most Enterprise workloads. Epsv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types with no local-SSD support. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	Ampere Altra [ARM-64]
Memory	16 - 208 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3750 - 51200 IOPS 85 - 865 MBps
Network	2 - 8 NICs	12500 - 16000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ps_v5	2	16
Standard_E4ps_v5	4	32
Standard_E8ps_v5	8	64
Standard_E16ps_v5	16	128
Standard_E20ps_v5	20	160
Standard_E32ps_v5	32	208

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Epdsv5 sizes series

Epdsv5-series virtual machines feature the Ampere® Altra® Arm-based processor operating at 3.0 GHz, which provides an entire physical core for each virtual machine vCPU. These virtual machines offer up to 32 vCPU, 208 GiB of RAM, and fast local SSD storage up to 1,200 GiB and are ideal for memory-intensive scale-out and most Enterprise workloads. Epdsv5-series virtual machines support Standard SSD, Standard HDD, and Premium SSD types. You can also attach Ultra Disk storage based on its regional availability. Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	Ampere Altra [ARM-64]
Memory	16 - 208 GiB	
Local Storage	1 Disk	75 - 1200 GiB 9375 - 150000 IOPS 125 - 2000 MBps
Remote Storage	4 - 32 Disks	3750 - 51200 IOPS 85 - 865 MBps
Network	2 - 8 NICs	12500 - 16000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2pds_v5	2	16
Standard_E4pds_v5	4	32
Standard_E8pds_v5	8	64
Standard_E16pds_v5	16	128
Standard_E20pds_v5	20	160
Standard_E32pds_v5	32	208

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Edv4 sizes series

Summarize this article for me

Edv4-series sizes run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) or the Intel® Xeon® Platinum 8272CL (Cascade Lake) processors. The Edv4 virtual machine sizes feature up to 504 GiB of RAM, in addition to fast and large local SSD storage (up to 2,400 GiB). These virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage. You can attach Standard SSDs and Standard HDDs disk storage to the Edv4 VMs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 504 GiB	
Local Storage	1 Disk	75 - 2400 GiB 9000 - 300000 IOPS 125 - 4000 MBps
Remote Storage	4 - Disks	3200 - IOPS 48 - MBps
Network	2 - 8 NICs	5000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2d_v4	2	16
Standard_E4d_v4	4	32
Standard_E8d_v4	8	64
Standard_E16d_v4	16	128
Standard_E20d_v4	20	160
Standard_E32d_v4	32	256
Standard_E48d_v4	48	384
Standard_E64d_v4	64	504

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Edsv4 sizes series

Edsv4-series sizes run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake) or the Intel® Xeon® Platinum 8272CL (Cascade Lake) processors. The Edsv4 virtual machine sizes feature up to 504 GiB of RAM, in addition to fast and large local SSD storage (up to 2,400 GiB). These virtual machines are ideal for memory-intensive enterprise applications and applications that benefit from low latency, high-speed local storage.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 80 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 504 GiB	
Local Storage	1 Disk	75 - 2400 GiB 9000 - 375000 IOPS 125 - 4000 MBps
Remote Storage	4 - 64 Disks	3200 - 80000 IOPS 48 - 1500 MBps
Network	2 - 8 NICs	5000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2ds_v4	2	16
Standard_E4ds_v4	4	32
Standard_E8ds_v4	8	64
Standard_E16ds_v4	16	128
Standard_E20ds_v4	20	160
Standard_E32ds_v4	32	256
Standard_E48ds_v4	48	384
Standard_E64ds_v4	64	504
Standard_E80ids_v4	80	504

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Eav4 sizes series

Summarize this article for me

The Eav4-series run on 2nd Generation AMD EPYCTM 7452 (up to 3.35GHz) or 3rd Generation EPYCTM 7763v processors (up to 3.5GHz). The Eav4-series sizes are ideal for memory-intensive enterprise applications. Data disk storage is billed separately from virtual machines. To use Premium SSD, use the Easv4-series sizes. The pricing and billing meters for Easv4 sizes are the same as the Eav3-series.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7452 (Rome) [x86-64] AMD EPYC 7763v (Milan) [x86-64]
Memory	16 - 672 GiB	
Local Storage	4 - 32 Disks	16 - 672 GiB 3000 - 96000 IOPS 46 - 1000 MBps
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 48 - 1200 MBps
Network	2 - 8 NICs	2000 - 32000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2a_v4	2	16
Standard_E4a_v4	4	32
Standard_E8a_v4	8	64
Standard_E16a_v4	16	128
Standard_E20a_v4	20	160
Standard_E32a_v4	32	256
Standard_E48a_v4	48	384
Standard_E64a_v4	64	512
Standard_E96a_v4	96	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Easv4 sizes series

Summarize this article for me

The Easv4-series run on 2nd Generation AMD EPYC™ 7452 (up to 3.35GHz) or 3rd Generation EPYC™ 7763v processors (up to 3.5GHz) and use Premium SSD. The Easv4-series sizes are ideal for memory-intensive enterprise applications.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 7452 (Rome) [x86-64] AMD EPYC 7763v (Milan) [x86-64]
Memory	16 - 672 GiB	
Local Storage	1 Disk	32 - 1344 GiB 4000 - 192000 IOPS 32 - 1020 MBps
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 48 - 1200 MBps
Network	2 - 8 NICs	2000 - 32000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2as_v4	2	16
Standard_E4as_v4	4	32

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E8as_v4	8	64
Standard_E16as_v4	16	128
Standard_E20as_v4	20	160
Standard_E32as_v4	32	256
Standard_E48as_v4	48	384
Standard_E64as_v4	64	512
Standard_E96as_v4	96	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Ev4 sizes series

Ev4-series sizes run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), or Intel® Xeon® Platinum 8272CL (Cascade Lake). The Ev4-series instances are ideal for memory-intensive enterprise applications. Ev4-series VMs feature Intel® Hyper-Threading Technology.

Remote Data disk storage is billed separately from virtual machines. To use premium storage disks, use the Esv4 sizes. The pricing and billing meters for Esv4 sizes are the same as Ev4-series.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 504 GiB	
Local Storage	None	
Remote Storage	4 - 32 Disks	3200 - 80000 IOPS 48 - 1500 MBps
Network	2 - 8 NICs	5000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2_v4	2	16
Standard_E4_v4	4	32
Standard_E8_v4	8	64
Standard_E16_v4	16	128
Standard_E20_v4	20	160
Standard_E32_v4	32	256
Standard_E48_v4	48	384
Standard_E64_v4	64	504

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Esv4 sizes series

Summarize this article for me

Esv4-series sizes run on Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8473C (Sapphire Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), or Intel® Xeon® Platinum 8272CL (Cascade Lake). The Esv4-series instances are ideal for memory-intensive enterprise applications. Evs4-series VMs feature Intel® Hyper-Threading Technology. Remote Data disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 80 vCPUs	Intel Xeon 8473C (Sapphire Rapids) [x86-64] Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 504 GiB	
Local Storage	None	
Remote Storage	4 - 64 Disks	3200 - 80000 IOPS 48 - 1500 MBps
Network	2 - 8 NICs	5000 - 30000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2s_v4	2	16
Standard_E4s_v4	4	32
Standard_E8s_v4	8	64
Standard_E16s_v4	16	128
Standard_E20s_v4	20	160
Standard_E32s_v4	32	256
Standard_E48s_v4	48	384
Standard_E64s_v4	64	504
Standard_E80is_v4	80	504

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

'Eb' family memory optimized VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'Eb' family of VM size series are one of Azure's memory-optimized VM instances. They're designed for memory-intensive workloads with high remote storage performance, such as large databases, big data analytics, and enterprise applications that require significant amounts of RAM to maintain high performance. Equipped with high memory-to-core ratios, Eb-series VMs support applications and services that benefit from faster data access and more efficient data processing capabilities. This makes them particularly well-suited for scenarios involving in-memory databases and extensive data processing tasks where ample memory is crucial for optimal performance.

Workloads and use cases

Memory-Intensive Workloads: Eb-family VMs are for workloads that demand a large memory footprint to efficiently handle tasks, such as simulations, large-scale computations in scientific research, or financial risk modeling.

Large Databases and SQL Servers: Eb-family VMs are ideal for hosting large relational databases like SQL Server and NoSQL databases that benefit from high memory capacities for improved performance in data processing and transaction handling.

Enterprise Applications: Eb-family VMs are suitable for resource-intensive enterprise applications, including large-scale ERP and CRM systems, where the availability of ample memory is crucial for managing complex transactions and user loads.

Big Data Applications: Eb-family VMs are effective for big data analytics applications that need to process vast amounts of data in memory to speed up analysis and insights generation.

In-Memory Computing: Eb-family VMs are great for in-memory databases (e.g., SAP HANA) that require large amounts of RAM to keep the entire dataset in memory, allowing for ultra-fast data processing and query responses.

Data Warehousing: Eb-family VMs provide the necessary resources for data warehousing solutions that handle and analyze large datasets, improving query performance and reducing response times.

Series in family

Ebsv6 and Ebdsv6-series

Ebsv6-series

The Ebsv6-series virtual machines run on the fifth Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor and deliver higher remote storage performance compared to the previous generation. The Ebsv6-series is ideal for storage throughput intensive workloads. These virtual machines offer up to 800,000 IOPS and 14,000 MBps of remote disk storage performance. Additionally, the Ebsv6 VM series offer up to 192vCPU and 1,832 GiB of RAM providing larger processing power to consolidate workloads. Refer to the Premium SSD, Premium SSD v2, and Ultra disks to attain published disk performance.

View the full Ebsv6-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1832 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	8800 - 800000 IOPS 253 - 14000 MBps
Network	2 - 8 NICs	12500 - 200000 Mbps
Accelerators	None	

Ebdsv5 and Ebsv5-series

The Ebdsv5 and Ebsv5 series run on the Intel® Xeon® Platinum 8370C (Ice Lake) processors in a hyper-threaded configuration. The series are ideal for various memory-intensive enterprise applications. The Ebsv5 and Ebdsv5 VMs offer up to 260000 IOPS and 8000 MBps of remote disk storage throughput. Both series also include up to 672 GiB of RAM. The Ebdsv5 series has local SSD storage up to 3800 GiB. Both series provide a 3X increase in remote storage performance of data-intensive workloads compared to prior VM generations. You can use these series to consolidate existing workloads on fewer VMs or smaller VM sizes while achieving potential cost savings. The Ebdsv5 series comes with a local disk and Ebsv5 is without

a local disk. We recommend choosing Premium SSD, Premium SSD v2 or Ultra disks to attain the published disk performance.

View the full Ebdsv5 and Ebsv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 112 vCores	Intel® Xeon® Platinum 8370C (Ice Lake)
Memory	16 - 672 GiB	
Data Disks	4 - 64 Disks	5500 - 450000 IOPS / 125 - 8000 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

Previous-generation Eb family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 11/18/2025

Ebsv6 sizes series (Preview)

Summarize this article for me

The Ebsv6-series virtual machines run on the fifth Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor and deliver higher remote storage performance compared to the previous generation. The Ebsv6-series is ideal for storage throughput intensive workloads. These virtual machines offer up to 800,000 IOPS and 14,000 MBps of remote disk storage performance. Additionally, the Ebsv6 VM series offer up to 192vCPU and 1,832 GiB of RAM providing larger processing power to consolidate workloads. Refer to the Premium SSD, Premium SSD v2, and Ultra Disks to attain published disk performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1832 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	8800 - 800000 IOPS 253 - 14000 MBps
Network	2 - 8 NICs	12500 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2bs_v6	2	16
Standard_E4bs_v6	4	32
Standard_E8bs_v6	8	64
Standard_E16bs_v6	16	128
Standard_E32bs_v6	32	256
Standard_E48bs_v6	48	384
Standard_E64bs_v6	64	512
Standard_E96bs_v6	96	768
Standard_E128bs_v6	128	1024
Standard_E192ibs_v6	192	1832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported

Feature name	Support status
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Ebdsv6 sizes series (Preview)

Summarize this article for me

The Ebdsv6-series virtual machines run on the fifth Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor and deliver higher remote storage performance compared to the previous generation. The Ebdsv6-series is ideal for storage throughput intensive workloads. These virtual machines offer up to 800,000 IOPS and 14,000 MBps of remote disk storage performance. Additionally, the Ebsv6 VM series offer up to 192vCPU and 1,832 GiB of RAM providing larger processing power to consolidate workloads. Refer to the Premium SSD, Premium SSD v2, and Ultra Disks to attain published disk performance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 192 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 1832 GiB	
Local Storage	1 - 6 Disks	110 - 1760 GiB 37500 - 3600000 IOPS 180 - 17280 MBps
Remote Storage	8 - 64 Disks	8800 - 800000 IOPS 253 - 14000 MBps
Network	2 - 8 NICs	12500 - 200000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_E2bds_v6	2	16
Standard_E4bds_v6	4	32
Standard_E8bds_v6	8	64
Standard_E16bds_v6	16	128
Standard_E32bds_v6	32	256
Standard_E48bds_v6	48	384
Standard_E64bds_v6	64	512
Standard_E96bds_v6	96	768
Standard_E128bds_v6	128	1024
Standard_E192ibds_v6	192	1832

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported

Feature name	Support status
Nested Virtualization	Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Ebdsv5 and Ebsv5 series

Summarize this article for me

The memory-optimized Ebsv5 and Ebdsv5 Azure virtual machine (VM) series deliver higher remote storage performance in each VM size than the Ev4 series. The increased remote storage performance of the Ebsv5 and Ebdsv5 VMs is ideal for storage throughput-intensive workloads. For example, relational databases and data analytics applications.

The Ebsv5 and Ebdsv5 VMs offer up to 260000 IOPS and 8000 MBps of remote disk storage throughput. Both series also include up to 672 GiB of RAM. The Ebdsv5 series has local SSD storage up to 3800 GiB. Both series provide a 3X increase in remote storage performance of data-intensive workloads compared to prior VM generations. You can use these series to consolidate existing workloads on fewer VMs or smaller VM sizes while achieving potential cost savings. The Ebdsv5 series comes with a local disk and Ebsv5 is without a local disk. We recommend choosing Premium SSD, Premium SSD v2 or Ultra Disks to attain the published disk performance.

The Ebdsv5 and Ebsv5 series run on the Intel® Xeon® Platinum 8370C (Ice Lake) processors in a hyper-threaded configuration. The series are ideal for various memory-intensive enterprise applications. They feature:

- Up to 512 GiB of RAM
- Intel® Turbo Boost Technology 2.0
- Intel® Hyper-Threading Technology
- Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
- Support for Intel® Deep Learning Boost
- NVMe interface for higher remote disk storage IOPS and throughput performance

Important

- Accelerated networking is required and turned on by default on all Ebsv5 and Ebdsv5 VMs.
- Ebsv5 and Ebdsv5-series VMs can **burst their disk performance** and get up to their bursting max for up to 30 minutes at a time.
- The E112i size is offered as NVMe only to provide the highest IOPS and throughput performance. See the NVMe VM spec table to see the improved performance details.

Ebdsv5 series

Ebdsv5-series sizes run on the Intel® Xeon® Platinum 8370C (Ice Lake) processors. The Ebdsv5 VM sizes feature up to 672 GiB of RAM, in addition to fast and large local SSD storage (up to 3800 GiB). These VMs are ideal for memory-intensive enterprise applications and applications that benefit from high remote storage performance, low latency, high-speed local storage. Remote Data disk storage is billed separately from VMs.

- Premium Storage: Supported
- Premium Storage caching: Supported
- Live Migration: Supported
- Memory Preserving Updates: Supported
- VM Generation Support: Generation 1 and Generation 2
- Accelerated Networking: Supported (required)
- Ephemeral OS Disks: Supported
- Nested virtualization: Supported
- NVMe Interface: Supported only on Generation 2 VMs
- SCSI Interface: Supported on Generation 1 and 2 VMs

Ebdsv5 Series (NVMe)

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max temp storage throughput: IOPS / MBps	Max uncached Premium SSD throughput: IOPS/MBps	Max burst uncached Premium SSD throughput: IOPS/MBps	Max uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max burst uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max NICs	Network bandwidth
Standard_E2bds_v5	2	16	75	4	9000/125	8000/230	10000/1200	12000/300	15000/1200	2	12500
Standard_E4bds_v5	4	32	150	8	19000/250	16000/460	20000/1200	21400/600	30000/1200	2	12500
Standard_E8bds_v5	8	64	300	16	38000/500	33000/930	40000/1200	44200/1200	60000/1200	4	12500
Standard_E16bds_v5	16	128	600	32	75000/1000	66000/1800	70000/2000	88400/2300	96000/2600	8	12500
Standard_E32bds_v5	32	256	1200	32	150000/2000	130000/3700	140000/4000	174200/4800	180000/5200	8	16000
Standard_E48bds_v5	48	384	1800	32	225000/3000	190000/5600	200000/6000	253300/7300	260000/7850	8	16000
Standard_E64bds_v5	64	512	2400	32	300000/4000	220000/6000	230000/6500	294800/7800	310000/8500	8	20000
Standard_E96bds_v5	96	672	3600	32	450000/4000	260000/7500	260000/8000	390000/8500	390000/9000	8	25000
Standard_E112ibds_v5	112	672	3800	64	450000/4000	260000/8000	260000/8000	400000/10000	400000/10000	8	40000

Ebdsv5 Series (SCSI)

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max temp storage throughput: IOPS / MBps	Max uncached Premium SSD throughput: IOPS/MBps	Max burst uncached Premium SSD throughput: IOPS/MBps	Max uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max burst uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max NICs	Network bandwidth
Standard_E2bds_v5	2	16	75	4	9000/125	6000/170	10000/1200	8000/170	15000/1200	2	12500
Standard_E4bds_v5	4	32	150	8	19000/250	12100/350	20000/1200	16200/350	30000/1200	2	12500
Standard_E8bds_v5	8	64	300	16	38000/500	24200/680	40000/1200	32400/680	60000/1200	4	12500
Standard_E16bds_v5	16	128	600	32	75000/1000	48400/1370	64000/2000	64800/1370	96000/2000	8	12500
Standard_E32bds_v5	32	256	1200	32	150000/2000	96800/2740	120000/4000	129700/2740	160000/4000	8	16000
Standard_E48bds_v5	48	384	1800	32	225000/3000	120000/4000	120000/4000	160000/4000	160000/4000	8	16000
Standard_E64bds_v5	64	512	2400	32	300000/4000	120000/4000	120000/4000	160800/6500	160800/6500	8	20000
Standard_E96bds_v5	96	672	3600	32	450000/4000	120000/4000	120000/4000	180000/6500	180000/6500	8	25000

Ebsv5 series

Ebsv5-series sizes run on the Intel® Xeon® Platinum 8272CL (Cascade Lake). These VMs are ideal for memory-intensive enterprise applications and applications that benefit from high remote storage performance but with no local SSD storage. Ebsv5-series VMs feature Intel® Hyper-Threading Technology. Remote Data disk storage is billed separately from VMs.

- Premium Storage: Supported
- Premium Storage caching: Supported
- Live Migration: Supported
- Memory Preserving Updates: Supported
- VM Generation Support: Generation 1 and Generation 2
- Accelerated Networking: Supported (required)
- Ephemeral OS Disks: Not supported
- Nested virtualization: Supported

- NVMe Interface: Supported only on Generation 2 VMs
- SCSI Interface: Supported on Generation 1 and Generation 2 VMs

Ebsv5 Series (NVMe)

Expand table

Size	vCPU	Memory: GiB	Max data disks	Max uncached Premium SSD throughput: IOPS/MBps	Max burst uncached Premium SSD throughput: IOPS/MBps	Max uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max burst uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max NICs	Network bandwidth
Standard_E2bs_v5	2	16	4	8000/230	10000/1200	12000/300	15000/1200	2	12500
Standard_E4bs_v5	4	32	8	16000/460	20000/1200	21400/600	30000/1200	2	12500
Standard_E8bs_v5	8	64	16	33000/930	40000/1200	44200/1200	60000/1200	4	12500
Standard_E16bs_v5	16	128	32	66000/1800	70000/2000	88400/2300	96000/2600	8	12500
Standard_E32bs_v5	32	256	32	130000/3700	140000/4000	174200/4800	180000/5200	8	16000
Standard_E48bs_v5	48	384	32	190000/5600	200000/6000	253300/7300	260000/7850	8	16000
Standard_E64bs_v5	64	512	32	220000/6000	230000/6500	294800/7800	310000/8500	8	20000
Standard_E96bs_v5	96	672	32	260000/7500	260000/8000	390000/8500	390000/9000	8	25000
Standard_E112ibs_v5	112	672	64	260000/8000	260000/8000	400000/10000	400000/10000	8	40000

Ebsv5 Series (SCSI)

Expand table

Size	vCPU	Memory: GiB	Max data disks	Max uncached Premium SSD throughput: IOPS/MBps	Max burst uncached Premium SSD throughput: IOPS/MBps	Max uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max burst uncached Ultra Disk and Premium SSD v2 throughput: IOPS/MBps	Max NICs	Network bandwidth
Standard_E2bs_v5	2	16	4	6000/170	10000/1200	8000/170	15000/1200	2	12500
Standard_E4bs_v5	4	32	8	12100/350	20000/1200	16200/350	30000/1200	2	12500
Standard_E8bs_v5	8	64	16	24200/680	40000/1200	32400/680	60000/1200	4	12500
Standard_E16bs_v5	16	128	32	48400/1370	64000/2000	64800/1370	96000/2000	8	12500
Standard_E32bs_v5	32	256	32	96800/2740	120000/4000	129700/2740	160000/4000	8	16000
Standard_E48bs_v5	48	384	32	120000/4000	120000/4000	160000/4000	160000/4000	8	16000
Standard_E64bs_v5	64	512	32	120000/4000	120000/4000	160800/6500	160800/6500	8	20000
Standard_E96bs_v5	96	672	32	120000/4000	120000/4000	180000/6500	180000/6500	8	25000

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.

- To learn how to get the best storage performance for your VMs, see Virtual machine and disk performance.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth.

  Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see Optimize network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Ebsv5 NVMe FAQ

How is the NVMe enabled Ebsv5 different from the L series VM that Azure offers?

The NVMe enabled Ebsv5 series is designed to offer the highest Azure Managed Disk storage performance. The L series VMs are designed to offer higher IOPS and throughout on the local NVMe disks, which are ephemeral. Refer to the VM sizes documentation for details on the performance offered by the Ebsv5 and L series.

What I/O size is recommended to achieve the published performance?

To achieve the maximum IOPS, we recommend using a 4 KiB or 8 KiB block size. For maximum performance throughput, you can choose to use one of the following block sizes: 64 KiB, 128 KiB, 256 KiB, 512 KiB or 1024 KiB. However, it's important to optimize the I/O size based on the specific requirements of your application and to use the recommended block sizes only as a guideline.

What workloads benefit with NVMe on Ebsv5 family?

The Ebsv5 VM families are suitable for various workloads that require high I/O and improved remote storage performance. Some examples of such workloads include:

- Online transaction processing (OLTP) workloads: These workloads involve frequent, small, and fast database transactions, such as online banking, e-commerce, and point-of-sale systems.
- Online analytical processing (OLAP) workloads: These workloads involve complex queries and large-scale data processing, such as data mining, business intelligence, and decision support systems.
- Data warehousing workloads: These workloads involve collecting, storing, and analyzing large volumes of data from multiple sources, such as customer data, sales data, and financial data.
- Replication and disaster recovery workloads: These workloads involve replicating data between multiple databases or sites for backup and disaster recovery purposes.
- Database development and testing workloads: These workloads involve creating, modifying, and testing database schemas, queries, and applications.

What platforms and generations support NVMe VMs?

NVMe VMs are only accessible on the platform with the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake) processor. However, support for more platforms and generations is coming soon. Stay informed by following our product launch announcements in Azure updates.

The Azure region I need doesn't support NVMe, when will NVMe be available?

Watch out for our product launch announcements in the Azure updates.

What sizes in the Ebsv5 and Ebdsv5 family support NVMe?

The sizes E2-E112i support NVMe on Ebsv5 and Ebdsv5 families.

What sizes in the Ebsv5 and Ebdsv5 family support SCSI?

All sizes (E2-E96) on the Ebsv5 and Ebsdv5 families support SCSI except E112i.

I have a SCSI Ebsv5 VM. How do I switch to NVMe of the same VM size?

The steps to switch from SCSI to NVMe are the same as explained here.

How can I switch back to SCSI interface from NVMe VM?

To switch back to SCSI from NVMe, follow the same steps as explained here.

What is the price for NVMe Ebsv5 prices?

The NVMe enabled Ebsv5 and Ebdsv5 VMs are the same price as SCSI VMs. Refer to the pricing pages for Windows and Linux. With NVMe, you get higher performance at no extra cost.

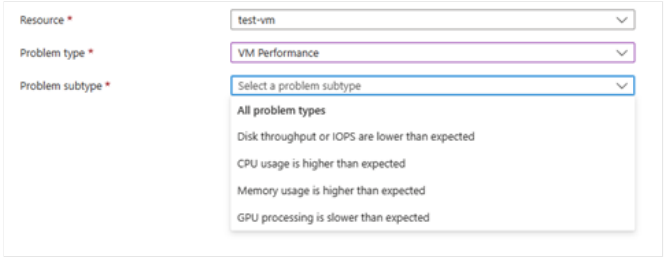
How can I try before purchasing this VM series? Is preview still available?

The preview period for this offer is over, and it is now generally available for purchase. You can request a quota for one of the available Azure regions to try out the new NVMe Ebsv5 or Ebdsv5 sizes.

Reporting Issues

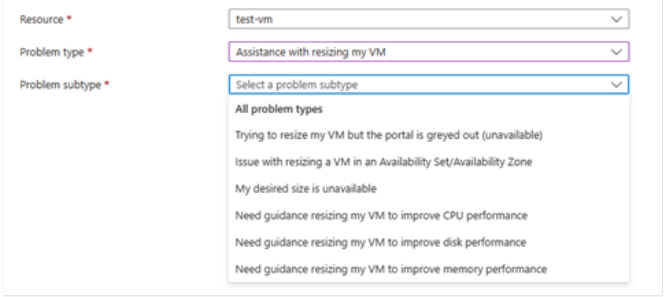
My VMs don't reach the published performance limits. Where do I report this issue?

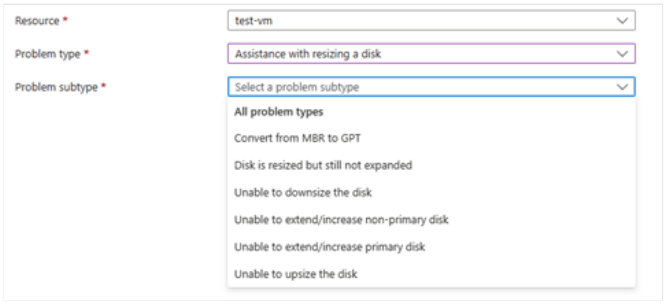
If you see performance issues, you can submit a support ticket. Provide all relevant information on the ticket, such as the subscription, VM size used, region, logs, and screenshot.



How can I get more help if I run into issues while setting up the VMs with the NVMe interface?

If you run into issues while creating or resizing Ebsv5 or Ebdsv5 to NVMe, and need assistance, you can submit a support ticket.





Other sizes and information

- General purpose
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Next steps

- Enabling NVMe Interface
- Enable NVMe FAQs
- Use the Azure Pricing Calculator

Last updated on 02/26/2026

'EC' sub-family memory optimized VM size series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'EC' sub-family of VM size series are one of Azure's security focused memory-optimized VM instances. They're designed for confidential computing with enhanced data protection and integrity, featuring various hardware-based Trusted Execution Environments (TEEs). These instances are ideal for memory-intensive workloads, such as large databases, big data analytics, and enterprise applications that require significant amounts of RAM to maintain high performance.

Workloads and use cases

Memory-Intensive Workloads: Any workload that demands a large memory footprint to efficiently handle tasks, such as simulations, large-scale computations in scientific research, or financial risk modeling.

Large Databases and SQL Servers: They are ideal for hosting large relational databases like SQL Server and NoSQL databases that benefit from high memory capacities for improved performance in data processing and transaction handling.

Enterprise Applications: Suitable for resource-intensive enterprise applications, including large-scale ERP and CRM systems, where the availability of ample memory is crucial for managing complex transactions and user loads.

Big Data Applications: Effective for big data analytics applications that need to process vast amounts of data in memory to speed up analysis and insights generation.

In-Memory Computing: Such as in-memory databases (e.g., SAP HANA) that require large amounts of RAM to keep the entire dataset in memory, allowing for ultra-fast data processing and query responses.

Data Warehousing: Provides the necessary resources for data warehousing solutions that handle and analyze large datasets, improving query performance and reducing response times.

Series in family

ECasv6 and ECadsv6-series

ECasv6-series

Azure's ECasv6-series virtual machines provide robust confidential computing capabilities, ensuring code and data remain secure during processing. Companies can migrate sensitive workloads to cloud infrastructure without modifying existing applications. The series runs on AMD's fourth-generation EPYC™ processors configured for multi-threaded operations with enhanced L3 cache architecture. AMD's SEV-SNP technology creates hardware-isolated environments that prevent unauthorized access from hypervisors, host management systems, and administrative users. This isolation protects virtual machines from neighboring VMs, hypervisor vulnerabilities, and both hardware and software-based security threats.

The platform includes hardware-based VM memory encryption alongside native support for confidential disk encryption feature. Organizations can implement OS disk encryption at boot through customer-managed keys (CMK) or platform-managed keys (PMK). This feature is fully integrated with Azure KeyVault and Azure Managed HSM ensuring FIPS 140-2 Level 3 compliance.

ECasv6 instances provide balanced memory-to-vCPU ratios suitable for production environments. Configurations scale up to 96 vCPUs paired with 672 GiB RAM and remote storage connectivity. These specifications support diverse workloads including e-commerce platforms, web services, desktop virtualization, secure databases, enterprise applications and more.

View the full ECasv6-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	2 - 96 vCPUs	AMD EPYC™ 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 34000 Mbps
Accelerators	None	

ECas_ccv5 and ECads_ccv5-series

Confidential child capable VMs allow you to borrow resources from the parent VM you deploy, to create AMD SEV-SNP protected child VMs. The parent VM has almost complete feature parity with any other general purpose Azure VM (for example, E-series VMs). This parent-child deployment model can help you achieve higher levels of isolation from the Azure host and parent VM. These confidential child capable VMs are built on the same hardware that powers our Azure confidential VMs. Azure confidential VMs are now generally available.

View the full ECas_ccv5 and ECads_ccv5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 96 vCores	AMD EPYC (Milan)
Memory	32 - 672 GiB	
Data Disks	4 - 32 Disks	6400 - 80000 IOPS / 144 - 1600 MBps
Network	2 - 8 NICs	
Accelerators	None	

Previous-generation EC family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 02/24/2026

ECasv5 and ECadsv5-series

Article • 12/03/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

The ECasv5-series and ECadsv5-series are confidential VMs for use in Confidential Computing.

These confidential VMs use AMD's third-Generation EPYC™ 7763v processor in a multi-threaded configuration with up to 256 MB L3 cache. This processor can achieve a boosted maximum frequency of 3.5 GHz. Both series offer Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP). SEV-SNP provides hardware-isolated VMs that protect data from other VMs, the hypervisor, and host management code. Confidential VMs offer hardware-based VM memory encryption. These series also offer OS disk pre-encryption before VM provisioning with different key management solutions.

These VM series also offer a combination of vCPUs and memory to meet the requirements of most memory-intensive enterprise applications.

ECasv5-series

ECasv5-series VMs offer a combination of vCPU and memory for memory-intensive enterprise applications. These VMs with no local disk provide a better value proposition for workloads where you don't need a local temp disk. For more information, see the FAQ for Azure VM sizes with no local temporary disk.

This series supports Standard SSD, Standard HDD, and Premium SSD disk types. Billing for disk storage and VMs is separate. To estimate your costs, use the Pricing Calculator.

> **Note**
>
> There are some pricing differences based on your encryption settings for confidential VMs.

ECasv5-series feature support

Supported features in ECasv5-series VMs:

- Premium Storage
- Premium Storage caching
- VM Generation 2

Unsupported features in ECasv5-series VMs:

- Live Migration

- Memory Preserving Updates
- Accelerated Networking
- Ephemeral OS Disks
- Nested Virtualization

ECasv5-series products

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max uncached disk throughput: IOPS/MBps	Max NICs	Max network bandwidth (Mbps)
Standard_EC2as_v5	2	16	Remote Storage Only	4	3750/48	2	3000
Standard_EC4as_v5	4	32	Remote Storage Only	8	6400/96	2	5000
Standard_EC8as_v5	8	64	Remote Storage Only	16	12800/192	4	5000
Standard_EC16as_v5	16	128	Remote Storage Only	32	25600/384	4	10000
Standard_EC20as_v5	20	160	Remote Storage Only	32	32000/480	8	10000
Standard_EC32as_v5	32	256	Remote Storage Only	32	51200/768	8	12500
Standard_EC48as_v5	48	384	Remote Storage Only	32	76800/1152	8	15000
Standard_EC64as_v5	64	512	Remote Storage Only	32	80000/1200	8	20000
Standard_EC96as_v5	96	672	Remote Storage Only	32	80000/1200	8	20000

ECadsv5-series

ECadsv5-series VMs offer a combination of vCPU, memory, and temporary storage for memory-intensive enterprise applications. These VMs offer local storage.

This series supports Standard SSD, Standard HDD, and Premium SSD disk types. Billing for disk storage and VMs is separate. To estimate your costs, use the Pricing Calculator.

> **Note**
>
> There are some pricing differences based on your encryption settings for confidential VMs.

ECadsv5-series feature support

Supported features in DCasv5-series VMs:

- Premium Storage
- Premium Storage caching
- VM Generation 2
- Ephemeral OS Disks

Unsupported features in DCasv5-series VMs:

- Live Migration
- Memory Preserving Updates
- Accelerated Networking

ECadsv5-series products

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max temp storage throughput: IOPS/MBps	Max uncached disk throughput: IOPS/MBps	Max NICs	Max network bandwidth (Mbps)
Standard_EC2ads_v5	2	16	75	4	9000 / 125	3750/48	2	3000
Standard_EC4ads_v5	4	32	150	8	19000 / 250	6400/96	2	5000
Standard_EC8ads_v5	8	64	300	16	38000 / 500	12800/192	4	5000
Standard_EC16ads_v5	16	128	600	32	75000 / 1000	25600/384	4	10000

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max temp storage throughput: IOPS/MBps	Max uncached disk throughput: IOPS/MBps	Max NICs	Max network bandwidth (Mbps)
Standard_EC20ads_v5	20	160	750	32	94000 / 1250	32000/480	8	10000
Standard_EC32ads_v5	32	256	1200	32	150000 / 2000	51200/768	8	12500
Standard_EC48ads_v5	48	384	1800	32	225000 / 3000	76800/1152	8	15000
Standard_EC64ads_v5	64	512	2400	32	300000 / 4000	80000/1200	8	20000
Standard_EC96ads_v5	96	672	3600	32	450000 / 4000	80000/1200	8	20000

> **Note**
>
> To achieve these IOPs, use Gen2 VMs.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- To learn how to get the best storage performance for your VMs, see Virtual machine and disk performance.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth.

  Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information

on optimizing network throughput, see Optimize network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Next steps

Confidential virtual machine options on AMD processors

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

ECasv6 sizes series

Summarize this article for me

Azure's ECasv6-series virtual machines provide robust confidential computing capabilities, ensuring code and data remain secure during processing. Companies can migrate sensitive workloads to cloud infrastructure without modifying existing applications. The series runs on AMD's fourth-generation EPYC™ processors configured for multi-threaded operations with enhanced L3 cache architecture. AMD's SEV-SNP technology creates hardware-isolated environments that prevent unauthorized access from hypervisors, host management systems, and administrative users. This isolation protects virtual machines from neighboring VMs, hypervisor vulnerabilities, and both hardware and software-based security threats.

The platform includes hardware-based VM memory encryption alongside native support for confidential disk encryption feature. Organizations can implement OS disk encryption at boot through customer-managed keys (CMK) or platform-managed keys (PMK). This feature is fully integrated with Azure KeyVault and Azure Managed HSM ensuring FIPS 140-2 Level 3 compliance.

ECasv6 instances provide balanced memory-to-vCPU ratios suitable for production environments. Configurations scale up to 96 vCPUs paired with 672 GiB RAM and remote storage connectivity. These specifications support diverse workloads including e-commerce platforms, web services, desktop virtualization, secure databases, enterprise applications and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC™ 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 34000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_EC2as_v6	2	16
Standard_EC4as_v6	4	32
Standard_EC8as_v6	8	64
Standard_EC16as_v6	16	128
Standard_EC32as_v6	32	256
Standard_EC48as_v6	48	384
Standard_EC64as_v6	64	512
Standard_EC96as_v6	96	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported

Feature name	Support status
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

ECadsv6 sizes series

Summarize this article for me

Azure's ECadsv6-series virtual machines provide robust confidential computing capabilities, ensuring code and data remain secure during processing. Companies can migrate sensitive workloads to cloud infrastructure without modifying existing applications. The series runs on AMD's fourth-generation EPYC™ processors configured for multi-threaded operations with enhanced L3 cache architecture. AMD's SEV-SNP technology creates hardware-isolated environments that prevent unauthorized access from hypervisors, host management systems, and administrative users. This isolation protects virtual machines from neighboring VMs, hypervisor vulnerabilities, and both hardware and software-based security threats.

The platform includes hardware-based VM memory encryption alongside native support for confidential disk encryption feature. Organizations can implement OS disk encryption at boot through customer-managed keys (CMK) or platform-managed keys (PMK). This feature is fully integrated with Azure KeyVault and Azure Managed HSM ensuring FIPS 140-2 Level 3 compliance.

ECadsv6 instances provide balanced memory-to-vCPU ratios suitable for production environments. Configurations scale up to 96 vCPUs paired with 672 GiB RAM and remote storage connectivity. These specifications support diverse workloads including e-commerce platforms, web services, desktop virtualization, secure databases, enterprise applications and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC™ 9004 (Genoa) [x86-64]
Memory	16 - 672 GiB	
Local Storage	None	
Remote Storage	8 - 64 Disks	4000 - 175000 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 34000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_EC2ads_v6	2	16
Standard_EC4ads_v6	4	32
Standard_EC8ads_v6	8	64
Standard_EC16ads_v6	16	128
Standard_EC32ads_v6	32	256
Standard_EC48ads_v6	48	384
Standard_EC64ads_v6	64	512
Standard_EC96ads_v6	96	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported

Feature name	Support status
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

ECesv6 series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs

The ECesv6-series are Azure confidential VMs that can be used to protect the confidentiality and integrity of your code and data while it's being processed in the public cloud. Organizations can use these VMs to seamlessly bring confidential workloads to the cloud without any code changes to the application.

These machines are powered by Intel® 5th Generation Xeon® Scalable processors reaching an all-core turbo clock speed of 3.0 GHz and Intel® Advanced Matrix Extensions (AMX) for AI acceleration.

Featuring Intel® Trust Domain Extensions (TDX), these VMs are hardened from the cloud virtualized environment by denying the hypervisor, other host management code and administrators access to the VM memory and state. It helps to protect VMs against a broad range of sophisticated hardware and software attacks.

These VMs have native support for confidential disk encryption meaning organizations can encrypt their VM disks at boot with either a customer-managed key (CMK), or platform-managed key (PMK). This feature is fully integrated with Azure KeyVault or Azure Managed HSM with validation for FIPS 140-2 Level 3.

> **Note**
>
> There are some pricing differences based on your encryption settings for confidential VMs.

ECesv6-series feature support

Supported features in ECesv6-series VMs:

- Premium Storage
- Premium Storage caching
- VM Generation 2

Unsupported features in ECesv6-series VMs:

- Live Migration
- Memory Preserving Updates
- Accelerated Networking

- Nested Virtualization

ECesv6-series

The ECesv6 VMs offer even higher memory to vCPU ratio and an all new VM size with up to 64 vCPUs and 512 GiB of RAM. These VMs are ideal for memory intensive applications, large relational database servers, business intelligence applications, and critical applications that process sensitive and regulated data.

This series supports Standard SSD, Standard HDD, and Premium SSD types. Billing for disk storage and VMs is separate. To estimate your costs, use the Pricing Calculator.

ECesv6-series specifications

Expand table

Size	vCPU	RAM (GiB)	Temp storage (SSD) GiB	Max data disks	Max temp disk throughput IOPS/MBps	Max uncached disk throughput IOPS/MBps	Max NICs	Max Network Bandwidth (Mbps)
Standard_EC2es_v6	2	16	RS*	8	N/A	3750/106	2	12500
Standard_EC4es_v6	4	32	RS*	12	N/A	6400/212	2	12500
Standard_EC8es_v6	8	64	RS*	24	N/A	12800/424	4	12500
Standard_EC16es_v6	16	128	RS*	48	N/A	25600/848	8	12500
Standard_EC32es_v6	32	256	RS*	64	N/A	51200/1696	8	16000
Standard_EC48es_v6	48	384	RS*	64	N/A	76800/2544	8	24000
Standard_EC64es_v6	64	512	RS*	64	N/A	80000/3392	8	30000

*RS: These VMs have support for remote storage only

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.

- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- To learn how to get the best storage performance for your VMs, see Virtual machine and disk performance.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth.

  Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see Optimize network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Next steps

Create a confidential VM in Azure Portal | Create a confidential VM in Azure CLI

Last updated on 02/24/2026

ECedsv6 series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs

The ECedsv6-series are Azure confidential VMs that can be used to protect the confidentiality and integrity of your code and data while it's being processed in the public cloud. Organizations can use these VMs to seamlessly bring confidential workloads to the cloud without any code changes to the application.

These machines are powered by Intel® 5th Generation Xeon® Scalable processors reaching an all-core turbo clock speed of 3.0 GHz and Intel® Advanced Matrix Extensions (AMX) for AI acceleration.

Featuring Intel® Trust Domain Extensions (TDX), these VMs are hardened from the cloud virtualized environment by denying the hypervisor, other host management code and administrators access to the VM memory and state. It helps to protect VMs against a broad range of sophisticated hardware and software attacks.

These VMs have native support for confidential disk encryption meaning organizations can encrypt their VM disks at boot with either a customer-managed key (CMK), or platform-managed key (PMK). This feature is fully integrated with Azure KeyVault or Azure Managed HSM with validation for FIPS 140-2 Level 3.

> **Note**
>
> There are some pricing differences based on your encryption settings for confidential VMs.

ECedsv6-series feature support

Supported features in ECedsv6-series VMs:

- Premium Storage
- Premium Storage caching
- VM Generation 2

Unsupported features in ECedsv6-series VMs:

- Live Migration
- Memory Preserving Updates
- Accelerated Networking

- Nested Virtualization

ECedsv6-series

The ECedsv6 VMs offer even higher memory to vCPU ratio and an all new VM size with up to 64 vCPUs and 512 GiB of RAM. These VMs are ideal for memory intensive applications, large relational database servers, business intelligence applications, and critical applications that process sensitive and regulated data.

This series supports Standard SSD, Standard HDD, and Premium SSD types. Billing for disk storage and VMs is separate. To estimate your costs, use the Pricing Calculator.

ECedsv6-series specifications

Expand table

Size	vCPU	RAM (GiB)	Temp storage (SSD) GiB	Max data disks	Max temp disk throughput IOPS/MBps	Max uncached disk throughput IOPS/MBps	Max NICs	Max Network Bandwidth (Mbps)
Standard_EC2eds_v6	2	16	110	8	N/A	3750/106	2	7000
Standard_EC4eds_v6	4	32	220	12	N/A	6400/212	2	7000
Standard_EC8eds_v6	8	64	440	24	N/A	12800/424	4	7000
Standard_EC16eds_v6	16	128	880	48	N/A	25600/848	8	7000
Standard_EC32eds_v6	32	256	1760	64	N/A	51200/1696	8	9000
Standard_EC48eds_v6	48	384	2640	64	N/A	76800/2544	8	12500
Standard_EC64eds_v6	64	512	3520	64	N/A	80000/3392	8	30000

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.

- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- To learn how to get the best storage performance for your VMs, see Virtual machine and disk performance.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth.

  Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see Optimize network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Next steps

Create a confidential VM in Azure Portal | Create a confidential VM in Azure CLI

Last updated on 02/24/2026

ECas_cc_v5 and ECads_cc_v5-series (Preview)

Article • 08/23/2024

Applies to: ✔ Linux VMs in Azure Kubernetes Service

> **Note**
>
> Preview Terms - These VM sizes are subject to the Supplemental Terms of Use for Microsoft Azure Previews.

> **Note**
>
> Confidential child capable VMs are currently enabled only through Azure Kubernetes Service (AKS) when you choose these VMs as your agent node sizes. If you wish to enable it outside AKS, please contact azconfidentialpm@microsoft.com.

Confidential child capable VMs allow you to borrow resources from the parent VM you deploy, to create AMD SEV-SNP protected child VMs. The parent VM has almost complete feature parity with any other general purpose Azure VM (for example, E-series VMs). This parent-child deployment model can help you achieve higher levels of isolation from the Azure host and parent VM. These confidential child capable VMs are built on the same hardware that powers our Azure confidential VMs. Azure confidential VMs are now generally available.

This series supports Standard SSD, Standard HDD, and Premium SSD disk types. Billing for disk storage and VMs is separate. To estimate your costs, use the Pricing Calculator. For more information on disk types, see What disk types are available in Azure?

> **Note**
>
> There are some pricing differences based on your encryption settings for nested confidential VMs.

ECas_cc_v5-series products

The ECas_cc_v5-series sizes offer a combination of vCPU and memory for most production workloads. The ECas_cc_v5-series sizes offer a combination of vCPU and memory that is ideal for memory-intensive enterprise applications. These new VMs with no local disk provide a better value proposition for workloads that do not require local temp disk.

Trusted Launch: Not Supported - we are working to support for this feature. For now - please choose Standard security type when using this VM size)
Premium Storage: Supported
Premium Storage caching: Supported
Live Migration: Not Supported
Memory Preserving Updates: Not Supported
VM Generation Support: Generation 2
Accelerated Networking: Supported
Ephemeral OS Disks: Supported
Nested Virtualization: Supported

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max uncached disk throughput: IOPS/MBps	Max NICs
Standard_EC4as_cc_v5	4	32	Remote Storage Only	8	6400/144	2
Standard_EC8as_cc_v5	8	64	Remote Storage Only	16	12800/200	4
Standard_EC16as_cc_v5	16	128	Remote Storage Only	32	25600/384	4
Standard_EC20as_cc_v5	20	160	Remote Storage Only	32	32000/480	8
Standard_EC32as_cc_v5	32	256	Remote Storage Only	32	51200/768	8
Standard_EC48as_cc_v5	48	384	Remote Storage Only	32	76800/1152	8

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max uncached disk throughput: IOPS/MBps	Max NICs
Standard_EC64as_cc_v5	64	512	Remote Storage Only	32	80000/1200	8
Standard_EC96as_cc_v5	96	672	Remote Storage Only	32	80000/1600	8

ECads_cc_v5-series products

The ECads_cc_v5-series sizes offer a combination of vCPU, memory and temporary storage for most production workloads. The ECads_cc_v5-series sizes offer a combination of vCPU, memory and temporary storage that is ideal for memory-intensive enterprise applications.

Trusted Launch: Not Supported - we are working to support for this feature. For now - please choose Standard security type when using this VM size)
Premium Storage: Supported
Premium Storage caching: Supported
Live Migration: Not Supported
Memory Preserving Updates: Not Supported
VM Generation Support: Generation 2
Accelerated Networking: Supported
Ephemeral OS Disks: Supported
Nested Virtualization: Supported

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max uncached disk throughput: IOPS/MBps	Max NICs
Standard_EC4ads_cc_v5	4	32	150	8	6400/144	2
Standard_EC8ads_cc_v5	8	64	300	16	12800/200	4
Standard_EC16ads_cc_v5	16	128	600	32	25600/384	4
Standard_EC20ads_cc_v5	20	160	750	32	32000/480	8

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max uncached disk throughput: IOPS/MBps	Max NICs
Standard_EC32ads_cc_v5	32	256	1200	32	51200/768	8
Standard_EC48ads_cc_v5	48	384	1800	32	76800/1152	8
Standard_EC64ads_cc_v5	64	512	2400	32	80000/1200	8
Standard_EC96ads_cc_v5	96	672	3600	32	80000/1600	8

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache mode is set to **None**.
- To learn how to get the best storage performance for your VMs, see Virtual machine and disk performance.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth.

  Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see Optimize network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Other sizes and information

- General purpose
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Next steps

Confidential virtual machine options on AMD processors

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

'M' family memory optimized VM size series

06/06/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'M' family of VM size series are one of Azure's ultra memory-optimized VM instances designed for extremely memory-intensive workloads, such as large in-memory databases, data warehousing, and high-performance computing (HPC). Equipped with substantial RAM capacities and high vCPU capabilities, the M-family VMs support applications and services that require massive amounts of memory and significant computational power. High resource allocation makes the M-family particularly well-suited for handling tasks such as heavy SQL Server and other RDBMS workloads, complex scientific simulations, real-time data processing, and large-scale enterprise resource planning (ERP) systems, ensuring peak performance for the most demanding data-centric applications.

Workloads and use cases

SQL Server workloads with high memory needs: The M-family is particularly effective for running SQL Server machines with high memory requirements, such as for online transaction processing (OLTP) or data analytics.

In-memory databases: The M-family is particularly effective for running in-memory databases that require large amounts of RAM like SQL Server, or SAP HANA.

Big data applications: The M-family is ideal for handling big data applications that need to process and analyze huge datasets in memory, improving performance and reducing the time to insights.

Data warehousing: M-family VMs provide the performance and memory needed for data warehousing applications, facilitating faster queries and better handling of large volumes of data.

Enterprise applications: The M-family supports large-scale enterprise applications, including ERP and CRM systems, which benefit from having more memory to manage larger datasets and more complex transactions efficiently.

Heavy workloads in virtualized environments: The M-family is well-equipped to handle heavy virtualized environments, offering substantial memory for hosting multiple virtual machines and applications on a single physical server.

Series in family

Mbsv3 and Mbdsv3-series

Mbsv3-series

The Memory-and-storage optimized Mbv3 VM series are based on the 4th generation Intel® Xeon® Scalable processors and deliver higher remote disk storage performance. These new VM sizes offer up to 650,000 IOPS and 10 GBps of remote disk storage throughput Premium SSD v2 and Ultra Disk, up to 4 TiB of RAM and up to 650,000 IOPS and 10 GBps throughput to remote disk storage with NVMe interface by using Ultra Disk and Premium SSD v2.

The increased remote storage performance of these VMs is ideal for storage throughput-intensive workloads such as relational databases and data analytics applications.

The resource allocation and high-performance capabilities of the Mbv3 VM series make them particularly well suited for SQL Server workloads with high memory needs, such as critical online transaction processing (OLTP), data analytics, in-memory databases, data warehousing, and the consolidation of SQL Server workloads for efficient management of multiple SQL Server instances.

View the full Mbsv3-series page

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 176 vCPUs	Intel Xeon 4th gen Scalable (Sapphire Rapids) [x86-64]
Memory	128 - 1536 GiB	
Local Storage	None	
Remote Storage	64 Disks	44,000 - 260,000 IOPS 1,000 - 6,000 MBps
Network	8 NICs	8,000 - 50,000 Mbps
Accelerators	None	

Msv3 and Mdsv3 Medium Memory series

Msv3-MM-series

The Msv3 Medium Memory(MM) Virtual Machine (VM) series, powered by 4th generation Intel® Xeon® Scalable processors, are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership (TCO) and improved resilience to failures compared to previous generation Mv2 VMs. The Msv3 MM offers VM sizes of up to 4TB of memory and 4,000 MBps throughput to remote storage and provides up to 25% networking performance improvements over previous generations.

View the full Msv3-MM-series page

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	12 - 176 vCPUs	Intel Xeon 4th Gen Scalable (Sapphire Rapids) [x86-64]
Memory	240 - 3892 GiB	
Local Storage	None	
Remote Storage	64 Disks	16,250 - 130,000 IOPS 390 - 4,000 MBps
Network	4 - 8 NICs	4,000 - 40,000 Mbps
Accelerators	None	

Msv3 and Mdsv3 High Memory series

Msv3-HM-series

The Msv3 High Memory (HM) Virtual Machine (VM) series are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership and improved resilience to failures compared to the previous generation, Mv2-series VMs. Msv3 HM offers VM sizes with memory ranging from 6TB to 16TB, up to 8,000 MBps throughout to remote storage, and up to 25% networking performance improvements over previous generations.

View the full Msv3-HM-series page

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	416 - 832 vCPUs	Intel Xeon 4th Gen Scalable (Sapphire Rapids) [x86-64]
Memory	5,696 - 15,200 GiB	
Local Storage	None	
Remote Storage	64 Disks	130,000 - 260,000 IOPS 4,000 - 8,000 MBps
Network	8 NICs	40,000 Mbps
Accelerators	None	

Mdsv3 Very High Memory series

The Mdsv3 Very High Memory (VHM) series, powered by 4th generation Intel® Xeon® Platinum 8490H (Sapphire Rapids) processor with an all core base frequency of 1.9 GHz and 3.5 GHz single core turbo frequency, are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership and improved resilience to failures compared to the previous generation. The Mv3 VHM sizes offer 32TB of memory, up to 8 GBps of throughput for remote storage, and provide up to 185 Gbps of networking performance with the current VHM generation.

View the full Mdsv3-VHM-series page

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	896 - 1,792 vCPUs	Intel Xeon Platinum 8490H (Sapphire Rapids) [x86-64]
Memory	30,400 GiB	
Local Storage	1 Disk	4,096 GiB
Remote Storage	64 Disks	110,000 IOPS 8,000 MBps
Network	8 NICs	185,000 Mbps
Accelerators	None	

Msv2 and Mdsv2 Medium Memory series

Msv2-series

The Msv2 Medium Memory VM Series features Intel® Xeon® Platinum 8280 (Cascade Lake) processor with an all core base frequency of 2.7 GHz and 4.0 GHz single core turbo frequency. With these VMs, customers achieve increased flexibility with local disk and diskless options. Customers also have access to a set of new isolated VM sizes with more CPU and memory that go up to 192 vCPU with 4 TiB of memory.

View the full Msv2-MM-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	32 - 192 vCPUs	Intel Xeon Platinum 8280 (Cascade Lake) [x86-64]
Memory	875 - 4096 GiB	
Local Storage	0 - Disks	
Remote Storage	32 - 64 Disks	20000 - 80000 IOPS 500 - 4000 MBps
Network	8 NICs	8000 - 30000 Mbps
Accelerators	None	

Mv2-series

The Mv2-series features high throughput, low latency platform running on a hyper-threaded Intel® Xeon® Platinum 8180M 2.5GHz (Skylake) processor with an all core base frequency of 2.5 GHz and a max turbo frequency of 3.8 GHz. All Mv2-series virtual machine sizes can use both standard and premium persistent disks. Mv2-series instances are memory optimized VM sizes providing unparalleled computational performance to support large in-memory databases and workloads, with a high memory-to-CPU ratio that is ideal for relational database servers, large caches, and in-memory analytics.

View the full Mv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	208 - 416 vCPUs	Intel Xeon Platinum 8180M (Skylake) [x86-64]
Memory	2,850 - 11,400 GiB	
Local Storage	1 Disk	4096 - 8192 GiB 80000 - 250000 IOPS (RR) 800 - 1600 MBps (RR)
Remote Storage	64 Disks	40000 - 80000 IOPS 1000 - 2000 MBps
Network	8 NICs	16000 - 32000 Mbps
Accelerators	None	

M-series

The M-series offers a high vCPU count (up to 128 vCPUs) and a large amount of memory (up to 3.8 TiB). It's also ideal for extremely large databases or other applications that benefit from high vCPU counts and large amounts of memory. M-series sizes are supported both on the Intel® Xeon® CPU E7-8890 v3 @ 2.50GHz and on the Intel® Xeon® Platinum 8280M (Cascade Lake).

View the full M-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 128 vCPUs	Intel Xeon E7-8890 v3 (Haswell) [x86-64] Intel Xeon Platinum 8280M (Cascade Lake) [x86-64]
Memory	218.75 - 3892 GiB	
Local Storage	1 Disk	256 - 14336 GiB 10000 - 250000 IOPS (RR) 100 - 1600 MBps (RR)
Remote Storage	8 - 64 Disks	5000/125 - 80000/2000 IOPS 5000/125 - 80000/2000 MBps
Network	4 - 8 NICs	2000 - 32000 Mbps
Accelerators	None	

Previous-generation M family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Mdsv3 Medium Memory sizes series

Summarize this article for me

The Mdsv3 Medium Memory(MM) Virtual Machine (VM) series, powered by 4th generation Intel® Xeon® Scalable processors, are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership (TCO) and improved resilience to failures compared to previous generation Mv2 VMs. The Mdsv3 MM offers VM sizes of up to 4TB of memory and 4,000 MBps throughput to remote storage and provides up to 25% networking performance improvements over previous generations.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	12 - 176 vCPUs	Intel Xeon 4th Gen Scalable (Sapphire Rapids) [x86-64]
Memory	240 - 3892 GiB	
Local Storage	1 Disk	400 GiB 10,000 - 160,000 IOPS (RR) 100 - 1600 MBps (RR)
Remote Storage Premium SSD	64 Disks	16,250 - 130,000 IOPS 390 - 4,000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	64 Disks	16,250 - 130,000 IOPS 390 - 4,000 MBps
Network	4 - 8 NICs	4,000 - 40,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M12ds_v3	12	240
Standard_M24ds_v3	24	480
Standard_M48ds_1_v3	48	974
Standard_M96ds_1_v3	96	974
Standard_M96ds_2_v3	96	1,946
Standard_M176ds_3_v3	176	2,794
Standard_M176ds_4_v3	176	3,892

VM Basics resources

- To run SQL Server on a VM with more than 64 vCores per NUMA node, disable SMT.
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported

Feature name	Support status
Nested Virtualization	Not Supported
Hibernation	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Msv3 Medium Memory sizes series

Summarize this article for me

The Msv3 Medium Memory(MM) Virtual Machine (VM) series, powered by 4th generation Intel® Xeon® Scalable processors, are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership (TCO) and improved resilience to failures compared to previous generation Mv2 VMs. The Msv3 MM offers VM sizes of up to 4TB of memory and 4,000 MBps throughput to remote storage and provides up to 25% networking performance improvements over previous generations.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	12 - 176 vCPUs	Intel Xeon 4th Gen Scalable (Sapphire Rapids) [x86-64]
Memory	240 - 3892 GiB	
Local Storage	None	
Remote Storage Premium SSD	64 Disks	16,250 - 130,000 IOPS 390 - 4,000 MBps
Remote Storage Premium SSD v2 / Ultra Disks	64 Disks	16,250 - 130,000 IOPS 390 - 4,000 MBps
Network	4 - 8 NICs	4,000 - 40,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M12s_v3	12	240
Standard_M24s_v3	24	480
Standard_M48s_1_v3	48	974
Standard_M96s_1_v3	96	974
Standard_M96s_2_v3	96	1,946
Standard_M176s_3_v3	176	2,794
Standard_M176s_4_v3	176	3,892

VM Basics resources

- To run SQL Server on a VM with more than 64 vCores per NUMA node, disable SMT.
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Msv2 Medium Memory sizes series

Summarize this article for me

The Msv2 Medium Memory VM Series features Intel® Xeon® Platinum 8280 (Cascade Lake) processor with an all core base frequency of 2.7 GHz and 4.0 GHz single core turbo frequency. With these VMs, customers achieve increased flexibility with local disk and diskless options. Customers also have access to a set of new isolated VM sizes with more CPU and memory that go up to 192 vCPU with 4 TiB of memory.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	32 - 192 vCPUs	Intel Xeon Platinum 8280 (Cascade Lake) [x86-64]
Memory	875 - 4096 GiB	
Local Storage	0 - Disks	
Remote Storage	32 - 64 Disks	20000 - 80000 IOPS 500 - 4000 MBps
Network	8 NICs	8000 - 30000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

> **Note**
>
> On March 31, 2027, all isolated sizes in this series are being retired.
>
> - Standard_M192is_v2
> - Standard_M192ims_v2

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M32ms_v2	32	875
Standard_M64s_v2	64	1,024
Standard_M64ms_v2	64	1,792
Standard_M128s_v2	128	2,048
Standard_M128ms_v2	128	3,892
Standard_M192is_v2	192	2,048
Standard_M192ims_v2	192	4,096

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Restricted Support
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported

Feature name	Support status
Nested Virtualization	Not Supported
Hibernation	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Mdsv2 Medium Memory sizes series

Summarize this article for me

The Mdsv2 Medium Memory VM Series features Intel® Xeon® Platinum 8280 (Cascade Lake) processor with an all core base frequency of 2.7 GHz and 4.0 GHz single core turbo frequency. With these VMs, customers achieve increased flexibility with local disk and diskless options. Customers also have access to a set of new isolated VM sizes with more CPU and memory that go up to 192 vCPU with 4 TiB of memory.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	32 - 192 vCPUs	Intel Xeon Platinum 8280 (Cascade Lake) [x86-64]
Memory	875 - 4096 GiB	
Local Storage	1 Disk	1024 - 4096 GiB 40000 - 160000 IOPS (RR) 400 - 4000 MBps (RR)
Remote Storage	32 - 64 Disks	20000 - 80000 IOPS 500 - 4000 MBps
Network	8 NICs	8000 - 30000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

> **Note**
>
> On March 31, 2027, all isolated sizes in this series are being retired. Standard_M192ids_v2 Standard_M192idms_v2

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M32dms_v2	32	875
Standard_M64ds_v2	64	1024
Standard_M64dms_v2	64	1792
Standard_M128ds_v2	128	2048
Standard_M128dms_v2	128	3892
Standard_M192ids_v2	192	2048
Standard_M192idms_v2	192	4096

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Restricted Support
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported

Feature name	Support status
Nested Virtualization	Not Supported
Hibernation	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

M sizes series

Summarize this article for me

The M-series offers a high vCPU count (up to 128 vCPUs) and a large amount of memory (up to 3.8 TiB). It's also ideal for extremely large databases or other applications that benefit from high vCPU counts and large amounts of memory. M-series sizes are supported both on the Intel® Xeon® CPU E7-8890 v3 @ 2.50GHz and on the Intel® Xeon® Platinum 8280M (Cascade Lake).

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 128 vCPUs	Intel Xeon E7-8890 v3 (Haswell) [x86-64] Intel Xeon Platinum 8280M (Cascade Lake) [x86-64]
Memory	218.75 - 3892 GiB	
Local Storage	1 Disk	256 - 14336 GiB 10000 - 250000 IOPS (RR) 100 - 1600 MBps (RR)
Remote Storage	8 - 64 Disks	5000/125 - 80000/2000 IOPS 5000/125 - 80000/2000 MBps
Network	4 - 8 NICs	2000 - 32000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M8ms	8	218.75
Standard_M16ms	16	437.5
Standard_M32ts	32	192
Standard_M32ls	32	256
Standard_M32ms	32	875
Standard_M64s	64	1,024
Standard_M64ls	64	512
Standard_M64ms	64	1,792
Standard_M128s	128	2,048
Standard_M128ms	128	3,892
Standard_M64	64	1,024
Standard_M64m	64	1,792
Standard_M128	128	2,048
Standard_M128m	128	3,892

VM Basics resources

- More than 64 vCPU's require one of these supported guest versions: Windows Server 2016, Ubuntu 18.04+ LTS, SLES 12 SP2+, Red Hat Enterprise Linux 7/8/9, CentOS 7.3+ or Oracle Linux 7.3+ with LIS 4.2.1 or higher.
- Instance is isolated to hardware dedicated to a single customer.
- Constrained core sizes are available for Standard_M16ms, Standard_M32ms, Standard_M64ms, and Standard_M128ms.
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported
Hibernation	Not Supported
Write Accelerator	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Msv3 High Memory sizes series

The Msv3 High Memory (HM) Virtual Machine (VM) series are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership and improved resilience to failures compared to the previous generation, Mv2-series VMs. Msv3 HM offers VM sizes with memory ranging from 6TB to 16TB, up to 8,000 MBps throughout to remote storage, and up to 25% networking performance improvements over previous generations.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	416 - 832 vCPUs	Intel Xeon 4th Gen Scalable (Sapphire Rapids) [x86-64]
Memory	5,696 - 15,200 GiB	
Local Storage	None	
Remote Storage	64 Disks	130,000 - 260,000 IOPS 4,000 - 8,000 MBps
Network	8 NICs	40,000 - 100,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Msv3 High Memory series (NVMe)

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M416s_6_v3	416	5,696
Standard_M416s_8_v3	416	7,600
Standard_M624s_12_v3	624	11,400
Standard_M832s_12_v3	832	11,400
Standard_M832is_16_v3	832	15,200

Msv3 High Memory series (SCSI)

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M416s_6_v3	416	5,696
Standard_M416s_8_v3	416	7,600
Standard_M624s_12_v3	624	11,400
Standard_M832s_12_v3	832	11,400
Standard_M832is_16_v3	832	15,200

VM Basics resources

- Disable SMT to run SQL Server on a VM with more than 64 vCores per NUMA node.
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported

Feature name	Support status
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported
Write Accelerator	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Mdsv3 High Memory sizes series

Summarize this article for me

The Mdsv3 High Memory (HM) Virtual Machine (VM) series are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership and improved resilience to failures compared to the previous generation, Mv2-series VMs. Mv3 HM offers VM sizes with memory ranging from 6TB to 16TB, up to 8,000 MBps throughout to remote storage, and up to 25% networking performance improvements over previous generations.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	416 - 832 vCPUs	Intel Xeon 4th Gen Scalable (Sapphire Rapids) [x86-64]
Memory	5,696 - 15,200 GiB	
Local Storage	1 Disk	400 GiB 250,000 IOPS (RR) 1,600 MBps (RR)
Remote Storage	64 Disks	130,000 - 260,000 IOPS 4,000 - 8,000 MBps
Network	8 NICs	40,000 - 100,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Mdsv3 High Memory series (NVMe)

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M416ds_6_v3	416	5,696
Standard_M416ds_8_v3	416	7,600
Standard_M624ds_12_v3	624	11,400
Standard_M832ds_12_v3	832	11,400
Standard_M832ids_16_v3	832	15,200

Mdsv3 High Memory series (SCSI)

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M416ds_6_v3	416	5,696
Standard_M416ds_8_v3	416	7,600
Standard_M624ds_12_v3	624	11,400
Standard_M832ds_12_v3	832	11,400
Standard_M832ids_16_v3	832	15,200

VM Basics resources

- Disable SMT to run SQL Server on a VM with more than 64 vCores per NUMA node.
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported

Feature name	Support status
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported
Write Accelerator	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Msv2 High Memory sizes series

The Msv2 High Memory series features high throughput, low latency platform running on a hyper-threaded Intel® Xeon® Platinum 8180M 2.5GHz (Skylake) processor with an all core base frequency of 2.5 GHz and a max turbo frequency of 3.8 GHz. All Msv2 High Memory series virtual machine sizes can use both standard and premium persistent disks. Msv2 High Memory series instances are memory optimized VM sizes providing unparalleled computational performance to support large in-memory databases and workloads, with a high memory-to-CPU ratio that is ideal for relational database servers, large caches, and in-memory analytics.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	208 - 416 vCPUs	Intel Xeon Platinum 8180M (Skylake) [x86-64]
Memory	2,850 - 11,400 GiB	
Local Storage	1 Disk	4096 - 8192 GiB 80000 - 250000 IOPS (RR) 800 - 1600 MBps (RR)
Remote Storage	64 Disks	40000 - 80000 IOPS 1000 - 2000 MBps
Network	8 NICs	16000 - 32000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M208s_v2	208	2,850
Standard_M208ms_v2	208	5,700
Standard_M416s_v2	416	5,700
Standard_M416s_8_v2	416	7,600
Standard_M416ms_v2	416	11,400

VM Basics resources

- Constrained core sizes are available for Standard_M416s_v2, and Standard_M416ms_v2.
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Restricted Support
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Mdsv3 Very High Memory sizes series

Summarize this article for me

The Mdsv3 Very High Memory (VHM) series, powered by 4th generation Intel® Xeon® Platinum 8490H (Sapphire Rapids) processor with an all core base frequency of 1.9 GHz and 3.5 GHz single core turbo frequency, are the next generation of memory-optimized VM sizes delivering faster performance, lower total cost of ownership and improved resilience to failures compared to the previous generation. The Mv3 VHM sizes offer 32TB of memory, up to 8 GBps of throughput for remote storage, and provide up to 185 Gbps of networking performance with the current VHM generation.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	896 - 1,792 vCPUs	Intel Xeon Platinum 8490H (Sapphire Rapids) [x86-64]
Memory	30,400 GiB	
Local Storage	1 Disk	4,096 GiB
Remote Storage	64 Disks	110,000 IOPS 8,000 MBps
Network	8 NICs	185,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M896ixds_32_v3	896	30,400
Standard_M896ixds_24_v3	896	23088
Standard_M1792ixds_32_v3	1,792	30,400

VM Basics resources

- Disable SMT to run SQL Server on a VM with more than 64 vCores per NUMA node.
- VHM VM Sizes are virtual machine sizes that are Isolated to a specific hardware type and dedicated to a single customer.
- The Standard_M896ixds_32_v3 VM is the Microsoft recommended VM type with 32TB to host S/4HANA workload. This VM type has Simultaneous Multithreading (SMT) disabled. With that complies with the SAP recommendations stated in SAP note #2711650 for the specific underlying hardware used in Azure to host this Virtual Machine (VM) type. With typical S/4HANA workload, tests this VM realized the best performance.
- The Standard_M1792ixds_32_v3 VM has Simultaneous Multithreading (SMT) enabled and is ideal for analytical workloads as documented in SAP note #2711650 for the specific underlying hardware used in Azure to host this VM type. Typical S/4HANA workloads may show performance regressions compared to the Standard_M896ixds_32_v3 VM type. It is acknowledged that S/4HANA customer workloads are varying and can be different in nature. As a result, there might be cases where S/4HANA customer workloads could eventually benefit. And the Standard_M1792ixds_32_v3 VM type could provide performance improvements compared to the Standard_M896ixds_32_v3 VM type for a customer specific S/4HANA workload. Evaluating and hosting S/4HANA workload on Standard_M1792ixds_32_v3 is in the customer's own responsibilities.
- It's also important to note that these VMs are compatible with only certain generation 2 Images. For a list of images that are compatible with the Mdsv3-series, please see below
  - Windows Server 2022 Datacenter Edition latest builds
  - SUSE Linux enterprise Server 15 SP4 and later
  - Red Hat Enterprise Linux 8.8 or later
  - Ubuntu 23.10 or later
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported
Write Accelerator	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Mbsv3 sizes series

Summarize this article for me

The Memory-and-storage optimized Mbv3 VM series are based on the 4th generation Intel® Xeon® Scalable processors and deliver higher remote disk storage performance. These new VM sizes offer up to 650,000 IOPS and 10 GBps of remote disk storage throughput Premium SSD v2 and Ultra Disk, up to 4 TiB of RAM and up to 650,000 IOPS and 10 GBps throughput to remote disk storage with NVMe interface by using Ultra Disk and Premium SSD v2.

The increased remote storage performance of these VMs is ideal for storage throughput-intensive workloads such as relational databases and data analytics applications.

The resource allocation and high-performance capabilities of the Mbv3 VM series make them particularly well suited for SQL Server workloads with high memory needs, such as critical online transaction processing (OLTP), data analytics, in-memory databases, data warehousing, and the consolidation of SQL Server workloads for efficient management of multiple SQL Server instances.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 416 vCPUs	Intel Xeon 4th gen Scalable (Sapphire Rapids) [x86-64]
Memory	128 - 3800 GiB	
Local Storage	None	
Remote Storage	64 Disks	44,000 - 650,000 IOPS 1,000 - 10,000 MBps
Network	8 NICs	8,000 - 50,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series (NVMe)

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M16bs_v3	16	128
Standard_M32bs_v3	32	256
Standard_M48bs_v3	48	384
Standard_M64bs_v3	64	512
Standard_M96bs_v3	96	768
Standard_M128bs_v3	128	1024
Standard_M176bs_v3	176	1536
Standard_M416bs_v3	416	3800

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Restricted Support
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported
Write Accelerator	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Mbdsv3 sizes series

Summarize this article for me

The Memory-and-storage optimized Mbdsv3 VM series are based on the 4th generation Intel® Xeon® Scalable processors and deliver higher remote disk storage performance. These new VM sizes offer up to 650,000 IOPS and 10 GBps of remote disk storage throughput Premium SSD v2 and Ultra Disk, up to 4 TiB of RAM and up to 650,000 IOPS and 10 GBps throughput to remote disk storage with NVMe interface by using Ultra Disk and Premium SSD v2.

The increased remote storage performance of these VMs is ideal for storage throughput-intensive workloads such as relational databases and data analytics applications.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 176 vCPUs	Intel Xeon 4th gen Scalable (Sapphire Rapids) [x86-64]
Memory	128 - 3892 GiB	
Local Storage	1 Disk	400 - 8000 GiB 10,000 - 160,000 IOPS (RR) 100 - 1600 MBps (RR)
Remote Storage	64 Disks	44,000 - 650,000 IOPS 1,000 - 8,000 MBps
Network	8 NICs	8,000 - 50,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series (NVMe)

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_M16bds_v3	16	128
Standard_M32bds_v3	32	256
Standard_M48bds_v3	48	384
Standard_M64bds_v3	64	512
Standard_M96bds_v3	96	768
Standard_M128bds_v3	128	1,024
Standard_M176bds_v3	176	1,536
Standard_M64bds_1_v3	64	1397
Standard_M96bds_2_v3	96	1946
Standard_M128bds_3_v3	128	2794
Standard_M176bds_4_v3	176	3892

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported

Feature name	Support status
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported
Write Accelerator	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Msv2 and Mdsv2 isolated sizes retirement

06/06/2025

On March 31, 2027, Azure will retire the Msv2 and Mdsv2-series Medium Memory virtual machines (VM) listed:

- Msv2 Medium Memory Diskless
  - Standard_M192is_v2
  - Standard_M192ims_v2
- Mdsv2 Medium Memory with Disk
  - Standard_M192ids_v2
  - Standard_M192idms_v2

From now to March 31, 2027, you can continue to use the VMs listed without disruption. On March 31, 2027, the remaining VMs with these specific sizes on your subscription will be set to a deallocated state. These VMs are stopped and removed from the host. These VMs won't be billed in the deallocated state. To avoid service disruptions, refer to the instructions listed to migrate to your selected Mv3 replacement size by March 31, 2027.

Migrate workloads to Mv3 Medium Memory Series VMs

The Msv3 and Mdsv3 Medium Memory Series, powered by 4th generation Intel® Xeon® Scalable processors, are the next generation of memory-optimized VM sizes delivers faster performance, lower total cost of ownership and improvement to failures, compared to previous generation Mv2 VMs. The Mv3 MM series offers VM sizes of up to 4TB of memory and 4,000 MBps throughout to remote storage and provides up to 25% networking performance improvements over previous generations.

Follow the instructions listed to migrate your M192i(d)(m)s VM to your chosen Msv3 and Mdsv3 Medium Memory Series replacement.

Migration steps

1. Choose a series and size for migration. Use the pricing calculator for further insights.
2. Get quota for the target VM series.
3. Check whether the existing M192i(d)(m)s_v2 VM is part of an Availability Set or Proximity Placement Group, which can be verified in Azure portal in the VM properties.

4. If the VM isn't part of an Availability Set nor a Proximity Placement Group, it can be resized.

5. If the VM isn't part of an Availability Set, but is part of a Proximity Placement Group, you need to shut down all VMs in the PPG, resize then restart the M192i(d)(m)s_v2 VMs, then restart the other VMs in the PPG.

6. If the VM is part of an Availability Set, check your records to determine if the Availability Set is pinned to a specific M-series compute cluster for proximity purposes (for example, to collocate with services like NFS that shares hosted on Azure NetApp Files). If you aren't sure about pinning having taken place, open an Azure Support Request to confirm.

7. If your VM is part of a non-pinned Availability Set, you need to shut down all VMs in that particular Availability Set and go through the resize procedure for every VM of the Availability Set.

8. If the VM is part of a non-pinned Availability set and a Proximity Placement Group, you have two options:

    - Shut down all VMs in the PPG, resize then restart the M192i(d)(m)s_v2 VMs, then restart the other VMs in the PPG.

    - Switch availability options to Availability Zone as described using the open source supported script and procedure linked or setting up Database Replication (for example, HANA System Replication, SQL Server AlwaysOn).
        - How to Migrate a Highly Available SAP System in Azure from Availability Set to Availability Zone

9. If your VM is part of a pinned Availability set or if you can't switch your availability option to Availability Zone (as recommended above), open an Azure Support request to assist with resizing.

Help and support

If you have questions, ask community experts in Microsoft Q&A. If you have a support plan and need technical help, create a support request:

In the Help + support page, select Create a support request. Follow the New support request page instructions. Use the following values:

1. For Issue type, select Technical.

2. For Service, select My services.

3. For Service type, select Virtual Machine running Windows/Linux or Virtual Machine running SAP in case you run SAP application workloads.
4. For Resource, select your VM.
5. For Problem type, select Assistance with resizing my VM.
6. For Problem subtype, select the option that applies to you.
7. Follow instructions in the Solutions and Details tabs, as applicable, and then Review + create.

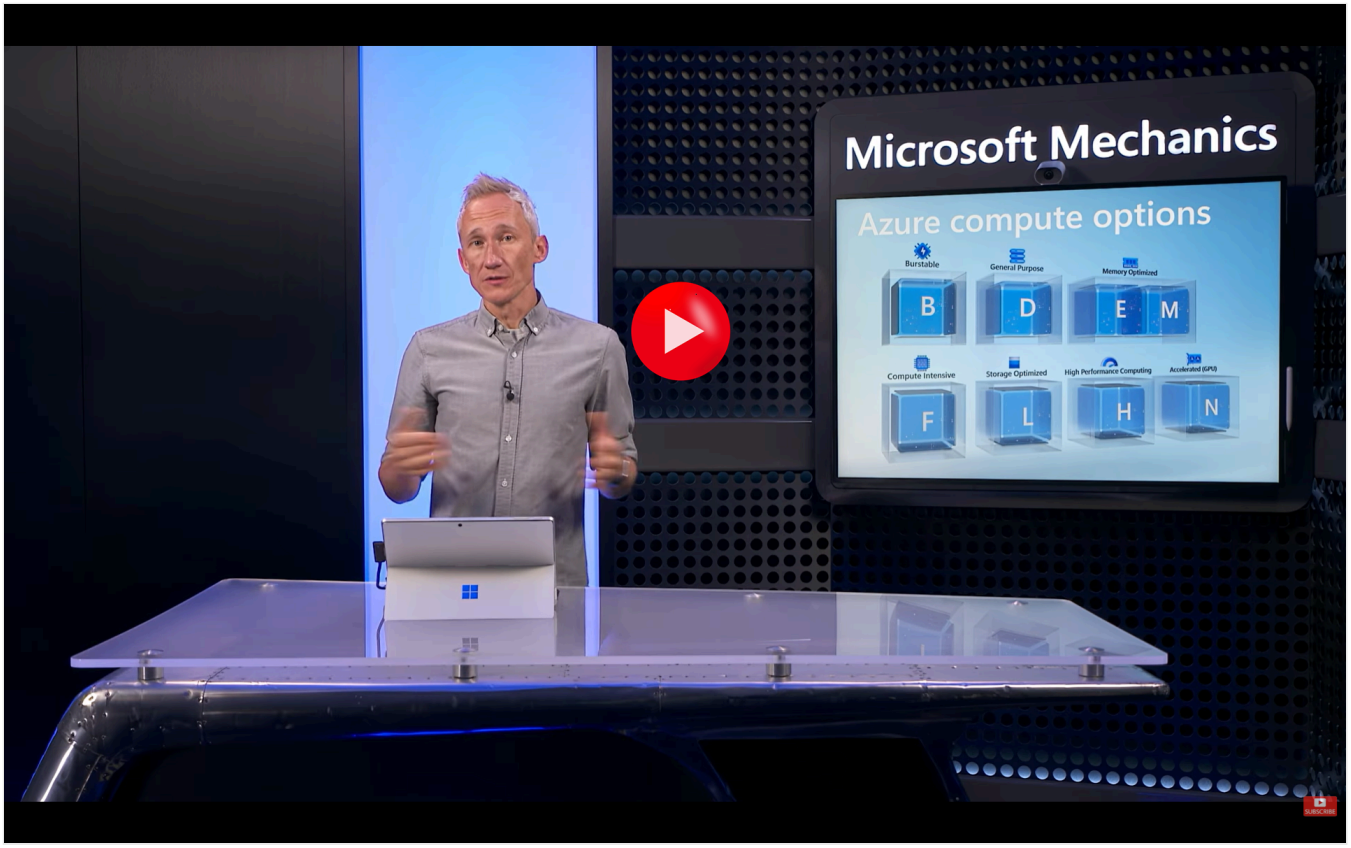
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

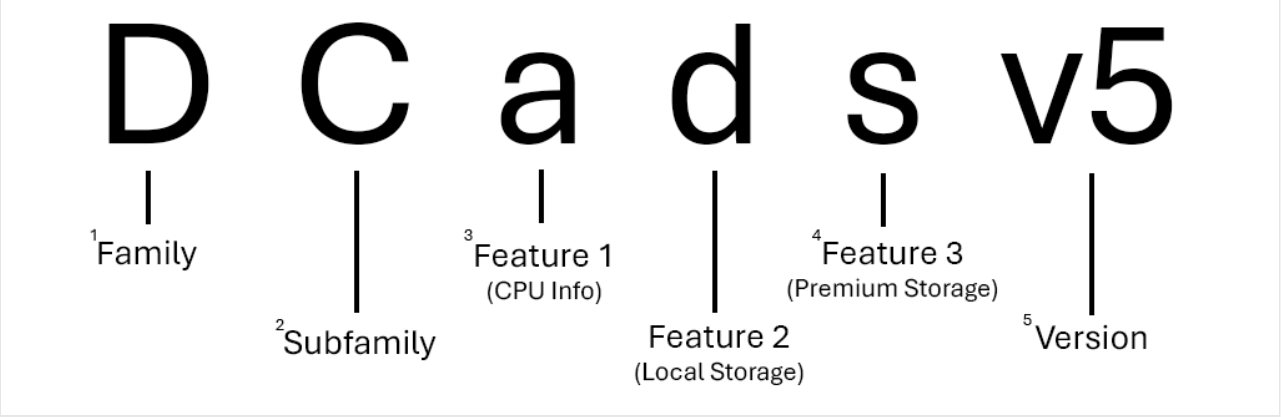
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

'L' family storage optimized VM size series

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'L' family of VM size series are one of Azure's storage-optimized VM instances. They're designed for workloads that require high disk throughput and I/O, such as databases, big data applications, and data warehousing. Equipped with high disk throughput and large local disk storage capacities, L-series VMs support applications and services that benefit from low latency and high sequential read and write speeds. This makes them particularly well-suited for handling tasks like large-scale log processing, real-time big data analytics, and scenarios involving large databases that perform frequent disk operations, ensuring efficient performance for storage-heavy applications.

Workloads and use cases

Big Data Applications: L-family VMs are perfect for big data applications that need to process, analyze, and manipulate large datasets stored directly on local disks, benefiting from the high I/O performance.

Database Servers: L-family VMs provide the necessary local disk performance for SQL Server, MySQL, PostgreSQL, and other database servers that benefit from fast access to disk storage.

File Servers: L-family VMs can be used effectively as file servers within a network, handling large files and serving them with high throughput, especially useful in environments with large media files.

Video Editing and Rendering: The high disk throughput and capacity of L-family VMs are beneficial for video editing and rendering tasks, where large video files are frequently read and written to disk.

Series in family

Lsv2-series

The Lsv2-series features high throughput, low latency, directly mapped local NVMe storage running on the AMD EPYC™ 7551 processor with an all core boost of 2.55GHz and a max boost of 3.0GHz. The Lsv2-series VMs come in sizes from 8 to 80 vCPU in a simultaneous multi-threading configuration. There is 8 GiB of memory per vCPU, and one 1.92TB NVMe SSD M.2 device per 8 vCPUs, with up to 19.2TB (10x1.92TB) available on the L80s v2.

View the full Lsv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 80 vCPUs	AMD EPYC 7551 (Naples) [x86-64]
Memory	64 - 640 GiB	
Local Storage	1 Temp Disk 1 - 10 NVMe Disks	80 - 800 GiB Temp Disks 1.92 TiB NVMe Disks
Remote Storage	16 - 32 Disks	8000 - 80000 IOPS 160 - 1400 MBps
Network	2 - 8 NICs	3200 - 16000+ Mbps
Accelerators	None	

Lsv3-series

The Lsv3-series of Azure Virtual Machines (Azure VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper-threaded configuration. This new processor features an all-core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. The Lsv3-series VMs are available in sizes from 8 to 80 vCPUs. There are 8 GiB of memory allocated per vCPU, and one 1.92TB NVMe SSD device allocated per 8 vCPUs, with up to 19.2TB (10x1.92TB) available on the L80s_v3 size.

> **Note**
>
> The Lsv3-series VMs are optimized to use the local disk on the node attached directly to the VM rather than using durable data disks. This method allows for greater IOPS and throughput for your workloads. The Lsv3, Lasv3, Lsv2, and Ls-

series VMs don't support the creation of a host cache to increase the IOPS achievable by durable data disks.

The high throughput and IOPS of the local disk makes the Lsv3-series VMs ideal for NoSQL stores such as Apache Cassandra and MongoDB. These stores replicate data across multiple VMs to achieve persistence in the event of the failure of a single VM.

View the full Lsv3-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 80 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64]
Memory	64 - 640 GiB	
Local Storage	1 Temp Disk 1 - 10 NVMe Disks	80 - 800 GiB Temp Disks 1.92 TiB NVMe Disks
Remote Storage	16 - 32 Disks	12800 - 80000 IOPS 290 - 2160 MBps
Network	4 - 8 NICs	12500 - 32000 Mbps
Accelerators	None	

Lasv3-series

The Lasv3-series of Azure Virtual Machines (Azure VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs run on an AMD 3rd Generation EPYC™ 7763v processor in a multi-threaded configuration with an L3 cache of up to 256 MB that can achieve a boosted maximum frequency of 3.5 GHz. The Lasv3-series VMs are available in sizes from 8 to 80 vCPUs in a simultaneous multi-threading configuration. There are 8 GiB of memory per vCPU, and one 1.92 TB NVMe SSD device per 8 vCPUs, with up to 19.2 TB (10x1.92TB) available on the L80as_v3 size.

Note

The Lasv3-series VMs are optimized to use the local disk on the node attached directly to the VM rather than using durable data disks. This method allows for greater IOPS and throughput for your workloads. The Lsv3, Lasv3, Lsv2, and Ls-

series don't support the creation of a local cache to increase the IOPS achievable by durable data disks.

The high throughput and IOPS of the local disk makes the Lasv3-series VMs ideal for NoSQL stores such as Apache Cassandra and MongoDB. These stores replicate data across multiple VMs to achieve persistence in the event of the failure of a single VM.

View the full Lasv3-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 80 vCPUs	AMD EPYC 7763v (Genoa) [x86-64]
Memory	64 - 640 GiB	
Local Storage	1 Temp Disk 1 - 10 NVMe Disks	80 - 800 GiB Temp Disks 1.92 TiB NVMe Disks
Remote Storage	16 - 32 Disks	12800 - 80000 IOPS 200 - 1400 MBps
Network	4 - 8 NICs	12500 - 32000 Mbps
Accelerators	None	

Previous-generation L family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Check out Azure Dedicated Hosts for physical servers able to host one or more virtual machines assigned to one Azure subscription.

Learn how to Monitor Azure virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Laosv4 sizes series

Summarize this article for me

The Laosv4-series of Azure Virtual Machines (VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs utilize AMD's fourth Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. The Laosv4-series VMs are available in sizes from 2 to 32 vCPUs, with 8 GiB of memory allocated per vCPU and 720GB of local NVMe temp disk capacity allocated per vCPU, with up to 23TB (12x1.92TB) of local temp disk capacity available on the L32aos_v4 size.

These VMs are perfect for distributed, scale-out workloads that require high amounts of local storage capacity per vCPU and the ability to move that data quickly over the network or to an Azure remote storage backend. Workloads like storage caching layers, Elasticsearch, distributed file systems, big data analytics, relational and NoSQL databases, and data warehouses all benefit from the dense storage capabilities of Laosv4 VMs.

Laosv4-series VMs support Standard SSD, Standard HDD, and Premium SSD remote disk types. You can also attach Ultra Disk storage based on its regional availability. Remote Disk storage is billed separately from virtual machines. For more information, see pricing for disks.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 32 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 256 GiB	
Local Storage	3 - 12 Disk	480 - 1,920 GiB 180,000 - 2,880,000 IOPS 1,100 - 17,600 MBps
Remote Storage	4 - 32 Disks	4,400 - 70,400 IOPS 150 - 2,400 MBps
Network	2 - 8 NICs	7,500 - 50,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size.

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_L2aos_v4	2	16
Standard_L4aos_v4	4	32
Standard_L8aos_v4	8	64
Standard_L12aos_v4	12	96
Standard_L16aos_v4	16	128
Standard_L24aos_v4	24	192
Standard_L32aos_v4	32	256

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported (except for L32aos_v4)
Live Migration	Not Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Lasv4 sizes series

Summarize this article for me

The Lasv4-series of Azure Virtual Machines (VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs utilize AMD's fourth Generation EPYC™ 9004 processors that can achieve a boosted maximum frequency of 3.7GHz. The Lasv4-series VMs are available in sizes from 2 to 96 vCPUs, with 8 GiB of memory allocated per vCPU and 240GB of local NVMe temp disk capacity allocated per vCPU, with up to 23TB (12x1.92TB) of local temp disk capacity available on the L96as_v4 size.

Lasv4-series VMs are well suited for scale-up or scale-out storage workloads that need a balance of SSD capacity, compute, and memory. These VMs are perfect for big data, relational, NoSQL databases, data analytics, and data warehousing workloads. Examples include Cassandra, MongoDB, Cloudera, Spark, Elastic Search, Redis, and other data-intensive applications.

Lasv4-series VMs support Standard SSD, Standard HDD, and Premium SSD remote disk types. You can also attach Ultra Disk storage based on its regional availability. Remote Disk storage is billed separately from virtual machines. For more information, see pricing for disks.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	AMD EPYC 9004 (Genoa) [x86-64]
Memory	16 - 768 GiB	
Local Storage	1 - 12 Disk	480 - 1920 GB 137500 - 6600000 IOPS 750 - 36000 MBps
Remote Storage	4 - 32 Disks	4000 - 172800 IOPS 90 - 4320 MBps
Network	2 - 8 NICs	12500 - 40000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size.

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_L2as_v4	2	16
Standard_L4as_v4	4	32
Standard_L8as_v4	8	64
Standard_L16as_v4	16	128
Standard_L32as_v4	32	256
Standard_L48as_v4	48	384
Standard_L64as_v4	64	512
Standard_L80as_v4	80	640
Standard_L96as_v4	96	768

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported (except for L96as_v4)
Live Migration	Not Supported
Memory Preserving Updates	Supported

Feature name	Support status
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Lsv4 sizes series

Summarize this article for me

The Lsv4-series of Azure Virtual Machines (VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs utilize the fifth Generation Intel® Xeon® Platinum 8573C (Emerald Rapids) processor reaching an all- core turbo clock speed of 3.0 GHz. The Lsv4-series VMs are available in sizes from 2 to 96 vCPUs, with 8 GiB of memory allocated per vCPU and 240GB of local NVMe temp disk capacity allocated per vCPU, with up to 23TB (12x1.92TB) of local temp disk capacity available on the L96s_v4 size.

Lsv4-series VMs are well suited for scale-up or scale-out storage workloads that need a balance of SSD capacity, compute, and memory. These VMs are perfect for big data, relational, NoSQL databases, data analytics, and data warehousing workloads. Examples include Cassandra, MongoDB, Cloudera, Spark, Elastic Search, Redis, and other data-intensive applications.

Lsv4-series VMs support Standard SSD, Standard HDD, and Premium SSD remote disk types. You can also attach Ultra Disk storage based on its regional availability. Remote Disk storage is billed separately from virtual machines.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 96 vCPUs	Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	16 - 768 GiB	
Local Storage	1 - 12 Disk	480 - 1920 GB 137500 - 6600000 IOPS 750 - 36000 MBps
Remote Storage	8 - 64 Disks	3,750 - 153,600 IOPS 106 - 5,088 MBps
Network	2 - 8 NICs	12,500 - 41,000 Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size.

Expand table

Size Name	vCPUs (Qty.)	Memory (GiB)
Standard_L2s_v4	2	16
Standard_L4s_v4	4	32
Standard_L8s_v4	8	64
Standard_L16s_v4	16	128
Standard_L32s_v4	32	256
Standard_L48s_v4	48	384
Standard_L64s_v4	64	512
Standard_L80s_v4	80	640
Standard_L96s_v4	96	768

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported (except for L96s_v4)
Live Migration	Not Supported
Memory Preserving Updates	Supported

Feature name	Support status
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

> **Note**
>
> This VM series will only work on OS images that support NVMe. If your current OS image doesn't have NVMe support, you'll see an error message. NVMe support is available on the most popular OS images, and we're continuously improving OS image compatibility.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Lsv3 sizes series

The Lsv3-series of Azure Virtual Machines (Azure VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake) processor in a hyper-threaded configuration. This new processor features an all-core turbo clock speed of 3.5 GHz with Intel® Turbo Boost Technology, Intel® Advanced-Vector Extensions 512 (Intel® AVX-512) and Intel® Deep Learning Boost. The Lsv3-series VMs are available in sizes from 8 to 80 vCPUs. There are 8 GiB of memory allocated per vCPU, and one 1.92TB NVMe SSD device allocated per 8 vCPUs, with up to 19.2TB (10x1.92TB) available on the L80s_v3 size.

> **Note**
>
> The Lsv3-series VMs are optimized to use the local disk on the node attached directly to the VM rather than using durable data disks. This method allows for greater IOPS and throughput for your workloads. The Lsv3, Lasv3, Lsv2, and Ls-series VMs don't support the creation of a host cache to increase the IOPS achievable by durable data disks.
>
> The high throughput and IOPS of the local disk makes the Lsv3-series VMs ideal for NoSQL stores such as Apache Cassandra and MongoDB. These stores replicate data across multiple VMs to achieve persistence in the event of the failure of a single VM.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 80 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64]
Memory	64 - 640 GiB	
Local Storage	1 Temp Disk 1 - 10 NVMe Disks	80 - 800 GiB Temp Disks 1.92 TiB NVMe Disks
Remote Storage	16 - 32 Disks	12800 - 80000 IOPS 290 - 2160 MBps
Network	4 - 8 NICs	12500 - 32000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_L8s_v3	8	64
Standard_L16s_v3	16	128
Standard_L32s_v3	32	256
Standard_L48s_v3	48	384
Standard_L64s_v3	64	512
Standard_L80s_v3	80	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Not Supported
Live Migration	Not Supported

Feature name	Support status
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Lasv3 sizes series

The Lasv3-series of Azure Virtual Machines (Azure VMs) features high-throughput, low latency, directly mapped local NVMe storage. These VMs run on an AMD 3rd Generation EPYC™ 7763v processor in a multi-threaded configuration with an L3 cache of up to 256 MB that can achieve a boosted maximum frequency of 3.5 GHz. The Lasv3-series VMs are available in sizes from 8 to 80 vCPUs in a simultaneous multi-threading configuration. There are 8 GiB of memory per vCPU, and one 1.92 TB NVMe SSD device per 8 vCPUs, with up to 19.2 TB (10x1.92TB) available on the L80as_v3 size.

> **Note**
>
> The Lasv3-series VMs are optimized to use the local disk on the node attached directly to the VM rather than using durable data disks. This method allows for greater IOPS and throughput for your workloads. The Lsv3, Lasv3, Lsv2, and Ls-series don't support the creation of a local cache to increase the IOPS achievable by durable data disks.
>
> The high throughput and IOPS of the local disk makes the Lasv3-series VMs ideal for NoSQL stores such as Apache Cassandra and MongoDB. These stores replicate data across multiple VMs to achieve persistence in the event of the failure of a single VM.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 80 vCPUs	AMD EPYC 7763v (Genoa) [x86-64]
Memory	64 - 640 GiB	
Local Storage	1 Temp Disk 1 - 10 NVMe Disks	80 - 800 GiB Temp Disks 1.92 TiB NVMe Disks
Remote Storage	16 - 32 Disks	12800 - 80000 IOPS 200 - 1400 MBps
Network	4 - 8 NICs	12500 - 32000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_L8as_v3	8	64
Standard_L16as_v3	16	128
Standard_L32as_v3	32	256
Standard_L48as_v3	48	384
Standard_L64as_v3	64	512
Standard_L80as_v3	80	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Not Supported
Live Migration	Not Supported

Feature name	Support status
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Optimize performance on Lsv3, Lasv3, and Lsv2-series Linux VMs

Article • 08/22/2024

> ⊗ **Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Uniform scale sets

Lsv3, Lasv3, and Lsv2-series Azure Virtual Machines (Azure VMs) support various workloads that need high I/O and throughput on local storage across a wide range of applications and industries. The L-series is ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases, including Cassandra, MongoDB, Cloudera, and Redis.

Several builds are available Azure Marketplace due to work with partners in Linux. These builds are optimized for Lsv3, Lasv3, and Lsv2-series performance. Available builds include the following and later versions of:

- Ubuntu 16.04
- RHEL 8.0 and clones, including CentOS, Rocky Linux, and Alma Linux
- Debian 9
- SUSE Linux 15
- Oracle Linux 8.0

This article provides tips and suggestions to ensure your workloads and applications achieve the maximum performance designed into the VMs.

AMD EPYC™ chipset architecture

Lasv3 and Lsv2-series VMs use AMD EPYC™ server processors based on the Zen micro-architecture. AMD developed Infinity Fabric (IF) for EPYC™ as scalable interconnect for its NUMA model that can be used for on-die, on-package, and multi-package communications. Compared with QPI (Quick-Path Interconnect) and UPI (Ultra-Path Interconnect) used on Intel modern monolithic-die processors, AMD's many-NUMA small-die architecture can bring both performance benefits and challenges. The actual

effects of memory bandwidth and latency constraints might vary depending on the type of workloads running.

Tips to maximize performance

- If you're uploading a custom Linux GuestOS for your workload, Accelerated Networking is turned off by default. If you intend to enable Accelerated Networking, enable it at the time of VM creation for best performance.
- To gain max performance, run multiple jobs with deep queue depth per device.
- Avoid mixing NVMe admin commands (for example, NVMe SMART info query, etc.) with NVMe I/O commands during active workloads. Lsv3, Lasv3, and Lsv2 NVMe devices are backed by Hyper-V NVMe Direct technology, which switches into “slow mode” whenever any NVMe admin commands are pending. Lsv3, Lasv3, and Lsv2 users might see a dramatic performance drop in NVMe I/O performance if that happens.
- Lsv2 users aren't recommended to rely on device NUMA information (all 0) reported from within the VM for data drives to decide the NUMA affinity for their apps. The recommended way for better performance is to spread workloads across CPUs if possible.
- The maximum supported queue depth per I/O queue pair for Lsv3, Lasv3, and Lsv2 VM NVMe device is 1024. Lsv3, Lasv3, and Lsv2 users are recommended to limit their (synthetic) benchmarking workloads to queue depth 1024 or lower to avoid triggering queue full conditions, which can reduce performance.
- The best performance is obtained when I/O is done directly to each of the raw NVMe devices with no partitioning, no file systems, no RAID config, etc. Before starting a testing session, ensure the configuration is in a known fresh/clean state by running `blkdiscard` on each of the NVMe devices. To obtain the most consistent performance during benchmarking, it's recommended to precondition the NVMe devices before testing by issuing random writes to all of the devices' LBAs twice as defined in the SNIA Solid State Storage Enterprise Performance Test Specification.

Utilizing local NVMe storage

Local storage on the 1.92 TB NVMe disk on all Lsv3, Lasv3, and Lsv2 VMs is ephemeral. During a successful standard reboot of the VM, the data on the local NVMe disk persists. The data doesn't persist on the NVMe if the VM is redeployed, deallocated, or deleted. Data doesn't persist if another issue causes the VM, or the hardware it's running on, to become unhealthy. When scenario happens, any data on the old host is securely erased.

There are also cases when the VM needs to be moved to a different host machine, for example, during a planned maintenance operation. Planned maintenance operations and some hardware failures can be anticipated with Scheduled Events. Use Scheduled Events to stay updated on any predicted maintenance and recovery operations.

In the case that a planned maintenance event requires the VM to be recreated on a new host with empty local disks, the data needs to be resynchronized (again, with any data on the old host being securely erased). This scenario occurs because Lsv3, Lasv3, and Lsv2-series VMs don't currently support live migration on the local NVMe disk.

There are two modes for planned maintenance.

Standard VM customer-controlled maintenance

- The VM is moved to an updated host during a 30-day window.
- Lsv3, Lasv3, and Lsv2 local storage data could be lost, so backing-up data prior to the event is recommended.

Automatic maintenance

- Occurs if the customer doesn't execute customer-controlled maintenance, or because of emergency procedures, such as a security zero-day event.
- Intended to preserve customer data, but there's a small risk of a VM freeze or reboot.
- Lsv3, Lasv3, and Lsv2 local storage data could be lost, so backing-up data prior to the event is recommended.

For any upcoming service events, use the controlled maintenance process to select a time most convenient to you for the update. Prior to the event, back up your data in premium storage. After the maintenance event completes, you can return your data to the refreshed Lsv3, Lasv3, and Lsv2 VMs local NVMe storage.

Scenarios that maintain data on local NVMe disks include:

- The VM is running and healthy.
- The VM is rebooted in place (by you or Azure).
- The VM is paused (stopped without deallocation).
- Most the planned maintenance servicing operations.

Scenarios that securely erase data to protect the customer include:

- The VM is redeployed, stopped (deallocated), or deleted (by you).

- The VM becomes unhealthy and has to service heal to another node due to a hardware issue.
- A few of the planned maintenance servicing operations that require the VM to be reallocated to another host for servicing.

Frequently asked questions

The following are frequently asked questions about these series.

How do I start deploying L-series VMs?

Much like any other VM, use the Portal, Azure CLI, or PowerShell to create a VM.

Does a single NVMe disk failure cause all VMs on the host to fail?

If a disk failure is detected on the hardware node, the hardware is in a failed state. When this problem occurs, all VMs on the node are automatically deallocated and moved to a healthy node. For Lsv3, Lasv3, and Lsv2-series VMs, this problem means that the customer's data on the failing node is also securely erased. The customer needs to recreate the data on the new node.

Do I need to change the blk_mq settings?

RHEL/CentOS 7.x automatically uses blk-mq for the NVMe devices. No configuration changes or settings are necessary.

Next steps

See specifications for all VMs optimized for storage performance on Azure

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Optimize performance on Lsv3, Lasv3, and Lsv2-series Windows VMs

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Uniform scale sets

Lsv3, Lasv3, and Lsv2-series Azure Virtual Machines (Azure VMs) support various workloads that need high I/O and throughput on local storage across a wide range of applications and industries. The L-series is ideal for Big Data, SQL, NoSQL databases, data warehousing and large transactional databases, including Cassandra, MongoDB, Cloudera, and Redis.

Lsv3, Lasv3, and Lsv2-series VMs are designed to work with the needs of Windows and Linux operating systems for better performance with hardware and the software.

Software and hardware tuning resulted in the optimized version of Windows Server 2019 Datacenter, released to the Azure Marketplace (and later versions), which support maximum performance on the NVMe devices in L-series VMs.

This article provides tips and suggestions to ensure your workloads and applications achieve the maximum performance designed into the VMs.

AMD EPYC™ chipset architecture

Lasv3 and Lsv2-series VMs use AMD EPYC™ server processors based on the Zen micro-architecture. AMD developed Infinity Fabric (IF) for EPYC™ as a scalable interconnect for its NUMA model that can be used for on-die, on-package, and multi-package communications. Compared with QPI (Quick-Path Interconnect) and UPI (Ultra-Path Interconnect), used on Intel modern monolithic-die processors, AMD's many-NUMA small-die architecture can bring both performance benefits and challenges. The actual effects of memory bandwidth and latency constraints can vary depending on the type of workloads.

Tips for maximizing performance

- To gain max performance, run multiple jobs with deep queue depth per device.
- Avoid mixing NVMe admin commands (for example, NVMe SMART info query) with NVMe I/O commands during active workloads. Lsv3, Lasv3, and Lsv2 NVMe devices are backed by Hyper-V NVMe Direct technology, which switches into "slow

mode" whenever any NVMe admin commands are pending. Lsv3, Lasv3, and Lsv2 users might see a dramatic performance drop in NVMe I/O performance if that scenario happens.

- It's not recommended for Lsv2 users to rely on device NUMA information (all 0) reported from within the VM for data drives to decide the NUMA affinity for their apps. For better performance, it's recommended to spread workloads across CPUs if possible.
- The maximum supported queue depth per I/O queue pair for Lsv3, Lasv3, and Lsv2 VM NVMe device is 1024. Lsv3, Lasv3, and Lsv2 users are recommended to limit their (synthetic) benchmarking workloads to queue depth 1024 or lower to avoid triggering queue full conditions, which can reduce performance.
- The best performance is obtained when I/O is done directly to each of the raw NVMe devices with no partitioning, no file systems, no RAID config, etc.

Utilizing local NVMe storage

Local storage on the 1.92 TB NVMe disk on all Lsv3, Lasv3, and Lsv2 VMs is ephemeral. During a successful standard reboot of the VM, the data on the local NVMe disk persists. The data doesn't persist on the NVMe if the VM is redeployed, deallocated, or deleted. Data doesn't persist if another issue causes the VM, or the hardware on which the VM is running, to become unhealthy. When this scenario happens, any data on the old host is securely erased.

There are also cases when the VM needs to be moved to a different host machine; for example, during a planned maintenance operation. Planned maintenance operations and some hardware failures can be anticipated with Scheduled Events. Use Scheduled Events to stay updated on any predicted maintenance and recovery operations.

In the case that a planned maintenance event requires the VM to be recreated on a new host with empty local disks, the data needs to be resynchronized (again, with any data on the old host being securely erased). This scenario occurs because Lsv3, Lasv3, and Lsv2-series VMs don't currently support live migration on the local NVMe disk.

There are two modes for planned maintenance: standard VM customer-controlled maintenance and automatic maintenance.

For any upcoming service events, use the controlled maintenance process to select a time most convenient to you for the update. Prior to the event, back up your data in premium storage. After the maintenance event completes, return your data to the refreshed Lsv2 VMs local NVMe storage.

Scenarios that maintain data on local NVMe disks include when:

- The VM is running and healthy.
- The VM is rebooted in place by you or by Azure.
- The VM is paused (stopped without deallocation).
- Most planned maintenance servicing operations.

Scenarios that securely erase data to protect the customer include when:

- The VM is redeployed, stopped (deallocated), or deleted by you.
- The VM becomes unhealthy and has to service heal to another node due to a hardware issue.
- A few the planned maintenance servicing operations that require the VM to be reallocated to another host for servicing.

Standard VM customer-controlled maintenance

In standard VM customer-controlled maintenance, the VM is moved to an updated host during a 30-day window.

Lsv3, Lasv3, and Lsv2 local storage data might be lost, so backing-up data prior to the event is recommended.

Automatic maintenance

Automatic maintenance occurs if the customer doesn't execute customer-controlled maintenance. Automatic maintenance can also occur because of emergency procedures, such as a security zero-day event.

This type of maintenance is intended to preserve customer data, but there's a small risk of a VM freeze or reboot.

Lsv3, Lasv3, and Lsv2 local storage data might be lost, so backing-up data prior to the event is recommended.

Frequently asked questions

The following are frequently asked questions about these series.

How do I start deploying L-series VMs?

Much like any other VM, create a VM using the Azure portal, through the Azure Command-Line Interface (Azure CLI), or through PowerShell.

Does a single NVMe disk failure cause all VMs on the host to fail?

If a disk failure is detected on the hardware node, the hardware is in a failed state. When this problem occurs, all VMs on the node are automatically deallocated and moved to a healthy node. For Lsv3, Lasv3, and Lsv2-series VMs, this scenario means that the customer's data on the failing node is also securely erased. The customer needs to recreate the data on the new node.

Do I need to make polling adjustments in Windows Server 2012 or Windows Server 2016?

NVMe polling is only available on Windows Server 2019 and later versions on Azure.

Can I switch back to a traditional interrupt service routine (ISR) model?

Lasv3, and Lsv2-series VMs are optimized for NVMe polling. Updates are continuously provided to improve polling performance.

Can I adjust the polling settings in Windows Server 2019 or later versions?

The polling settings aren't user adjustable.

Next steps

See specifications for all VMs optimized for storage performance on Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

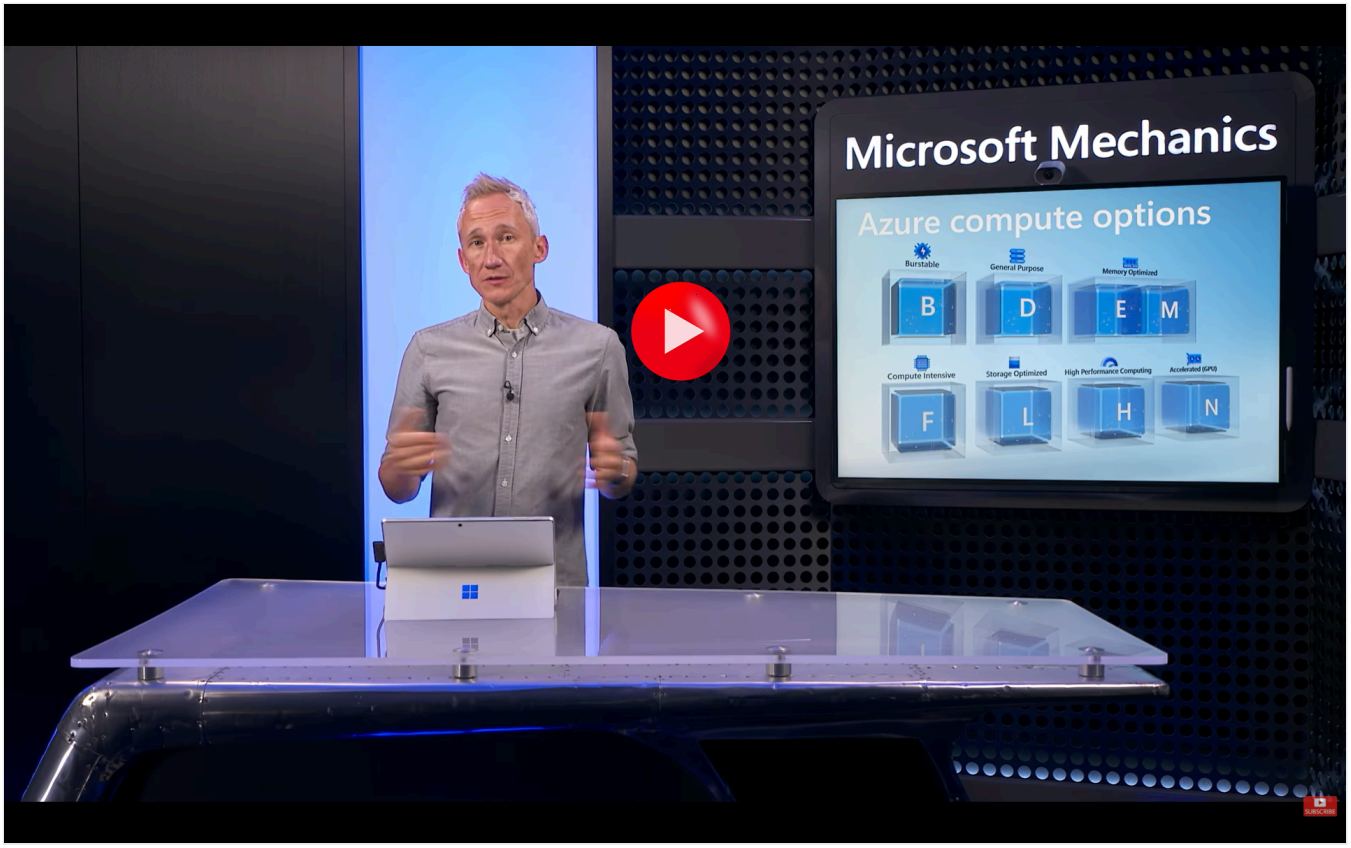
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

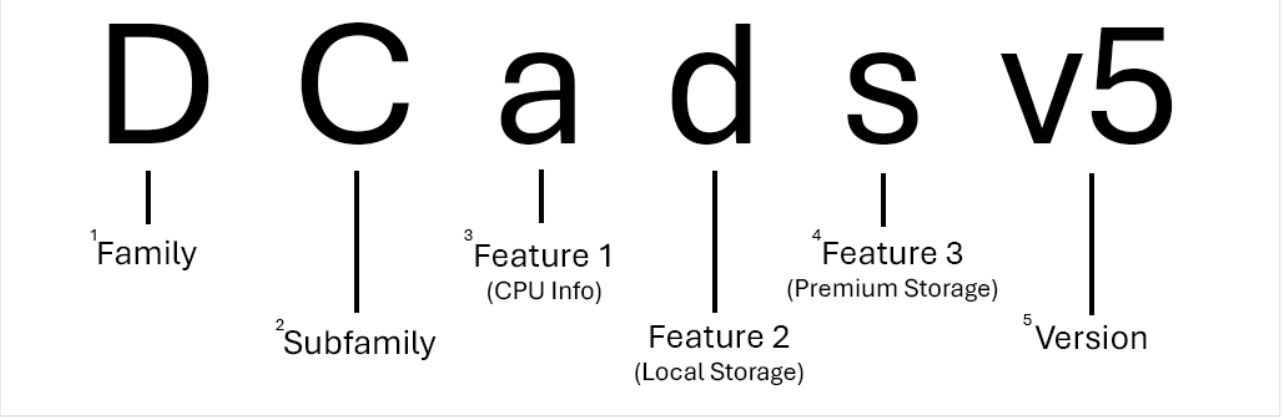
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

'NC' sub-family GPU accelerated VM size series

06/06/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'NC' sub-family of VM size series are one of Azure's GPU-optimized VM instances. They're designed for compute-intensive workloads, such as AI and machine learning model training, high-performance computing (HPC), and graphics-intensive applications. Equipped with powerful NVIDIA GPUs, NC-series VMs offer substantial acceleration for processes that require heavy computational power, including deep learning, scientific simulations, and 3D rendering. This makes them particularly well-suited for industries such as technology research, entertainment, and engineering, where rendering and processing speed are critical to productivity and innovation.

Workloads and use cases

AI and Machine Learning: NC-series VMs are ideal for training complex machine learning models and running AI applications. The NVIDIA GPUs provide significant acceleration for computations typically involved in deep learning and other intensive training tasks.

High-Performance Computing (HPC): These VMs are suitable for scientific simulations, rendering, and other HPC workloads that can be accelerated by GPUs. Fields like engineering, medical research, and financial modeling often use NC-series VMs to handle their computational needs efficiently.

Graphics Rendering: NC-series VMs are also used for graphics-intensive applications, including video editing, 3D rendering, and real-time graphics processing. They are particularly useful in industries such as game development and movie production.

Remote Visualization: For applications requiring high-end visualization capabilities, such as CAD and visual effects, NC-series VMs can provide the necessary GPU power remotely, allowing users to work on complex graphical tasks without needing powerful local hardware.

Simulation and Analysis: These VMs are also suitable for detailed simulations and analyses in areas like automotive crash testing, computational fluid dynamics, and weather modeling, where GPU capabilities can significantly speed up processing times.

Series in family

NCads_H100_v5-series

The NCads H100 v5 series virtual machines (VMs) are a new addition to the Azure GPU family. You can use this series for real-world Azure Applied AI training and batch inference workloads. The NCads H100 v5 series virtual machines are powered by NVIDIA H100 NVL GPU and 4th-generation AMD EPYC™ Genoa processors. The VMs feature up to 2 NVIDIA H100 NVL GPUs with 94GB memory each, up to 96 non-multithreaded AMD EPYC Genoa processor cores and 640 GiB of system memory. These VMs are ideal for real-world Applied AI workloads, such as:

- GPU-accelerated analytics and databases
- Batch inferencing with heavy pre- and post-processing
- Autonomy model training
- Oil and gas reservoir simulation
- Machine learning (ML) development
- Video processing
- AI/ML web services

View the full NCads_H100_v5-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	40 - 80 vCPUs	AMD EPYC (Genoa) [x86-64]
Memory	320 - 640 GiB	
Local Storage	1 Disk	3576 - 7152 GiB IOPS (RR) MBps (RR)
Remote Storage	8 - 16 Disks	100000 - 240000 IOPS 3000 - 7000 MBps
Network	2 - 4 NICs	40,000 - 80,000 Mbps
Accelerators	1 - 2 GPUs	Nvidia PCIe H100 GPU (94GB)

NCCads_H100_v5-series

The NCCads H100 v5 series of virtual machines are a new addition to the Azure GPU family. In this VM SKU, Trusted Execution Environment (TEE) spans confidential VM on the CPU and attached GPU, enabling secure offload of data, models, and computation to the GPU. The NCCads H100 v5 series is powered by 4th-generation AMD EPYC™ Genoa processors and

NVIDIA H100 Tensor Core GPU. These VMs feature 1 NVIDIA H100 NVL GPUs with 94 GB memory, 40 non-multithreaded AMD EPYC Genoa processor cores, and 320 GiB of system memory. These VMs are ideal for real-world Applied AI workloads, such as:

- GPU-accelerated analytics and databases
- Batch inferencing with heavy pre- and post-processing
- Machine Learning (ML) development
- Video processing
- AI/ML web services

View the full NCCads_H100_v5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	40 vCPUs	AMD EPYC (Genoa) [x86-64]
Memory	320 GiB	
Local Storage	1 Disk	800 GiB
Remote Storage	8 Disks	100000 IOPS 3000 MBps
Network	8 NICs	40000 Mbps
Accelerators	1 GPU	Nvidia H100 GPU (94GB)

NCv3-series

NCv3-series VMs are powered by NVIDIA Tesla V100 GPUs. These GPUs can provide 1.5x the computational performance of the NCv2-series. Customers can take advantage of these updated GPUs for traditional HPC workloads such as reservoir modeling, DNA sequencing, protein analysis, Monte Carlo simulations, and others. The NC24rs v3 configuration provides a low latency, high-throughput network interface optimized for tightly coupled parallel computing workloads. In addition to the GPUs, the NCv3-series VMs are also powered by Intel Xeon E5-2690 v4 (Broadwell) CPUs.

> **Important**
>
> For this VM series, the vCPU (core) quota in your subscription is initially set to 0 in each region. Request a vCPU quota increase for this series in an available region. These

SKUs aren't available to trial or Visual Studio Subscriber Azure subscriptions. Your subscription level might not support selecting or deploying these SKUs.

View the full NCv3-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 24 vCPUs	Intel Xeon E5-2690 v4 (Broadwell) [x86-64]
Memory	112 - 448 GiB	
Local Storage	1 Disk	736 - 2948 GiB IOPS (RR) MBps (RR)
Remote Storage	12 - 32 Disks	20000 - 80000 IOPS 200 - 800 MBps
Network	4 - 8 NICs	Mbps
Accelerators	1 - 4	Nvidia Tesla V100 GPU (16GB)

NCasT4_v3-series

The NCasT4_v3-series virtual machines are powered by Nvidia Tesla T4 GPUs and AMD EPYC 7V12(Rome) CPUs. The VMs feature up to 4 NVIDIA T4 GPUs with 16 GB of memory each, up to 64 non-multithreaded AMD EPYC 7V12 (Rome) processor cores(base frequency of 2.45 GHz, all-cores peak frequency of 3.1 GHz and single-core peak frequency of 3.3 GHz) and 440 GiB of system memory. These virtual machines are ideal for deploying AI services- such as real-time inferencing of user-generated requests, or for interactive graphics and visualization workloads using NVIDIA's GRID driver and virtual GPU technology. Standard GPU compute workloads based around CUDA, TensorRT, Caffe, ONNX and other frameworks, or GPU-accelerated graphical applications based on OpenGL and DirectX can be deployed economically, with close proximity to users, on the NCasT4_v3 series.

View the full NCasT4_v3-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 64 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	28 - 440 GiB	
Local Storage	1 Disk	176 - 2816 GiB IOPS (RR) MBps (RR)
Remote Storage	8 - 32 Disks	IOPS MBps
Network	2 - 8 NICs	8000 - 32000 Mbps
Accelerators	1 - 4 GPUs	Nvidia Tesla T4 GPU (16GB)

NC_A100_v4-series

The NC A100 v4 series virtual machine (VM) is a new addition to the Azure GPU family. You can use this series for real-world Azure Applied AI training and batch inference workloads. The NC A100 v4 series is powered by NVIDIA A100 PCIe GPU and third generation AMD EPYC™ 7V13 (Milan) processors. The VMs feature up to 4 NVIDIA A100 PCIe GPUs with 80 GB memory each, up to 96 non-multithreaded AMD EPYC Milan processor cores and 880 GiB of system memory. These VMs are ideal for real-world Applied AI workloads, such as:

GPU-accelerated analytics and databases Batch inferencing with heavy pre- and post-processing Autonomy model training Oil and gas reservoir simulation Machine learning (ML) development Video processing AI/ML web services

View the full NC_A100_v4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	24 - 96 vCPUs	AMD EPYC 7V13 (Milan) [x86-64]
Memory	220 - 880 GiB	
Local Storage	1 Temp Disk 1 - 4 NVMe Disks	64 - 256 GiB Temp Disk 960 GiB NVMe Disks
Remote Storage	8 - 32 Disks	30000 - 120000 IOPS 1000 - 4000 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - 8 NICs	20,000 - 80,000 Mbps
Accelerators	1 - 4 GPUs	Nvidia PCIe A100 GPU (80GB)

Previous-generation NC family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

NCads_H100_v5 sizes series

Summarize this article for me

The NCads H100 v5 series virtual machines (VMs) are a new addition to the Azure GPU family. You can use this series for real-world Azure Applied AI training and batch inference workloads. The NCads H100 v5 series virtual machines are powered by NVIDIA H100 NVL GPU and 4th-generation AMD EPYC™ Genoa processors. The VMs feature up to 2 NVIDIA H100 NVL GPUs with 94GB memory each, up to 96 non-multithreaded AMD EPYC Genoa processor cores and 640 GiB of system memory. These VMs are ideal for real-world Applied AI workloads, such as:

- GPU-accelerated analytics and databases
- Batch inferencing with heavy pre- and post-processing
- Autonomy model training
- Oil and gas reservoir simulation
- Machine learning (ML) development
- Video processing
- AI/ML web services

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	40 - 80 vCPUs	AMD EPYC (Genoa) [x86-64]
Memory	320 - 640 GiB	
Local Storage	1 Disk	3576 - 7152 GiB IOPS (RR) MBps (RR)
Remote Storage	8 - 16 Disks	100000 - 240000 IOPS 3000 - 7000 MBps
Network	2 - 4 NICs	40,000 - 80,000 Mbps
Accelerators	1 - 2 GPUs	Nvidia PCIe H100 GPU (94GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NC40ads_H100_v5	40	320
Standard_NC80adis_H100_v5	80	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

NCCads_H100_v5 sizes series

The NCCads H100 v5 series of virtual machines are a new addition to the Azure GPU family. In this VM SKU, Trusted Execution Environment (TEE) spans confidential VM on the CPU and attached GPU, enabling secure offload of data, models, and computation to the GPU. The NCCads H100 v5 series is powered by 4th-generation AMD EPYC™ Genoa processors and NVIDIA H100 Tensor Core GPU. These VMs feature 1 NVIDIA H100 NVL GPUs with 94 GB memory, 40 non-multithreaded AMD EPYC Genoa processor cores, and 320 GiB of system memory. These VMs are ideal for real-world Applied AI workloads, such as:

- GPU-accelerated analytics and databases
- Batch inferencing with heavy pre- and post-processing
- Machine Learning (ML) development
- Video processing
- AI/ML web services

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	40 vCPUs	AMD EPYC (Genoa) [x86-64]
Memory	320 GiB	
Local Storage	1 Disk	800 GiB
Remote Storage	8 Disks	100000 IOPS 3000 MBps
Network	8 NICs	40000 Mbps
Accelerators	1 GPU	Nvidia H100 GPU (94GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NCC40ads_H100_v5	40	320

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Creating VM

To get started with NCCads H100 v5 VMs and for information regarding confidential GPU driver and attesetation configuration, see confidential GPU Configuration

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Migrate your Standard_NC24rs_v3 virtual machine size by September 30, 2025

06/26/2025

> **Note**
>
> All the NCv3-series VM sizes, including Standard_NC24rs_v3, will be retired on September 30, 2025. For retirement details and SKU migration guidance, please refer to the NCv3-series retirement page rather than this page.

On September 30, 2025, Microsoft Azure will retire the Standard_NC24rs_v3 virtual machine (VM) size in NCv3-series virtual machines (VMs). To avoid any disruption to your service, we recommend that you change the VM sizing from the Standard_NC24rs_v3 to the newer VM series in the same NC product line.

Microsoft recommends the Azure NCadsH100_v5-series VMs, which offer greater GPU memory bandwidth per GPU, improved accelerated networking capabilities, and larger and faster local solid state drives. These VMs are targeted for GPU accelerated midrange AI training, batch inferencing, and high-performance computing simulation workloads.

However, depending on the workload being run, regional affinity, and cost preferences, other VMs that may be migrated to from Standard_NC24rs_v3 include NVadsA10_v5, NCasT4_v3, and NVadsV710_v5:

Expand table

Workload	Recommended SKU to Migrate to
High GPU compute workloads such as real-time inferencing and LLM inferencing.	NCadsH100_v5 which has the best inference per dollar value.
Offline inferencing workloads where latency is not the primary concern, and there is an interest in purchasing smaller VM SKUs or reducing costs.	NCasT4_v3
Graphics, visualization, or small AI workloads where optimal performance is not a priority.	NVadsA10_v5, NVadsV710_v5

How does the retirement of the Standard_NC24rs_v3 affect me?

After September 30, 2025, any remaining Standard_NC24rs_v3 VM subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. The Standard_NC24rs_v3 VM size will no longer be under SLA or have support included.

> **Note**
>
> This retirement announcement doesn't apply to NCasT4_v3, NC_A100_v4, and NCadsH100_v5 series virtual machines.

What action do I need to take before the retirement date?

You need to resize or deallocate your Standard_NC24rs_v3 VM size. We recommend that you change VM sizes for these workloads from the original Standard_NC24rs_v3 VM size to the NCadsH100_v5-series VMs (or an alternative).

The NCadsH100_v5-series is powered by NVIDIA H100 NVL GPU and 4th-generation AMD EPYC™ Genoa processors. The VMs feature up to 2 NVIDIA H100 NVL GPUs with 94GB memory each, up to 96 non-multithreaded AMD EPYC Genoa processor cores and 640 GiB of system memory. Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Refer to the above tables for Microsoft's recommendation. You can also file a support request if more assistance is needed.
2. Request quota for the new target VM).
3. Resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.
2. Under *Subscription*, select your subscription.
3. Under *Service*, click **My services**.
4. Under *Service type*, select **Virtual Machine running Windows/Linux**.
5. Under *Summary*, enter the summary of your request.
6. Under *Problem type*, select **Assistance with resizing my VM.**

7. Under *Problem subtype*, select the option that applies to you.

Migrate your NCv3-series virtual machines by September 30, 2025

08/29/2025

> **Note**
>
> If you're currently using the NCv3-series Azure VMs in either of these regions, you can disregard the general NCv3-series retirement date of September 30, 2025, and continue to use the NCv3-series Azure VMs in that region until the respective extension date:
>
> The US Central region has an extension and a retirement date of **February 28, 2026**.
>
> The West Europe region has an extension and a retirement date of **February 28, 2026**.
>
> The East US 2 region has an extension and a retirement date of **February 28, 2026**.
>
> The Canada Central region has an extension and a retirement date of **March 31, 2026**.
>
> The US Gov Virginia, US Gov Arizona, Jio India West, and China North 3 regions have an extension and a retirement date of **June 30, 2026**.

On September 30, 2025, Microsoft Azure will retire the Standard_NC6s_v3, Standard_NC12s_v3, Standard_NC24s_v3, and Standard_NC24rs_v3 virtual machines (VMs) in NCv3-series virtual machines (VMs). To avoid any disruption to your service, we recommend that you change the VM sizing for your workloads from the current NCv3-series VMs to the newer VM series in the same NC product line.

Microsoft is recommending the Azure NCadsH100_v5-series VMs, which offer greater GPU memory bandwidth per GPU, improved accelerated networking capabilities, and larger and faster local solid state drives. These VMs are targeted for GPU accelerated midrange AI training, batch inferencing, and high-performance computing simulation workloads. For regions where NCadsH100_v5 is not available, please migrate to the other recommended SKUs below.

Depending on the workload being run, regional affinity, and cost preferences, other VMs that may be migrated to from the NCv3-series VMs include NVadsA10_v5, NCasT4_v3, and NVadsV710_v5:

Expand table

Workload	Recommended SKU to Migrate to
High GPU compute workloads such as real-time inferencing and LLM inferencing.	NCadsH100_v5 which has the best inference per dollar value.
Offline inferencing workloads where latency is not the primary concern, and there is an interest in purchasing smaller VM SKUs or reducing costs.	NCasT4_v3
Graphics, visualization, or small AI workloads where optimal performance is not a priority.	NVadsA10_v5, NVadsV710_v5

How does the retirement of the NCv3-series virtual machines affect me?

After September 30th, any remaining NCv3-series virtual machines (VMs) subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. NCv3 will no longer be under SLA or have support included.

> **Note**
>
> This retirement only impacts the virtual machine sizes in the original NCv3-series powered by NVIDIA V100 GPUs. This retirement announcement doesn't apply to NCasT4_v3, NC_A100_v4, and NCadsH100_v5 series virtual machines.

What action do I need to take before the retirement date?

You need to resize or deallocate your NCv3-series VMs. We recommend that you change VM sizes for these workloads from the original NCv3-series VMs to the NCadsH100_v5-series VMs (or an alternative).

The NCadsH100_v5-series is powered by NVIDIA H100 NVL GPU and 4th-generation AMD EPYC™ Genoa processors. The VMs feature up to 2 NVIDIA H100 NVL GPUs with 94GB memory each, up to 96 non-multithreaded AMD EPYC Genoa processor cores and 640 GiB of system memory. Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Refer to the above tables for Microsoft's recommendation. You can also file a support request if more assistance is needed.
2. Request quota for the new target VM.
3. You can resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.
2. Under *Subscription*, select your subscription.
3. Under *Service*, click **My services**.
4. Under *Service type*, select **Virtual Machine running Windows/Linux**.
5. Under *Summary*, enter the summary of your request.
6. Under *Problem type*, select **Assistance with resizing my VM.**
7. Under *Problem subtype*, select the option that applies to you.

NCasT4_v3 sizes series

Summarize this article for me

The NCasT4_v3-series virtual machines(VM) are powered by NVIDIA Tesla T4 GPUs and AMD EPYC 7V12(Rome) CPUs. The VMs feature up to 4 NVIDIA T4 GPUs with 16 GB of memory each, up to 64 non-multithreaded AMD EPYC 7V12 (Rome) processor cores(base frequency of 2.45 GHz, all-cores peak frequency of 3.1 GHz and single-core peak frequency of 3.3 GHz) and 440 GiB of system memory. These VMs are ideal for deploying AI services- such as real-time inferencing of user-generated requests, or for interactive graphics and visualization workloads using NVIDIA's GRID driver and virtual GPU technology. Standard GPU compute workloads based around CUDA, TensorRT, Caffe, ONNX and other frameworks, or GPU-accelerated graphical applications based on OpenGL and DirectX can be deployed economically, with close proximity to users, on the NCasT4_v3 series.

On NCasT4_v3-series VMs, the Azure NVIDIA GPU driver extension installs CUDA drivers. For graphics and visualization workloads, manually install the Azure-supported GRID drivers.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 64 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	28 - 440 GiB	
Local Storage	1 Disk	176 - 2816 GiB IOPS (RR) MBps (RR)
Remote Storage	8 - 32 Disks	IOPS MBps
Network	2 - 8 NICs	8000 - 32000 Mbps
Accelerators	1 - 4 GPUs	Nvidia Tesla T4 GPU (16GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NC4as_T4_v3	4	28
Standard_NC8as_T4_v3	8	56
Standard_NC16as_T4_v3	16	110
Standard_NC64as_T4_v3	64	440

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

NC_A100_v4 sizes series

The NC A100 v4 series virtual machine (VM) is a new addition to the Azure GPU family. You can use this series for real-world Azure Applied AI training and batch inference workloads. The NC A100 v4 series is powered by NVIDIA A100 PCIe GPU and third generation AMD EPYC™ 7V13 (Milan) processors. The VMs feature up to 4 NVIDIA A100 PCIe GPUs with 80 GB memory each, up to 96 non-multithreaded AMD EPYC Milan processor cores and 880 GiB of system memory. These VMs are ideal for real-world Applied AI workloads, such as:

- GPU-accelerated analytics and databases
- Batch inferencing with heavy pre- and post-processing
- Autonomy model training
- Oil and gas reservoir simulation
- Machine learning (ML) development
- Video processing
- AI/ML web services

> **Note**
>
> We are currently only deploying net new capacity for the latest generation of the NC product line, the NCads_H100_v5-series.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	24 - 96 vCPUs	AMD EPYC 7V13 (Milan) [x86-64]
Memory	220 - 880 GiB	
Local Storage	1 Temp Disk 1 - 4 NVMe Disks	64 - 256 GiB Temp Disk 960 GiB NVMe Disks
Remote Storage	8 - 32 Disks	30000 - 120000 IOPS 1000 - 4000 MBps
Network	2 - 8 NICs	20,000 - 80,000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	1 - 4 GPUs	Nvidia PCIe A100 GPU (80GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NC24ads_A100_v4	24	220
Standard_NC48ads_A100_v4	48	440
Standard_NC96ads_A100_v4	96	880

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

NC RTX PRO 6000 Blackwell Server Edition v6 sizes series

These NC RTX PRO 6000 BSE v6 VMs are powered by NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs, each featuring 96 GB of ultra-fast GDDR7 memory and the latest Blackwell architecture to enable breakthrough performance in multimodal AI, physical AI, and high-fidelity graphics. The VM host is equipped with Intel Granite Rapids CPUs, providing all-core turbo frequency of up to 4.2 GHz to handle demanding pre- and post-processing steps efficiently. These VMs are ideal for converged AI and visual computing workloads, such as:

- Real-time digital twin and NVIDIA Omniverse simulation.
- LLM Inference and RAG (Retrieval-Augmented Generation) for models under 70B parameters.
- High-fidelity 3D rendering, product design, and video streaming.
- GPU-accelerated desktop virtualization (VDI) with NVIDIA RTX Virtual Workstation.
- Scientific visualization and High-Performance Computing (HPC) using FP32.
- Agentic AI application development and deployment.

Sign up to Participate in NCv6 Public Preview !

> **Note**
>
> - This VM series is currently in **Preview**.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	128 - 320 vCPUs	Intel Xeon Granite Rapids-AP [x86-64]
Memory	256 - 1280 GiB	High-speed DDR5

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	1 Disk	1024 - 2048 GiB
Remote Storage	16 Disks	
Network	6 - 8 NICs	75000 - 200000 Mbps
Accelerators	1 - 2	Nvidia RTX PRO 6000 Blackwell Server Edition GPU (96GB)

For features supported by this series, see the Feature support section.

General Purpose Sizes

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NC128ds_xl_RTXPRO6000BSE_v6	128	512
Standard_NC256ds_xl_RTXPRO6000BSE_v6	256	1024
Standard_NC320ds_xl_RTXPRO6000BSE_v6	320	1280

VM Basics resources

- Check vCPU quotas

Compute Optimized Sizes

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NC128lds_xl_RTXPRO6000BSE_v6	128	256
Standard_NC256lds_xl_RTXPRO6000BSE_v6	256	512
Standard_NC320lds_xl_RTXPRO6000BSE_v6	320	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Hibernation	Not Supported
Azure Disk Encryption for Linux VMs	Not Supported
Azure Disk Encryption for Windows VMs	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

'ND' sub-family GPU accelerated virtual machine size series

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'ND' family of VM size series are one of Azure's GPU-accelerated VM instances. They're designed for deep learning, AI research, and high-performance computing tasks that benefit from powerful GPU acceleration. Equipped with NVIDIA GPUs, ND-series VMs offer specialized capabilities for training and inference of complex machine learning models, facilitating faster computations and efficient handling of large datasets. This makes them particularly well-suited for academic and commercial applications in AI development and simulation, where cutting-edge GPU technology is crucial for achieving rapid and accurate results in neural network processing and other computationally intensive tasks.

Workloads and use cases

AI and Deep Learning: ND-family VMs are ideal for training and deploying complex deep learning models. Equipped with powerful NVIDIA GPUs, they provide the computational power necessary for handling extensive neural network training with large datasets, significantly reducing training times.

High-Performance Computing (HPC): ND-family VMs are suitable for HPC applications that require GPU acceleration. Fields such as scientific research, engineering simulations (e.g., computational fluid dynamics), and genomic processing can benefit from the high-throughput computing capabilities of ND-series VMs.

Graphics Rendering: ND-family's GPUs make them a great choice for graphics-intensive tasks, including real-time rendering for animation and video production, as well as high-fidelity simulations for virtual reality environments.

Remote Visualization: ND-family VMs can be used for remote visualization of data-intensive tasks, where high-end GPU capabilities are necessary to process and render complex visualizations over the cloud, facilitating access from less powerful client machines.

Series in family

ND-series V1

The ND-series virtual machines are a new addition to the GPU family designed for AI, and Deep Learning workloads. They offer excellent performance for training and inference. ND instances are powered by NVIDIA Tesla P40 GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. These instances provide excellent performance for single-precision floating point operations, for AI workloads utilizing Microsoft Cognitive Toolkit, TensorFlow, Caffe, and other frameworks. The ND-series also offers a much larger GPU memory size (24 GB), enabling to fit much larger neural net models. Like the NC-series, the ND-series offers a configuration with a secondary low-latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

View the full ND-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 24 vCPUs	Intel Xeon E5-2690 v4 (Broadwell) [x86-64]
Memory	112 - 448 GiB	
Local Storage	1 Disk	736 - 2948 GiB
Remote Storage	12 - 32 Disks	20000 - 80000 IOPS 200 - 800 MBps
Network	4 - 8 NICs	
Accelerators	1 - 4 GPUs	Nvidia Tesla P40 GPU (24GB)

NDv2-series

The NDv2-series virtual machine is a new addition to the GPU family designed for the needs of the most demanding GPU-accelerated AI, machine learning, simulation, and HPC workloads.

NDv2 is powered by 8 NVIDIA Tesla V100 NVLINK-connected GPUs, each with 32 GB of GPU memory. Each NDv2 VM also has 40 non-HyperThreaded Intel Xeon Platinum 8168 (Skylake) cores and 672 GiB of system memory.

NDv2 instances provide excellent performance for HPC and AI workloads utilizing CUDA GPU-optimized computation kernels, and the many AI, ML, and analytics tools that

support GPU acceleration 'out-of-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks.

Critically, the NDv2 is built for both computationally intense scale-up (harnessing 8 GPUs per VM) and scale-out (harnessing multiple VMs working together) workloads. The NDv2 series now supports 100-Gigabit InfiniBand EDR backend networking, similar to that available on the HB series of HPC VM, to allow high-performance clustering for parallel scenarios including distributed training for AI and ML. This backend network supports all major InfiniBand protocols, including those employed by NVIDIA's NCCL2 libraries, allowing for seamless clustering of GPUs.

View the full NDv2-series page

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	40 vCPUs	Intel Xeon Platinum 8168 (Skylake) [x86-64]
Memory	672 GiB	
Local Storage	1 Disk	2948 GiB
Remote Storage	32 Disks	80000 IOPS 800 MBps
Network	8 NICs	24000 Mbps
Accelerators	None	

ND_A100_v4-series

The ND A100 v4 series virtual machine(VM) is a new flagship addition to the Azure GPU family. These sizes are designed for high-end Deep Learning training and tightly coupled scale-up and scale-out HPC workloads.

The ND A100 v4 series starts with a single VM and eight NVIDIA Ampere A100 40GB Tensor Core GPUs. ND A100 v4-based deployments can scale up to thousands of GPUs with an 1.6 TB/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 200 GB/s NVIDIA Mellanox HDR InfiniBand connection. These connections are automatically configured between VMs occupying the same Azure Virtual Machine Scale Set, and support GPU Direct RDMA.

Each GPU features NVLINK 3.0 connectivity for communication within the VM with 96 physical 2nd-generation AMD Epyc™ 7V12 (Rome) CPU cores behind them.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration 'out-of-the-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on NVIDIA's NCCL2 communication libraries for seamless clustering of GPUs.

View the full ND_A100_v4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	96 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	900 GiB	
Local Storage	1 Disk	6000 GiB
Remote Storage	32 Disks	80000 IOPS 800 MBps
Network	8 NICs	24000 Mbps
Accelerators	8 GPUs	Nvidia PCIe A100 GPU (40GB)

NDm_A100_v4-series

The NDm A100 v4 series virtual machine(VM) is a new flagship addition to the Azure GPU family. These sizes are designed for high-end Deep Learning training and tightly coupled scale-up and scale-out HPC workloads.

The NDm A100 v4 series starts with a single VM and eight NVIDIA Ampere A100 80GB Tensor Core GPUs. NDm A100 v4-based deployments can scale up to thousands of GPUs with an 1.6 TB/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 200 GB/s NVIDIA Mellanox HDR InfiniBand connection. These connections are automatically configured between VMs occupying the same Azure Virtual Machine Scale Set, and support GPU Direct RDMA.

Each GPU features NVLINK 3.0 connectivity for communication within the VM with 96 physical 2nd-generation AMD Epyc™ 7V12 (Rome) CPU cores behind them.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration 'out-of-the-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on NVIDIA's NCCL2 communication libraries for seamless clustering of GPUs.

View the full NDm_A100_v4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	96 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	1900 GiB	
Local Storage	1 Disk	6400 GiB
Remote Storage	32 Disks	80000 IOPS 800 MBps
Network	8 NICs	24000 Mbps
Accelerators	8 GPUs	Nvidia PCIe A100 GPU (80GB)

ND_H100_v5-series

The ND H100 v5 series virtual machine (VM) is a new flagship addition to the Azure GPU family. This series is designed for high-end Deep Learning training and tightly coupled scale-up and scale-out Generative AI and HPC workloads.

The ND H100 v5 series starts with a single VM and eight NVIDIA H100 Tensor Core GPUs. ND H100 v5-based deployments can scale up to thousands of GPUs with 3.2 Tbps of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 400 Gb/s NVIDIA Quantum-2 CX7 InfiniBand connection. These connections are automatically configured between VMs occupying the same virtual machine scale set, and support GPU Direct RDMA.

Each GPU features NVLINK 4.0 connectivity for communication within the VM, and the instance has 96 physical fourth Gen Intel Xeon Scalable processor cores.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration 'out-of-the-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a

large set of existing AI and HPC tools that are built on NVIDIA's NCCL communication libraries for seamless clustering of GPUs.

View the full ND_H100_v5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	96 vCPUs	Intel Xeon (Sapphire Rapids) [x86-64]
Memory	1900 GiB	
Local Storage	1 Disk	28000 GiB
Remote Storage	32Disks	
Network	8 NICs	
Accelerators	8 GPUs	Nvidia PCIe H100 GPU (80GB)

ND_MI300X_v5-series

The ND MI300X v5 series virtual machine (VM) is a new flagship addition to the Azure GPU family. It was designed for high-end Deep Learning training and tightly coupled scale-up and scale-out Generative AI and HPC workloads.

The ND MI300X v5 series VM starts with eight AMD Instinct MI300 GPUs and two fourth Gen Intel Xeon Scalable processors for a total 96 physical cores. Each GPU within the VM is then connected to one another via 4th-Gen AMD Infinity Fabric links with 128 GB/s bandwidth per GPU and 896 GB/s aggregate bandwidth.

ND MI300X v5-based deployments can scale up to thousands of GPUs with 3.2 Tb/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 400 Gb/s NVIDIA Quantum-2 CX7 InfiniBand connection. These connections are automatically configured between VMs occupying the same virtual machine scale set, and support GPUDirect RDMA.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration "out-of-the-box," such as TensorFlow, Pytorch, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on AMD's ROCm Communication Collectives Library (RCCL) for seamless clustering of GPUs.

View the full ND_MI300X_v5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	96 vCPUs	Intel Xeon (Sapphire Rapids) [x86-64]
Memory	1850 GiB	
Local Storage	1 Disk	1000 GiB
Remote Storage	32 Disks	40800 IOPS 612 MBps
Network	8 NICs	
Accelerators	8 GPUs	AMD Instinct MI300X GPU (192GB)

Previous-generation ND family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Check out Azure Dedicated Hosts for physical servers able to host one or more virtual machines assigned to one Azure subscription.

Learn how to Monitor Azure virtual machines.

Feedback

Was this page helpful? Yes No

NDasrA100_v4 sizes series

The ND A100 v4 series virtual machine(VM) is a new flagship addition to the Azure GPU family. These sizes are designed for high-end Deep Learning training and tightly coupled scale-up and scale-out HPC workloads.

The ND A100 v4 series starts with a single VM and eight NVIDIA Ampere A100 40GB Tensor Core GPUs. ND A100 v4-based deployments can scale up to thousands of GPUs with an 1.6 TB/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 200 GB/s NVIDIA Mellanox HDR InfiniBand connection. These connections are automatically configured between VMs occupying the same Azure Virtual Machine Scale Set, and support GPU Direct RDMA.

Each GPU features NVLINK 3.0 connectivity for communication within the VM with 96 physical 2nd-generation AMD Epyc™ 7V12 (Rome) CPU cores behind them.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration 'out-of-the-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on NVIDIA's NCCL2 communication libraries for seamless clustering of GPUs.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	96 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	900 GiB	
Local Storage	1 Disk	6000 GiB
Remote Storage	32 Disks	80000 IOPS 800 MBps
Network	8 NICs	24000 Mbps
Accelerators	8 GPUs	Nvidia A100 GPU (40GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND96asr_v4	96	900

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

NDm_A100_v4 sizes series

Summarize this article for me

The NDm A100 v4 series virtual machine(VM) is a new flagship addition to the Azure GPU family. These sizes are designed for high-end Deep Learning training and tightly coupled scale-up and scale-out HPC workloads.

The NDm A100 v4 series starts with a single VM and eight NVIDIA Ampere A100 80GB Tensor Core GPUs. NDm A100 v4-based deployments can scale up to thousands of GPUs with an 1.6 TB/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 200 GB/s NVIDIA Mellanox HDR InfiniBand connection. These connections are automatically configured between VMs occupying the same Azure Virtual Machine Scale Set, and support GPU Direct RDMA.

Each GPU features NVLINK 3.0 connectivity for communication within the VM with 96 physical 2nd-generation AMD Epyc™ 7V12 (Rome) CPU cores behind them.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration 'out-of-the-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on NVIDIA's NCCL2 communication libraries for seamless clustering of GPUs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	96 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	1900 GiB	
Local Storage	1 Disk	6400 GiB
Remote Storage	32 Disks	80000 IOPS 800 MBps
Network	8 NICs	24000 Mbps
Accelerators	8 GPUs	Nvidia A100 GPU (80GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND96amsr_A100_v4	96	1900

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

ND sizes series

Summarize this article for me

> **Important**
>
> The ND-series has retired as of September 6th, 2023, and is no longer available. Please refer to the ND-series retirement page for more information.

The ND-series virtual machines are a new addition to the GPU family designed for AI, and Deep Learning workloads. They offer excellent performance for training and inference. ND instances are powered by NVIDIA Tesla P40 GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. These instances provide excellent performance for single-precision floating point operations, for AI workloads utilizing Microsoft Cognitive Toolkit, TensorFlow, Caffe, and other frameworks. The ND-series also offers a much larger GPU memory size (24 GB), enabling to fit much larger neural net models. Like the NC-series, the ND-series offers a configuration with a secondary low-latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 24 vCPUs	Intel Xeon E5-2690 v4 (Broadwell) [x86-64]
Memory	112 - 448 GiB	
Local Storage	1 Disk	736 - 2948 GiB
Remote Storage	12 - 32 Disks	20000 - 80000 IOPS 200 - 800 MBps
Network	4 - 8 NICs	
Accelerators	1 - 4 GPUs	Nvidia Tesla P40 GPU (24GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND6s	6	112
Standard_ND12s	12	224
Standard_ND24s	24	448
Standard_ND24rs	24	448

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Migrate your ND series virtual machines by September 6, 2023

03/17/2025

Based on feedback we've received from customers we're happy to announce that we're extending the retirement date by one year to 6 September 2023, for the Azure ND-Series virtual machine to give you more time to plan your migration.

As we continue to bring modern and optimized virtual machine instances to Azure leveraging the latest innovations in datacenter technologies, we thoughtfully plan how we retire aging hardware. With this in mind, we're retiring our ND GPU VM sizes, powered by NVIDIA Tesla P40 GPUs on 6 September 2023.

How does the ND-series migration affect me?

After 6 September 2023, any remaining ND size virtual machines remaining in your subscription will be set to a deallocated state. These virtual machines will be stopped and removed from the host. These virtual machines will no longer be billed in the deallocated state.

This VM size retirement only impacts the VM sizes in the ND-series. This retirement doesn't impact the newer NCv3, NC T4 v3, and ND v2 series virtual machines.

What actions should I take?

You'll need to resize or deallocate your ND virtual machines. We recommend moving your GPU workloads to another GPU Virtual Machine size. Learn more about migrating your workloads to another GPU Accelerated Virtual Machine size.

Next steps

Learn more about migrating your workloads to other GPU Azure Virtual Machine sizes.

If you have questions, contact us through customer support.

NDv2 sizes series

The NDv2-series virtual machine is a new addition to the GPU family designed for the needs of the most demanding GPU-accelerated AI, machine learning, simulation, and HPC workloads.

NDv2 is powered by 8 NVIDIA Tesla V100 NVLINK-connected GPUs, each with 32 GB of GPU memory. Each NDv2 VM also has 40 non-HyperThreaded Intel Xeon Platinum 8168 (Skylake) cores and 672 GiB of system memory.

NDv2 instances provide excellent performance for HPC and AI workloads utilizing CUDA GPU-optimized computation kernels, and the many AI, ML, and analytics tools that support GPU acceleration 'out-of-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks.

Critically, the NDv2 is built for both computationally intense scale-up (harnessing 8 GPUs per VM) and scale-out (harnessing multiple VMs working together) workloads. The NDv2 series now supports 100-Gigabit InfiniBand EDR backend networking, similar to that available on the HB series of HPC VM, to allow high-performance clustering for parallel scenarios including distributed training for AI and ML. This backend network supports all major InfiniBand protocols, including those employed by NVIDIA's NCCL2 libraries, allowing for seamless clustering of GPUs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	40 vCPUs	Intel Xeon Platinum 8168 (Skylake) [x86-64]
Memory	672 GiB	
Local Storage	1 Disk	2948 GiB
Remote Storage	32 Disks	80000 IOPS 800 MBps
Network	8 NICs	24000 Mbps
Accelerators	8	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND40rs_v2	40	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

ND GB200-v6 sizes series

The ND-GB200-v6 series virtual machine (VM) is a flagship addition to the Azure GPU family, delivering unmatched performance for Deep Learning training, Generative AI, and HPC workloads. These VMs leverage the NVIDIA GB200 Tensor Core GPUs, built on the Blackwell architecture, which offer significant advancements in computational power, memory bandwidth, and scalability over previous generations. Each ND-GB200-v6 VM is powered by two NVIDIA Grace CPUs and four NVIDIA Blackwell GPUs. The GPUs are interconnected via fifth-generation NVLink, providing a total of 4× 1.8 TB/s NVLink bandwidth per VM. This robust scale-up interconnect enables seamless, high-speed communication between GPUs within the VM. In addition, the VM offers a scale-out backend network with 4× 400 GB/s NVIDIA Quantum-2 CX7 InfiniBand connections per VM, ensuring high-throughput and low-latency communication when interconnecting multiple VMs. NVIDIA GB200 NVL72 connects up to 72 GPUs per rack, enabling system to operate as a single computer. This 72 GPU rack scale system comprised of groups of 18 ND GB200 v6 VMs delivering up to 1.4 Exa-FLOPS of FP4 Tensor Core throughput, 13.5 TB of shared high bandwidth memory, 130TB/s of cross sectional NVLINK bandwidth, and 28.8Tb/s scale-out networking.

With 128 vCPUs per VM supporting the overall system, the architecture is optimized to efficiently distribute workloads and memory demands for AI and scientific applications. This design enables seamless multi-GPU scaling and robust handling of large-scale models. These instances deliver best-in-class performance for AI, ML, and analytics workloads with out-of-the-box support for frameworks like TensorFlow, PyTorch, JAX, RAPIDS, and more. The scale-out InfiniBand interconnect is optimized for existing AI and HPC tools built on NVIDIA's NCCL communication libraries, ensuring efficient distributed computing across large clusters.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	128 vCPUs	Nvidia Grace CPU
Memory	900GB VM	LPDDR
Local Storage	4 Disks	16TB NVME Direct

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage	16 Disks	80000 IOPS/1200 MBps
Network	1 NICs	160Gb/s Ethernet
Accelerators	4 GPUs	Nvidia Blackwell GPU (192GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND128isr_NDR_GB200_v6	128	900

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

ND GB300-v6 sizes series

Summarize this article for me

The NDGB300v6 series is Azure's next-generation GPU VM line purpose-built for large-scale AI, especially high-throughput inference for reasoning and agentic systems. These VMs are powered by NVIDIA GB300 NVL72 rackscale systems (Blackwell Ultra GPUs + Grace CPUs). Each ND-GB300-v6 VM has two NVIDIA Grace CPUs and four NVIDIA Blackwell B300 GPUs. The GPUs are interconnected via fifth-generation NVLINK, providing a total of 4× 1.8 TB/s NVLINK bandwidth per VM. Each GPU also has 800 Gb/s InfiniBand connectivity, enabling cluster-scale performance with low latency. There are 18 VMs per rack, so effectively 72 NVIDIA Blackwell Ultra GPUs with 36 NVIDIA Grace CPUs, exposing ~37 TB of fast memory (~20TB High Bandwidth Memory, ~17TB of CPU Memory), 130 TB/s of intrarack NVLink bandwidth.

At rack scale, ND GB300 v6 delivers up to 1.44 exaFLOPS FP4 Tensor Core performance per NVL72 domain and has demonstrated ~1.1 million tokens/second LLM inference throughput per rack, which is ~27% higher than ND GB200 v6. See here for more details. Compared to GB200, GB300 provides ~1.5× FP4 compute, 50% higher HBM capacity (288 GB HBM3E per GPU vs. 192 GB), and 2x the back-end network bandwidth per GPU (800Gbps vs 400 Gbps).

The ND GB300 v6 architecture builds on the ND v6 GB200 to efficiently distribute workloads and memory demands across multiple GPUs, driving markedly higher inference throughput for long context and multimodal workloads AI and scientific applications. These instances deliver best-in-class performance for AI, ML, and analytics workloads with out-of-the-box support for frameworks like PyTorch, Tensorflow, JAX, RAPIDS, and more.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	128 vCPUs	Nvidia Grace CPU
Memory	864 VM	LPDDR
Local Storage	4 Disks	16TB NVME Direct
Remote Storage	16 Disks	80000 IOPS/1200 MBps
Network	1 NICs	160Gb/s Ethernet
Accelerators	4 GPUs	Nvidia Blackwell Ultra GPU (288GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND128isr_GB300_v6	128	864

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

ND-H100-v5 sizes series

The ND H100 v5 series virtual machine (VM) is a new flagship addition to the Azure GPU family. This series is designed for high-end Deep Learning training and tightly coupled scale-up and scale-out Generative AI and HPC workloads.

The ND H100 v5 series starts with a single VM and eight NVIDIA H100 Tensor Core GPUs. ND H100 v5-based deployments can scale up to thousands of GPUs with 3.2 Tbps of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 400 Gb/s NVIDIA Quantum-2 CX7 InfiniBand connection. These connections are automatically configured between VMs occupying the same virtual machine scale set, and support GPU Direct RDMA.

Each GPU features NVLINK 4.0 connectivity for communication within the VM, and the instance has 96 physical fourth Gen Intel Xeon Scalable processor cores.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration 'out-of-the-box,' such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on NVIDIA's NCCL communication libraries for seamless clustering of GPUs.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	96 vCPUs	Intel Xeon (Sapphire Rapids) [x86-64]
Memory	1900 GiB	
Local Storage	1 Disk	1024 GiB
Remote Storage	32Disks	
Network	8 NICs	
Accelerators	8 GPUs	Nvidia H100 GPU (80GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND96isr_H100_v5	96	1900

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported
Infiniband	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

ND-H200-v5 size series

The ND v5 H200 series virtual machine (VM) is designed to deliver exceptional performance for AI and high-performance computing (HPC) workloads. These VMs leverage the power of the NVIDIA H200 Tensor Core GPU, which offer a 76% increase in High Bandwidth Memory over the H100 GPUs to deliver higher performance on state-of-the-art Generative AI models. With 141 GB of high-speed memory, and 4.8 TB/s of memory bandwidth, the H200 GPU can handle larger datasets and more complex models, making it ideal for generative AI and scientific computing.

The ND H200 v5 series starts with a single VM and eight NVIDIA H200 Tensor Core GPUs, interconnected with 900 GB/s NVLink. ND H200 v5-based deployments can scale up to thousands of GPUs with 3.2Tb/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 400 Gb/s NVIDIA Quantum-2 CX7 InfiniBand connection. These connections are automatically configured between VMs occupying the same virtual machine scale set, and support GPUDirect RDMA.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration "out-of-the-box" such as TensorFlow, Pytorch, Caffe, RAPIDS, and other frameworks. Additionally, the scale-out InfiniBand interconnect is supported by a large set of existing AI and HPC tools that are built on NVIDIA's NCCL communication libraries for seamless clustering of GPUs.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	96 vCPUs	Intel Xeon (Sapphire Rapids) [x86-64]
Memory	1850 GiB	
Local Storage	1 Disk	1024 GiB
Remote Storage	16Disks	
Network	8 NICs	
Accelerators	8 GPUs	Nvidia H200 GPU (141GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND96isr_H200_v5	96	1850

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

ND-MI300X-v5 sizes series

The ND MI300X v5 series virtual machine (VM) is a new flagship addition to the Azure GPU family. It was designed for high-end Deep Learning training and tightly coupled scale-up and scale-out Generative AI and HPC workloads.

The ND MI300X v5 series VM starts with eight AMD Instinct MI300 GPUs and two fourth Gen Intel Xeon Scalable processors for a total 96 physical cores. Each GPU within the VM is then connected to one another via 4th-Gen AMD Infinity Fabric links with 128 GB/s bandwidth per GPU and 896 GB/s aggregate bandwidth.

ND MI300X v5-based deployments can scale up to thousands of GPUs with 3.2 Tb/s of interconnect bandwidth per VM. Each GPU within the VM is provided with its own dedicated, topology-agnostic 400 Gb/s NVIDIA Quantum-2 CX7 InfiniBand connection. These connections are automatically configured between VMs occupying the same virtual machine scale set, and support GPUDirect RDMA.

These instances provide excellent performance for many AI, ML, and analytics tools that support GPU acceleration "out-of-the-box," such as TensorFlow, Pytorch, and other frameworks. Additionally, the scale-out InfiniBand interconnect supports a large set of existing AI and HPC tools that are built on AMD's ROCm Communication Collectives Library (RCCL) for seamless clustering of GPUs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	96 vCPUs	Intel Xeon (Sapphire Rapids) [x86-64]
Memory	1850 GiB	
Local Storage	1 Temp Disk 8 NVMe Disks	1000 GiB Temp Disk 28000 GiB NVMe Disks
Remote Storage	16 Disks	80000 IOPS 1200 MBps
Network	8 NICs	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	8 GPUs	AMD Instinct MI300X GPU (192GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_ND96isr_MI300X_v5	96	1850

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported

Feature name	Support status
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

'NG' sub-family GPU accelerated VM size series

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'NG' family of VM size series are one of Azure's GPU-optimized VM instances, specifically designed for cloud gaming and remote desktop applications. They harness powerful AMD Radeon™ PRO GPUs to deliver high-quality, interactive gaming experiences in the cloud, optimized for rendering complex graphics and streaming high-definition video. This ensures gamers enjoy a seamless, responsive gaming environment accessible from any device. Additionally, NG-series VMs provide a high-quality, responsive remote desktop experience, making them ideal for users needing reliable, high-performance access to desktop applications from anywhere in the world.

Workloads and use cases

Cloud Gaming: NG-family VMs harness powerful AMD Radeon™ PRO GPUs to deliver high-quality, interactive gaming experiences in the cloud.

Remote Destkop: NG-family VMs can be used for remote desktop applications, providing users with a high-quality, responsive user experience.

Series in family

NGads V620-series

The NGads V620 series are GPU-enabled virtual machines with CPU, memory resources and storage resources balanced to generate and stream high quality graphics for a high performance, interactive gaming experience hosted in Azure. They're powered by AMD Radeon(tm) PRO V620 GPU and AMD EPYC 7763 (Milan) CPUs. The AMD Radeon PRO V620 GPUs have a maximum frame buffer of 32 GB, which can be divided up to four ways through hardware partitioning. The AMD EPYC CPUs have a base clock speed of 2.45 GHz and a boost1 speed of 3.5Ghz. VMs are assigned full cores instead of threads, enabling full access to AMD’s powerful “Zen 3” cores.

NGads instances come in four sizes, allowing customers to right-size their gaming environments for the performance and cost that best fits their business needs. The NG-

series virtual machines feature partial GPUs to enable you to pick the right-sized virtual machine for GPU accelerated graphics applications and virtual desktops. The vm sizes start with 1/4 of a GPU with 8-GiB frame buffer up to a full GPU with 32-GiB frame buffer. The NGads VMs also feature Direct Disk NVMe ranging from 1 to 4x 960 GB disks per VM.

> **Note**
>
> Max boost for AMD EPYC processors is the maximum frequency achievable by any single core on the processor under normal operating conditions for server systems.)

View the full NGads v620-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 32 vCPUs	AMD EPYC 7763 (Milan) [x86-64]
Memory	16 - 176 GiB	
Local Storage	1 Disk	256 - 1024 GiB
Remote Storage	8 - 32 Disks	12800 - 51200 IOPS 200 - 768 MBps
Network	2 - 8 NICs	10000 - 40000 Mbps
Accelerators	1/4 - 1 GPUs	AMD Radeon PRO V620 GPU (32GB)

Previous-generation NG family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Check out Azure Dedicated Hosts for physical servers able to host one or more virtual machines assigned to one Azure subscription.

Learn how to Monitor Azure virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

NGads_V620 sizes series

Summarize this article for me

The NGads V620 series are GPU-enabled virtual machines with CPU, memory resources and storage resources balanced to generate and stream high quality graphics for a high performance, interactive gaming experience hosted in Azure. They're powered by AMD Radeon(tm) PRO V620 GPU and AMD EPYC 7763 (Milan) CPUs. The AMD Radeon PRO V620 GPUs have a maximum frame buffer of 32 GB, which can be divided up to four ways through hardware partitioning. The AMD EPYC CPUs have a base clock speed of 2.45 GHz and a boost1 speed of 3.5Ghz. VMs are assigned full cores instead of threads, enabling full access to AMD's powerful "Zen 3" cores.

NGads instances come in four sizes, allowing customers to right-size their gaming environments for the performance and cost that best fits their business needs. The NG-series virtual machines feature partial GPUs to enable you to pick the right-sized virtual machine for GPU accelerated graphics applications and virtual desktops. The vm sizes start with 1/4 of a GPU with 8-GiB frame buffer up to a full GPU with 32-GiB frame buffer. The NGads VMs also feature Direct Disk NVMe ranging from 1 to 4x 960 GB disks per VM.

> **Note**
>
> Max boost for AMD EPYC processors is the maximum frequency achievable by any single core on the processor under normal operating conditions for server systems.)

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 32 vCPUs	AMD EPYC 7763 (Milan) [x86-64]
Memory	16 - 176 GiB	
Local Storage	1 Disk	256 - 1024 GiB
Remote Storage	8 - 32 Disks	12800 - 51200 IOPS 200 - 768 MBps
Network	2 - 8 NICs	10000 - 40000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	1/4 - 1 GPUs	AMD Radeon PRO V620 GPU (32GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NG8ads_V620_v1	8	16
Standard_NG16ads_V620_v1	16	32
Standard_NG32ads_V620_v1	32	64
Standard_NG32adms_V620_v1	32	176

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported

Feature name	Support status
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

'NV' sub-family GPU accelerated VM size series

Article • 11/07/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'NV' family of VM size series are one of Azure's GPU-accelerated VM instances, specifically designed for graphics-intensive applications such as graphics rendering, simulation, and virtual desktops. Equipped with NVIDIA GPUs, NV-series VMs provide a robust platform for rendering and processing graphics-heavy tasks, making them ideal for organizations that require virtual workstations with powerful graphical capabilities. These VMs support scenarios where remote visualization, real-time collaboration, and 3D visualization are necessary, allowing users to efficiently run graphic-intensive applications directly from Azure's cloud environment.

Workloads and use cases

Virtual Desktop Infrastructure (VDI): NV-family VMs are well-suited for virtual desktops that require GPU capabilities for tasks such as graphic design, video editing, and CAD applications. They provide the graphical performance necessary for smooth operation in remote desktop scenarios.

3D Visualization: NV-family VMs are ideal for running 3D applications that demand high-performance rendering, such as architectural visualizations, medical imaging, and other professional-grade graphics tasks.

Remote Graphics Work: NV-series VMs are beneficial for industries that rely on graphics-intensive software, allowing professionals to access and use applications like Adobe Photoshop, Autodesk AutoCAD, or Dassault SOLIDWORKS remotely with near-native performance.

High-Resolution Image Processing: NV-series VMs are ideal for handling extremely large vRAM applications such as high-resolution image processing and analysis. This includes tasks in fields like geospatial analysis, satellite image processing, and professional photo editing, where handling massive image files and performing complex manipulations in real-time are crucial for productivity and performance.

Video Streaming: NV-family VMs are suitable for streaming high-resolution video content, including training videos and virtual events, ensuring high-quality delivery

without local hardware constraints.

Series in family

NV-series V1

> **Important**
>
> NV and NV_Promo series Azure virtual machines (VMs) will be retired on September 6, 2023. For more information, see the **NV and NV_Promo retirement information**. For how to migrate your workloads to other VM sizes, see the **NV and NV_Promo series migration guide**.
>
> This retirement announcement doesn't apply to NVv3 and NVv4 series VMs.

The NV-series virtual machines are powered by NVIDIA Tesla M60 GPUs and NVIDIA GRID technology for desktop accelerated applications and virtual desktops where customers are able to visualize their data or simulations. Users are able to visualize their graphics intensive workflows on the NV instances to get superior graphics capability and additionally run single precision workloads such as encoding and rendering. NV-series VMs are also powered by Intel Xeon E5-2690 v3 (Haswell) CPUs.

Each GPU in NV instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

View the full NV-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 24 vCPUs	Intel Xeon E5-2690 v3 (Haswell) [x86-64]
Memory	56 - 224 GiB	
Local Storage	1 Disk	340 - 1440 GiB
Remote Storage	24 - 64	
Network	1 - 4 NICs	
Accelerators	1/2 - 2 GPUs	Nvidia Tesla M60 GPU (16GB)

NVv3-series

The NVv3-series virtual machines are powered by NVIDIA Tesla M60 GPUs and NVIDIA GRID technology with Intel E5-2690 v4 (Broadwell) CPUs and Intel Hyper-Threading Technology. These virtual machines are targeted for GPU accelerated graphics applications and virtual desktops where customers want to visualize their data, simulate results to view, work on CAD, or render and stream content. Additionally, these virtual machines can run single precision workloads such as encoding and rendering. NVv3 virtual machines support Premium Storage and come with twice the system memory (RAM) when compared with its predecessor NV-series.

Each GPU in NVv3 instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

View the full NVv3-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	12 - 48 vCPUs	Intel Xeon E5-2690 v4 (Broadwell) [x86-64]
Memory	112 - 448 GiB	
Local Storage	1 Disk	320 - 1280 GiB
Remote Storage	12 - 32 Disks	
Network	4 - 8 NICs	6000 - 24000 Mbps
Accelerators	1 - 4 GPUs	Nvidia Tesla M60 GPU (16GB)

NVv4-series

The NVv4-series virtual machines are powered by AMD Radeon Instinct MI25 GPUs and AMD EPYC 7V12(Rome) CPUs with a base frequency of 2.45GHz, all-cores peak frequency of 3.1GHz and single-core peak frequency of 3.3GHz. With NVv4-series Azure is introducing virtual machines with partial GPUs. Pick the right sized virtual machine for GPU accelerated graphics applications and virtual desktops starting at 1/8th of a GPU with 2 GiB frame buffer to a full GPU with 16 GiB frame buffer. NVv4 virtual machines currently support only Windows guest operating system.

View the full NVv4-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 32 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	14 - 112 GiB	
Local Storage	1 Disk	88 - 704 GiB
Remote Storage	4 - 32 Disks	6400 - 51200 IOPS 96 - 768 MBps
Network	2 - 8 NICs	1000 - 8000 Mbps
Accelerators	1/8 - 1 GPUs	AMD Instinct MI25 GPU (16GB)

NVads-A10 v5-series

The NVadsA10v5-series virtual machines are powered by NVIDIA A10 GPUs and AMD EPYC 74F3V(Milan) CPUs with a base frequency of 3.2 GHz, all-cores peak frequency of 4.0 GHz. With NVadsA10v5-series Azure is introducing virtual machines with partial NVIDIA GPUs. Pick the right sized virtual machine for GPU accelerated graphics applications and virtual desktops starting at 1/6th of a GPU with 4-GiB frame buffer to a full A10 GPU with 24-GiB frame buffer. Each virtual machine instance in NVadsA10v5-series comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

View the full NVads-A10 v5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 72 vCPUs	AMD EPYC 74F3v (Milan) [x86-64]
Memory	55 - 880 GiB	
Local Storage	1 Disk	180 - 2880 GiB
Remote Storage	4 - 32 Disks	6400 - 80000 IOPS 100 - 1200 MBps
Network	1 - 8 NICs	5000 - 80000 Mbps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	1/6 - 2 GPUs	Nvidia A10 GPU (24GB)

NVads-v710 v5-series

The NVads V710 v5-series virtual machines are powered by AMD Radeon™ Pro V710 GPUs and AMD EPYC™ 9V64 F (Genoa) CPUs with a base frequency of 3.95 GHz and all-cores peak frequency of 4.3 GHz. VMs take advantage of the AMD Simultaneous Multithreading technology to assign dedicated vCPU threads to each VM. Both Windows and Linux VMs are supported.

The series provides five options ranging from 1/6 of a GPU with 4-GiB frame buffer to a full V710 GPU with 24-GiB frame buffer. No other GPU licensing is required for use of AMD GPU-based VMs. The NVads V710 v5 VMs also support NVMe for ephemeral local storage capability. The NVads V710 v5-series enables right-sizing for demanding GPU-accelerated graphics applications and cloud-based virtual desktops to provide a seamless end user experience while providing a cost-effective choice for a full range of graphics-enabled virtual desktop experiences. The VMs are also sized to deliver high-quality, interactive gaming experiences in the cloud, optimized for rendering and streaming complex graphics.

The NVads V710 v5-series virtual machines also support small to medium AI/ML inference workloads such as Small Language Model (SLMs), recommendation systems and semantic indexing, by taking advantage of the computational IP blocks in the Radeon Pro V710 GPUs.

View the full NVadsv710 v5-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 28 vCPUs	AMD EPYC 9V64 F (Genoa) [x86-64]
Memory	16 - 160 GiB	
Local Storage	2 - 14 Disks	256 - 1536 GiB IOPS (RR) MBps (RR)
Remote Storage	2 - 14 Disks	7600 - 5320 IOPS 180 - 1260 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - 8 NICs	3300 - 20000 Mbps
Accelerators	None	

Previous-generation NV family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Check out Azure Dedicated Hosts for physical servers able to host one or more virtual machines assigned to one Azure subscription.

Learn how to Monitor Azure virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

NV sizes series

> **Important**
>
> The NV-series is retired as of September 6th, 2023, and is no longer available. Please refer to the NV-series retirement page for more information.

> **Important**
>
> NV and NV_Promo series Azure virtual machines (VMs) will be retired on September 6, 2023. For more information, see the NV and NV_Promo retirement information. For how to migrate your workloads to other VM sizes, see the NV and NV_Promo series migration guide.
>
> This retirement announcement doesn't apply to NVv3 and NVv4 series VMs.

The NV-series virtual machines are powered by NVIDIA Tesla M60 GPUs and NVIDIA GRID technology for desktop accelerated applications and virtual desktops where customers are able to visualize their data or simulations. Users are able to visualize their graphics intensive workflows on the NV instances to get superior graphics capability and additionally run single precision workloads such as encoding and rendering. NV-series VMs are also powered by Intel Xeon E5-2690 v3 (Haswell) CPUs.

Each GPU in NV instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 24 vCPUs	Intel Xeon E5-2690 v3 (Haswell) [x86-64]
Memory	56 - 224 GiB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	1 Disk	340 - 1440 GiB
Remote Storage	24 - 64	
Network	1 - 4 NICs	
Accelerators	1/2 - 2 GPUs	Nvidia Tesla M60 GPU (16GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NV6	6	56
Standard_NV12	12	112
Standard_NV24	24	224

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Not Supported

Feature name	Support status
Memory Preserving Updates	Not Supported
Generation 2 VMs	Not Supported
Generation 1 VMs	Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Migrate your NV and NV_Promo series virtual machines by September 6, 2023

03/17/2025

We're happy to announce that we're extending the retirement date by one year to September 6, 2023, for the Azure NV-Series and NV_Promo Series virtual machine to give you more time to plan your migration.

We continue to bring modern and optimized virtual machine (VM) instances to Azure by using the latest innovations in datacenter technologies. As we innovate, we also thoughtfully plan how we retire aging hardware. With this context in mind, we're retiring our NV-series Azure VM sizes on September 6, 2023.

How does the NV series migration affect me?

After September 6, 2023, any remaining NV and NV_Promo-size VMs remaining in your subscription will be set to a deallocated state. These VMs will be stopped and removed from the host. These VMs will no longer be billed in the deallocated state.

The current VM size retirement only affects the VM sizes in the NV series. This retirement doesn't affect the NVv3 and NVv4 series VMs.

What actions should I take?

You'll need to resize or deallocate your NV VMs. We recommend moving your GPU visualizations or graphics workloads to another GPU accelerated VM size.

Learn more about migrating your workloads to other GPU Azure VM sizes.

If you have questions, contact us through customer support.

NV series migration guide

> **Important**
>
> The NV-series is retired as of September 6th, 2023, and is no longer available. Please refer to the NV-series retirement page for more information.

As more powerful GPU VM sizes become available in Azure datacenters, assess your workloads and migrate virtual machines (VMs) in the NV and NV_Promo series. These legacy VMs can be migrated into new VM series, such as NVsv3 and NVasv4, for better performance with reduced cost. The NVsv3 VM series is powered by Nvidia M60 GPUs. The NVasv4 series is powered by AMD Radeon Instinct MI25 GPUs.

The main differences between the NV and NV_Promo series and the newer NVsv3 and NVasv4 series are:

- Improved performance.
- Support for Premium storage.
- The option to choose from a fractional GPU size to multi-GPU configurations.

Both the NVsv3 and NVasv4 series have more modern cores and greater capacity.

The following section summarizes the differences between the legacy NV series and the NVsv3 and NVv4 series.

NVsv3 series

The NVv3-series VMs are powered by NVIDIA Tesla M60 GPUs and NVIDIA GRID technology with Intel E5-2690 v4 (Broadwell) CPUs and Intel Hyper-Threading Technology. These VMs are targeted for GPU accelerated graphics applications and virtual desktops where customers want to:

- Visualize their data.
- Simulate results to view.
- Work on CAD.
- Render and stream content.

These VMs can also run single precision workloads, such as encoding and rendering.

NVv3 VMs support Premium storage and come with twice the system memory (RAM) when compared with the NV series. For the most up-to-date specifications, see GPU accelerated compute VM sizes: NVsv3 series.

Expand table

Current VM size	Target VM size	Difference in specification
Standard_NV6 Standard_NV6_Promo	Standard_NV12s_v3	vCPU: 12 (+6) Memory: GiB 112 (+56) Temp storage (SSD) GiB: 320 (-20) Max data disks: 12 (-12) Accelerated networking: Yes Premium storage: Yes
Standard_NV12 Standard_NV12_Promo	Standard_NV24s_v3	vCPU: 24 (+12) Memory: GiB 224 (+112) Temp storage (SSD) GiB: 640 (-40) Max data disks: 24 (-24) Accelerated networking: Yes Premium storage: Yes
Standard_NV24 Standard_NV24_Promo	Standard_NV48s_v3	vCPU: 48 (+24) Memory: GiB 448 (+224) Temp storage (SSD) GiB: 1280 (-160) Max data disks: 32 (-32) Accelerated networking: Yes Premium storage: Yes

NVv4 series

NVv4-series VMs are powered by AMD Radeon Instinct MI25 GPUs and AMD EPYC 7V12 (Rome) CPUs. With the NVv4 series, Azure introduces VMs with partial GPUs. Choose the right size VM for GPU accelerated graphics applications and virtual desktops that start at one-eighth of a GPU with a 2-GiB frame buffer to a full GPU with a 16-GiB frame buffer.

NVv4 VMs currently support only the Windows guest operating system. For the most up-to-date specifications, see GPU accelerated compute VM sizes: NVv4 series.

Expand table

Current VM size	Target VM size	Difference in specification
Standard_NV6 Standard_NV6_Promo	Standard_NV16as_v4	vCPU: 16 (+10) Memory: GiB 56

Current VM size	Target VM size	Difference in specification
		Temp storage (SSD) GiB: 352 (+12) Max data disks: 16 (-8) Accelerated networking: Yes Premium storage: Yes
Standard_NV12 Standard_NV12_Promo	Standard_NV32as_v4	vCPU: 32 (+20) Memory: GiB 112 Temp storage (SSD) GiB: 704 (+24) Max data disks: 32 (+16) Accelerated networking: Yes Premium storage: Yes
Standard_NV24 Standard_NV24_Promo	N/A	N/A

Migration steps for general changes

To deal with general changes:

1. Choose a series and size for migration.
2. Get a quota for the target VM series.
3. Resize the current NV-series VM size to the target size.

If the target size is NVv4, make sure to remove the Nvidia GPU driver and install the AMD GPU driver.

Migration steps for breaking changes

To deal with breaking changes, follow the steps in the next sections.

Select a target size for migration

After you assess your current usage, decide what type of GPU VM you need. Depending on the workload requirements, you have a few different choices. Here's how to choose:

- If the workload is graphics or visualizations and has a hard dependency on using the Nvidia GPU, migrate to the NVsv3 series.
- If the workload is graphics or visualizations and has no hard dependency on a specific type of GPU, migrate to the NVsv3 or NVVasv4 series.

> **Note**
>
> A best practice is to select a VM size based on both cost and performance. The recommendations in this article are based on a one-to-one comparison of performance metrics for the NV and NV_Promo sizes and the nearest match in another VM series. Before you decide on the right size, get a cost comparison by using the Azure Pricing Calculator.

Get a quota for the target VM family

Follow the guide to request an increase in vCPU quota by VM family. Select the NVSv3 series or NVv4 series as the VM family name depending on the target VM size you selected for migration.

Resize the current VM

You can resize the VM.

FAQ

Q: Which GPU driver should I use for the target VM size?

A: For the NVsv3 series, use the Nvidia GRID driver. For NVv4, use the AMD GPU drivers.

Q: I use the Nvidia GPU driver extension today. Will it work for the target VM size?

A: The current Nvidia driver extension will work for NVsv3. Use the AMD GPU driver extensions if the target VM size is NVv4.

Q: Which target VM series should I use if I have dependency on CUDA?

A: NVv3 supports CUDA. The NVv4 VM series with the AMD GPUs doesn't support CUDA.

Last updated on 07/11/2025

NVv3 sizes series

> **Note**
>
> The NVv3-series will be retired on September 30, 2026. Refer to the NVv3-series retirement page for more information and migration recommendations.

The NVv3-series virtual machines are powered by NVIDIA Tesla M60 GPUs and NVIDIA GRID technology with Intel E5-2690 v4 (Broadwell) CPUs and Intel Hyper-Threading Technology. These virtual machines are targeted for GPU accelerated graphics applications and virtual desktops where customers want to visualize their data, simulate results to view, work on CAD, or render and stream content. Additionally, these virtual machines can run single precision workloads such as encoding and rendering. NVv3 virtual machines support Premium Storage and come with twice the system memory (RAM) when compared with its predecessor NV-series.

Each GPU in NVv3 instances comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	12 - 48 vCPUs	Intel Xeon E5-2690 v4 (Broadwell) [x86-64]
Memory	112 - 448 GiB	
Local Storage	1 Disk	320 - 1280 GiB
Remote Storage	12 - 32 Disks	
Network	4 - 8 NICs	6000 - 24000 Mbps
Accelerators	1 - 4 GPUs	Nvidia Tesla M60 GPU (16GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NV12s_v3	12	112
Standard_NV24s_v3	24	224
Standard_NV48s_v3	48	448

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Migrate your NVv3-series virtual machines by September 30, 2026

09/20/2025

> **Note**
>
> 1-year and 3-year RI purchases for the NVv3-series will end September 30, 2025.

On September 30, 2026, Microsoft Azure will retire the Standard_NV12s_v3, Standard_NV24s_v3, Standard_NV24ms_v3, Standard_NV32ms_v3, and Standard_NV48s_v3 virtual machines (VMs) in the NVv3-series virtual machines (VMs). To avoid any disruptions to your service, we recommend that you change the VM sizing for your workloads from the current NVv3-series VMs to the newer VM series in the same NV product line.

Microsoft is recommending the Azure NVadsA10_v5-series VMs, which offer greater GPU memory bandwidth per GPU. With the NVadsA10_v5-series VMs, Azure introduces VMs with partial NVIDIA GPUs and each VM instance comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user or 25 concurrent users can connect to the VM for a virtual application scenario. These VMs are targeted for GPU accelerated graphics applications, virtual desktops, visualizations, or small AI workloads.

Depending on the workload being run, regional affinity, and cost preferences, other VMs that may be migrated to from the NVv3-series VMs include NCasT4_v3 and NVadsV710_v5:

Expand table

Workload	Recommended SKU to Migrate to
GPU accelerated graphics applications, virtual desktops, visualizations, or small AI workloads.	NVadsA10_v5
Offline inferencing workloads where latency is not a primary concern, and there is an interest in purchasing smaller VM SKUs or reducing costs.	NCasT4_v3
Graphics, visualization, or small AI workloads such as SLM inferencing and semantic search where optimal performance is not a priority or there is an interest in reducing costs.	NVadsV710_v5

How does the retirement of the NVv3-series virtual machines affect me?

After September 30th, 2026, any remaining NVv3-series virtual machines (VMs) subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. NVv3 will no longer be under SLA or have support included.

> **Note**
>
> This retirement only impacts the virtual machine sizes in the NVv3-series powered by NVIDIA Tesla M60 GPUs. For NVv4-series virtual machines, please refer to the NVv4-series virtual machines retirement guide. This retirement announcement does not apply to NVadsA10_v5 or NVadsV710_v5 series virtual machines.

What action do I need to take before the retirement date?

You need to resize or deallocate your NVv3-series VMs. We recommend that you change VM sizes for these workloads from the original NVv3-series VMs to the NVadsA10_v5-series VMs (or an alternative).

The NVadsA10_v5-series is powered by NVIDIA A10 GPUs and AMD EPYC™ 74F3V(Milan) processors. The VMs feature up to 2 NVIDIA A10 GPUs with 24GB memory each, up to 72 non-multithreaded AMD EPYC 74F3V processor cores and 880 GiB of system memory. Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Refer to the above tables for Microsoft's recommendation. You can also file a support request if more assistance is needed.
2. Request quota for the new target VM.
3. You can resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.
2. Under *Subscription*, select your subscription.

3. Under *Service*, click **My services**.

4. Under *Service type*, select **Virtual Machine running Windows/Linux**.

5. Under *Summary*, enter the summary of your request.

6. Under *Problem type*, select **Assistance with resizing my VM.**

7. Under *Problem subtype*, select the option that applies to you.

NVv4 sizes series

Summarize this article for me

> **Note**
>
> The NVv4-series will be retired on September 30, 2026. Refer to the NVv4-series retirement page for more information and migration recommendations.

The NVv4-series virtual machines are powered by AMD Radeon Instinct MI25 GPUs and AMD EPYC 7V12(Rome) CPUs with a base frequency of 2.45GHz, all-cores peak frequency of 3.1GHz and single-core peak frequency of 3.3GHz. With NVv4-series Azure is introducing virtual machines with partial GPUs. Pick the right sized virtual machine for GPU accelerated graphics applications and virtual desktops starting at 1/8th of a GPU with 2 GiB frame buffer to a full GPU with 16 GiB frame buffer. NVv4 virtual machines currently support only Windows guest operating system.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	4 - 32 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
Memory	14 - 112 GiB	
Local Storage	1 Disk	88 - 704 GiB
Remote Storage	4 - 32 Disks	6400 - 51200 IOPS 96 - 768 MBps
Network	2 - 8 NICs	1000 - 8000 Mbps
Accelerators	1/8 - 1 GPUs	AMD Instinct MI25 GPU (16GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NV4as_v4	4	14
Standard_NV8as_v4	8	28
Standard_NV16as_v4	16	56
Standard_NV32as_v4	32	112

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Migrate your NVv4-series virtual machines by September 30, 2026

09/20/2025

> **Note**
>
> There is a known resize operation error that occurs when migrating from the NVv4-series to the NVads_V710_v5-series. Microsoft is working on a fix that will be implemented in April 2026. In the meantime, we suggest that you follow this workaround.

> **Note**
>
> 1-year and 3-year RI purchases for the NVv4-series will end September 30, 2025.

On September 30, 2026, Microsoft Azure will retire the Standard_NV4as_v4, Standard_NV8as_v4, Standard_NV16as_v4, and Standard_NV32as_v4 virtual machines (VMs) in NVv4-series virtual machines (VMs). To avoid any disruption to your service, we recommend that you change the VM sizing for your workloads from the current NVv4-series VMs to the newer VM series in the same NV product line.

Microsoft is recommending the Azure NVads_V710_v5-series VMs, which offer greater GPU memory bandwidth per GPU. The NVads_V710_v5-series VMs take advantage of the AMD Simultaneous Multithreading technology to assign dedicated vCPU threads to each VM and support NVMe for ephemeral local storage capability.

The NVads_V710_v5-series enables right-sizing for demanding GPU-accelerated graphics applications and cloud-based virtual desktops to provide a seamless end user experience while providing a cost-effective choice for a full range of graphics-enabled virtual desktop experiences. The VMs are also sized to deliver high-quality, interactive gaming experiences in the cloud, optimized for rendering and streaming complex graphics. In addition, the NVads_V710_v5-series supports small AI inference workloads such as Small Language Model (SLMs), recommendation systems, and semantic indexing, by taking advantage of the computational IP blocks in the Radeon Pro V710 GPUs.

Expand table

Workload	Recommended SKU to Migrate to
Small AI workloads such as SLM inferencing and semantic search where optimal performance is not a priority or there is an interest in reducing costs.	NVads_V710_v5
GPU accelerated graphics applications, virtual desktops, and visualizations.	NVads_V710_v5

How does the retirement of the NVv4-series virtual machines affect me?

After September 30th, 2026, any remaining NVv4-series virtual machines (VMs) subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. NVv4 will no longer be under SLA or have support included.

> **Note**
>
> This retirement only impacts the virtual machine sizes in the NVv4-series powered by AMD Radeon Instinct MI25 GPUs. For NVv3-series virtual machines, please refer to the NVv3-series virtual machines retirement guide. This retirement announcement does not apply to NVadsA10_v5 or NVads_V710_v5 series virtual machines.

What action do I need to take before the retirement date?

You need to resize or deallocate your NVv4-series VMs. We recommend that you change VM sizes for these workloads from the original NVv4-series VMs to the NVads_V710_v5-series VMs (or an alternative).

The NVads_V710_v5-series is powered by AMD Radeon™ Pro V710 GPUs and AMD EPYC™ 9V64 F (Genoa) processors. The VMs feature up to 1 AMD Radeon™ Pro V710 GPU with 24GB memory each, up to 28 multithreaded AMD EPYC 9V64 F processor cores and 160 GiB of system memory. Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Refer to the above tables for Microsoft's recommendation. You can also file a support request if more assistance is needed.

2. Request quota for the new target VM.

3. You can resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.

2. Under *Subscription*, select your subscription.

3. Under *Service*, click **My services**.

4. Under *Service type*, select **Virtual Machine running Windows/Linux**.

5. Under *Summary*, enter the summary of your request.

6. Under *Problem type*, select **Assistance with resizing my VM.**

7. Under *Problem subtype*, select the option that applies to you.

NVadsA10_v5 sizes series

Summarize this article for me

The NVadsA10v5-series virtual machines are powered by NVIDIA A10 GPUs and AMD EPYC 74F3V(Milan) CPUs with a base frequency of 3.2 GHz, all-cores peak frequency of 4.0 GHz. With NVadsA10v5-series Azure is introducing virtual machines with partial NVIDIA GPUs. Pick the right sized virtual machine for GPU accelerated graphics applications and virtual desktops starting at 1/6th of a GPU with 4-GiB frame buffer to a full A10 GPU with 24-GiB frame buffer. Each virtual machine instance in NVadsA10v5-series comes with a GRID license. This license gives you the flexibility to use an NV instance as a virtual workstation for a single user, or 25 concurrent users can connect to the VM for a virtual application scenario.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	6 - 72 vCPUs	AMD EPYC 74F3v (Milan) [x86-64]
Memory	55 - 880 GiB	
Local Storage	1 Disk	180 - 2880 GiB
Remote Storage	4 - 32 Disks	6400 - 80000 IOPS 100 - 1200 MBps
Network	1 - 8 NICs	5000 - 80000 Mbps
Accelerators	1/6 - 2 GPUs	Nvidia A10 GPU (24GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NV6ads_A10_v5	6	55
Standard_NV12ads_A10_v5	12	110
Standard_NV18ads_A10_v5	18	220
Standard_NV36ads_A10_v5	36	440
Standard_NV36adms_A10_v5	36	880
Standard_NV72ads_A10_v5	72	880

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Local temporary storage	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

NVads V710 v5-series

The NVads V710 v5-series virtual machines are powered by AMD Radeon™ Pro V710 GPUs and AMD EPYC™ 9V64 F (Genoa) CPUs with a base frequency of 3.95 GHz and all-cores peak frequency of 4.3 GHz. VMs take advantage of the AMD Simultaneous Multithreading technology to assign dedicated vCPU threads to each VM. Both Windows and Linux VMs are supported.

The series provides five options ranging from 1/6 of a GPU with 4-GiB frame buffer to a full V710 GPU with 24-GiB frame buffer. No other GPU licensing is required for use of AMD GPU-based VMs. The NVads V710 v5 VMs also support NVMe for ephemeral local storage capability. The NVads V710 v5-series enables right-sizing for demanding GPU-accelerated graphics applications and cloud-based virtual desktops to provide a seamless end user experience while providing a cost-effective choice for a full range of graphics-enabled virtual desktop experiences. The VMs are also sized to deliver high-quality, interactive gaming experiences in the cloud, optimized for rendering and streaming complex graphics.

The NVads V710 v5-series virtual machines also support small to medium AI/ML inference workloads such as Small Language Model (SLMs), recommendation systems and semantic indexing, by taking advantage of the computational IP blocks in the Radeon Pro V710 GPUs.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	4 - 28 vCPUs	AMD EPYC 9V64 F (Genoa) [x86-64]
Memory	16 - 160 GiB	
Local Storage	2 - 14 Disks	256 - 1536 GiB IOPS (RR) MBps (RR)
Remote Storage	2 - 14 Disks	7600 - 5320 IOPS 180 - 1260 MBps
Network	2 - 8 NICs	3300 - 20000 Mbps
Accelerators	1/6 - 1	AMD Radeon™ Pro V710

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NV4ads_V710_v5	4	16
Standard_NV8ads_V710_v5	8	32
Standard_NV12ads_V710_v5	12	64
Standard_NV24ads_V710_v5	24	128
Standard_NV28adms_V710_v5	28	160

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported

Feature name	Support status
Local temporary storage	Supported
Nested Virtualization	Supported

Certified ISV Applications

Expand table

Application	Video Card	OS	Driver
3ds Max 2025	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
After Effects 25.1.0 (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Ansys Fluent 2025 R1 (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Ansys HFSS 2025 R1 (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Ansys Mechanical APDL 2025 R1 (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
AutoCAD 2025	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Autodesk Fusion 360	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Dassault 3DEXPERIENCE Part Design (CATIA) (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	25.Q2
Inventor 2025	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Maya 2025	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Photoshop 26.3.0 (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Premiere Pro 25.1.0 (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2
Revit 2024	AMD Radeon™ PRO	Windows 11 - 64-Bit	24.Q2

Application	Video Card	OS	Driver
	V710	Edition	
Siemens NX (Verified / Tested)	AMD Radeon™ PRO V710	Windows 11 - 64-Bit Edition	24.Q2

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/19/2026

Install NVIDIA GPU drivers on N-series VMs running Linux

> **⊗ Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs

> **ⓘ Important**
>
> To align with inclusive language practices, we've replaced the term "blacklist" with "blocklist" throughout this documentation. This change reflects our commitment to avoiding terminology that might carry unintended negative connotations or perceived racial bias. However, in code snippets and technical references where "blacklist" is part of established syntax or tooling (for example, configuration files, command-line parameters), the original term is retained to preserve functional accuracy. This usage is strictly technical and doesn't imply any discriminatory intent.

To take advantage of the GPU capabilities of Azure N-series VMs backed by NVIDIA GPUs, you must install NVIDIA GPU drivers. The NVIDIA GPU Driver Extension installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as the Azure CLI or Azure Resource Manager templates. See the NVIDIA GPU Driver Extension documentation for supported distributions and deployment steps.

If you choose to install NVIDIA GPU drivers manually, this article provides supported distributions, drivers, and installation and verification steps. Manual driver setup information is also available for Windows VMs.

For N-series VM specs, storage capacities, and disk details, see GPU Linux VM sizes.

> **⚠ Warning**
>
> Installing NVIDIA drivers using methods other than those outlined in this guide may result in failure of the intended driver installation. To ensure proper functionality and support, follow only the installation steps and use the driver versions specified in this documentation.

Supported distributions and drivers

NVIDIA CUDA drivers

For the latest CUDA drivers and supported operating systems, visit the NVIDIA website. Ensure that you install or upgrade to the latest supported CUDA drivers for your distribution.

> **Note**
>
> The latest supported CUDA driver for original NC-series SKU VMs is currently 470.82.01. Later driver versions aren't supported on the K80 cards in NC.

> **Note**
>
> The Azure NVads A10 v5 VMs only support GRID 17.x or higher driver versions. The vGPU driver for the A10 SKU is a unified driver that supports both graphics and compute workloads.

> **Caution**
>
> Secure Boot and vTPM should be disabled because the process hangs when they're enabled.

> **Tip**
>
> As an alternative to manual CUDA driver installation on a Linux VM, you can deploy an Azure Data Science Virtual Machine image. The DSVM edition for Ubuntu 16.04 LTS preinstalls NVIDIA CUDA drivers, the CUDA Deep Neural Network Library, and other tools.

NVIDIA GRID drivers

> **Note**
>
> vGPU18 is available for the NCasT4_v3-series.
>
> vGPU18 is now available for the NVadsA10_v5-series in **public regions only**. vGPU18 for the NVadsA10_v5-series is **not** supported in the Mooncake and Fairfax regions yet. We'll

provide an update once vGPU18 becomes supported for the NVadsA10_v5-series in the Mooncake and Fairfax regions.

Microsoft redistributes NVIDIA GRID driver installers for NV and NVv3-series VMs used as virtual workstations or for virtual applications. Install only these GRID drivers on Azure NV VMs, only on the operating systems listed in the following table. These drivers include licensing for GRID Virtual GPU Software in Azure. You don't need to set up an NVIDIA vGPU software license server.

The GRID drivers redistributed by Azure don't work on most non-NV series VMs, like NC, NCv2, NCv3, ND, and NDv2-series VMs, but they work on NCasT4v3 series.

For more information on the specific vGPU and driver branch versions, visit the NVIDIA website.

Expand table

Distribution	Driver
Ubuntu 20.04 LTS, 22.04 LTS, 24.04 LTS	NVIDIA vGPU 18.5, driver branch R570
Red Hat Enterprise Linux 8.6, 8.8, 8.9, 8.10, 9.0, 9.2, 9.3, 9.4, 9.5	NVIDIA vGPU 18.5, driver branch R570
SUSE Linux Enterprise Server 15 SP2, 12 SP2,12 SP5	
Rocky Linux 8.4	

Note

For Azure NVads A10 v5 VMs, we recommend that you use the latest driver version. The latest NVIDIA major driver branch (n) is only backward compatibile with the previous major branch (n-1). For example, vGPU 17.x is backward compatible with vGPU 16.x only. Driver failures might occur on any VMs still running n-2 or lower when the latest drive branch is rolled out to Azure hosts.

NVs_v3 VMs only support **vGPU 16 or lower** driver versions.

GRID Driver 17.3 currently supports only the NCasT4_v3 series of VMs. To use this driver, download and install GRID Driver 17.3 manually.

GRID drivers are having issues with installation on Azure kernel 6.11. To unblock, downgrade the kernel to version 6.8. For more information, see Known Issues.

Visit GitHub for the complete list of all previous Nvidia GRID driver links.

> ⚠ **Warning**
>
> Installation of third-party software on Red Hat products can affect the Red Hat support terms. See the Red Hat Knowledgebase article.

Install CUDA drivers on N-series VMs

Here are steps to install CUDA drivers from the NVIDIA CUDA Toolkit on N-series VMs.

C and C++ developers can optionally install the full Toolkit to build GPU-accelerated applications. For more information, see the CUDA Installation Guide.

To install CUDA drivers, make an SSH connection to each VM. To verify that the system has a CUDA-capable GPU, run the following command:

Bash

```
lspci | grep -i NVIDIA
```

Output is similar to the following example (showing an NVIDIA Tesla K80 card):

```
af8a:00:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)
```

lspci lists the PCIe devices on the VM, including the InfiniBand NIC and GPUs, if any. If lspci doesn't return successfully, you may need to install LIS on CentOS/RHEL.

Then run installation commands specific for your distribution.

Ubuntu

Ubuntu packages NVIDIA proprietary drivers. Those drivers come directly from NVIDIA and are simply packaged by Ubuntu so that they can be automatically managed by the system. Downloading and installing drivers from another source can lead to a broken system. Moreover, installing third-party drivers requires extra steps on VMs with TrustedLaunch and Secure Boot enabled. They require the user to add a new Machine Owner Key for the system to boot. Drivers from Ubuntu are signed by Canonical and will work with Secure Boot.

1. Install `ubuntu-drivers` utility:

```bash
sudo apt update && sudo apt install -y ubuntu-drivers-common
```

2. Install the latest NVIDIA drivers:

```bash
sudo ubuntu-drivers install
```

3. Reboot the VM after the GPU driver is installed:

```bash
sudo reboot
```

4. Download and install the CUDA toolkit from NVIDIA:

> **Note**
>
> The example shows the CUDA package path for Ubuntu 24.04 LTS. Use the path that's specific to the version you plan to use.
>
> Visit the NVIDIA Download Center or the NVIDIA CUDA Resources page for the full path that's specific to each version.

```bash
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/cuda
-keyring_1.1-1_all.deb
sudo apt install -y ./cuda-keyring_1.1-1_all.deb
sudo apt update
sudo apt -y install cuda-toolkit-12-5
```

The installation can take several minutes.

5. Reboot the VM after installation completes:

```bash
sudo reboot
```

6. Verify that the GPU is correctly recognized (after reboot):

```
Bash
nvidia-smi
```

NVIDIA driver updates

We recommend that you periodically update NVIDIA drivers after deployment.

```
Bash
sudo apt update
sudo apt full-upgrade
```

CentOS or Red Hat Enterprise Linux

1. Update the kernel (recommended). If you choose not to update the kernel, ensure that the versions of `kernel-devel`, and `dkms` are appropriate for your kernel.

```
sudo yum install kernel kernel-tools kernel-headers kernel-devel
sudo reboot
```

2. Install the latest Linux Integration Services for Hyper-V and Azure. Check if LIS is required by verifying the results of lspci. If all GPU devices are listed as expected, installing LIS isn't required.

   LIS is applicable to Red Hat Enterprise Linux, CentOS, and the Oracle Linux Red Hat Compatible Kernel 5.2-5.11, 6.0-6.10, and 7.0-7.7. Refer to the Linux Integration Services documentation for more details. Skip this step if you plan to use CentOS/RHEL 7.8 (or higher versions) as LIS is no longer required for these versions.

```
Bash
wget https://aka.ms/lis
tar xvzf lis
cd LISISO

sudo ./install.sh
sudo reboot
```

3. Reconnect to the VM and continue installation with the following commands:

```
Bash
sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
sudo yum-config-manager --add-repo https://developer.download.nvidia.com/compute/cuda/repos/rhel7/x86_64/cuda-rhel7.repo
sudo yum clean all
sudo yum -y install nvidia-driver-latest-dkms cuda-drivers
```

The installation can take several minutes.

> **Note**
>
> Visit Fedora and Nvidia CUDA repo to pick the correct package for the CentOS or RHEL version you want to use.

For example, CentOS 8 and RHEL 8 need the following steps.

```
Bash
sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
sudo yum install dkms

sudo wget https://developer.download.nvidia.com/compute/cuda/repos/rhel8/x86_64/cuda-rhel8.repo -O /etc/yum.repos.d/cuda-rhel8.repo

sudo yum install cuda-drivers
```

4. To optionally install the complete CUDA toolkit, type:

```
Bash
sudo yum install cuda
```

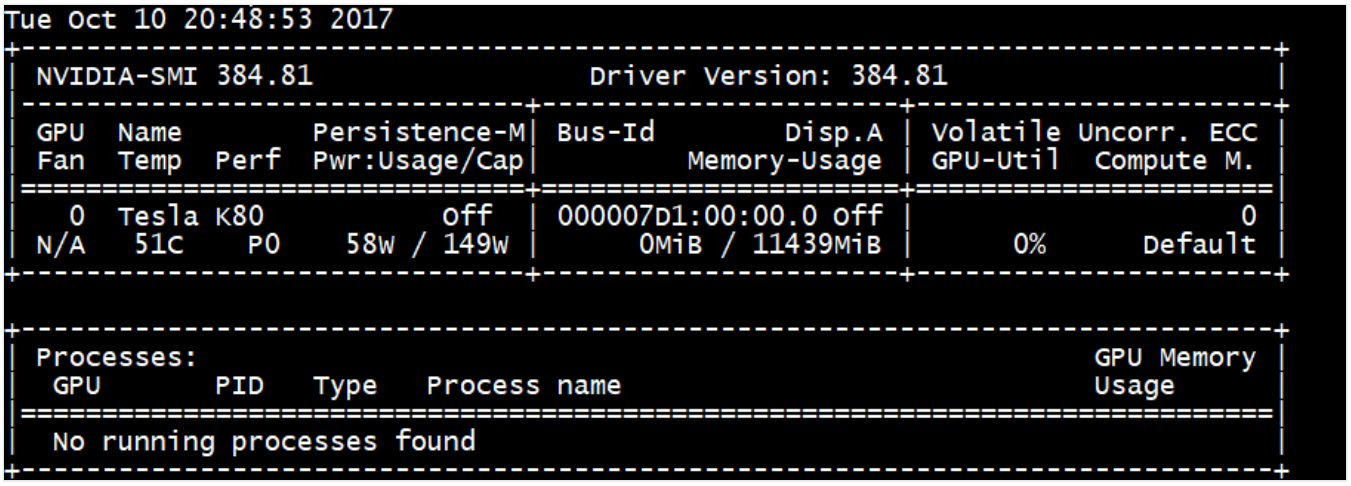
> **Note**
>
> If you see an error message related to missing packages like vulkan-filesystem, you may need to edit /etc/yum.repos.d/rh-cloud, look for optional-rpms, and set enabled to 1.

5. Reboot the VM and proceed to verify the installation.

Verify driver installation

To query the GPU device state, SSH to the VM and run the nvidia-smi command-line utility installed with the driver.

If the driver is installed, Nvidia SMI lists the **GPU-Util** as 0% until you run a GPU workload on the VM. Your driver version and GPU details may be different from the ones shown.



RDMA network connectivity

RDMA network connectivity can be enabled on RDMA-capable N-series VMs such as NC24r deployed in the same availability set or in a single placement group in a virtual machine (VM) scale set. The RDMA network supports Message Passing Interface (MPI) traffic for applications running with Intel MPI 5.x or a later version:

Distributions

Deploy RDMA-capable N-series VMs from one of the images in the Azure Marketplace that supports RDMA connectivity on N-series VMs:

- **Ubuntu 16.04 LTS** - Configure RDMA drivers on the VM and register with Intel to download Intel MPI:

    1. Install dapl, rdmacm, ibverbs, and mlx4

        Bash

        ```
        sudo apt-get update

        sudo apt-get install libdapl2 libmlx4-1
        ```

    2. In /etc/waagent.conf, enable RDMA by uncommenting the following configuration lines. You need root access to edit this file.

```
OS.EnableRDMA=y

OS.UpdateRdmaDriver=y
```

3. Add or change the following memory settings in KB in the /etc/security/limits.conf file. You need root access to edit this file. For testing purposes you can set memlock to unlimited. For example: `<User or group name> hard memlock unlimited`.

```
<User or group name> hard    memlock <memory required for your application in KB>

<User or group name> soft    memlock <memory required for your application in KB>
```

4. Install Intel MPI Library. Either purchase and download the library from Intel or download the free evaluation version.

```Bash
wget http://registrationcenter-download.intel.com/akdlm/irc_nas/tec/9278/l_mpi_p_5.1.3.223.tgz
```

   Only Intel MPI 5.x runtimes are supported.

   For installation steps, see the Intel MPI Library Installation Guide.

5. Enable ptrace for non-root non-debugger processes (needed for the most recent versions of Intel MPI).

```Bash
echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope
```

- **CentOS-based 7.4 HPC** - RDMA drivers and Intel MPI 5.1 are installed on the VM.
- **CentOS-based HPC** - CentOS-HPC 7.6 and later (for SKUs where InfiniBand is supported over SR-IOV). These images have Mellanox OFED and MPI libraries pre-installed.

> **Note**
>
> CX3-Pro cards are supported only through LTS versions of Mellanox OFED. Use LTS Mellanox OFED version (4.9-0.1.7.0) on the N-series VMs with ConnectX3-Pro cards. For

more information, see Linux Drivers.

Also, some of the latest Azure Marketplace HPC images have Mellanox OFED 5.1 and later, which don't support ConnectX3-Pro cards. Check the Mellanox OFED version in the HPC image before using it on VMs with ConnectX3-Pro cards.

The following images are the latest CentOS-HPC images that support ConnectX3-Pro cards:

- OpenLogic:CentOS-HPC:7.6:7.6.2020062900
- OpenLogic:CentOS-HPC:7_6gen2:7.6.2020062901
- OpenLogic:CentOS-HPC:7.7:7.7.2020062600
- OpenLogic:CentOS-HPC:7_7-gen2:7.7.2020062601
- OpenLogic:CentOS-HPC:8_1:8.1.2020062400
- OpenLogic:CentOS-HPC:8_1-gen2:8.1.2020062401

Install GRID drivers on NV or NVv3-series VMs

To install NVIDIA GRID drivers on NV or NVv3-series VMs, make an SSH connection to each VM and follow the steps for your Linux distribution.

Ubuntu

1. Run the `lspci` command. Verify that the NVIDIA M60 card or cards are visible as PCI devices.

2. Install updates.

   Bash

   ```
   sudo apt-get update
   sudo apt-get upgrade -y
   sudo apt-get dist-upgrade -y
   sudo apt-get install build-essential ubuntu-desktop -y
   sudo apt-get install linux-azure -y
   ```

3. Disable the Nouveau kernel driver, which is incompatible with the NVIDIA driver. (Only use the NVIDIA driver on NV or NVv2 VMs.) To disable the driver, create a file in `/etc/modprobe.d` named `nouveau.conf` with the following content:

```
blacklist nouveau
blacklist lbm-nouveau
```

4. Reboot the VM and reconnect. Exit X server:

```Bash
sudo systemctl stop lightdm.service
```

5. Download and install the GRID driver:

```Bash
wget -O NVIDIA-Linux-x86_64-grid.run https://go.microsoft.com/fwlink/?
linkid=874272
chmod +x NVIDIA-Linux-x86_64-grid.run
sudo ./NVIDIA-Linux-x86_64-grid.run
```

6. When you're asked whether you want to run the nvidia-xconfig utility to update your X configuration file, select **Yes**.

7. After installation completes, copy /etc/nvidia/gridd.conf.template to a new file gridd.conf at location /etc/nvidia/.

```Bash
sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

8. Add the following to `/etc/nvidia/gridd.conf`:

```
IgnoreSP=FALSE
EnableUI=FALSE
```

9. Remove the following from `/etc/nvidia/gridd.conf` if it's present:

```
FeatureType=0
```

10. Reboot the VM and proceed to verify the installation.

Install GRID driver on Ubuntu with Secure Boot enabled

The GRID driver installation process does not offer any options to skip kernel module build and installation and select a different source of signed kernel modules, so secure boot has to be disabled in Linux VMs in order to use them with GRID, after installing signed kernel modules.

CentOS or Red Hat Enterprise Linux

1. Update the kernel and DKMS (recommended). If you choose not to update the kernel, ensure that the versions of `kernel-devel` and `dkms` are appropriate for your kernel.

   Bash

   ```
   sudo yum update
   sudo yum install kernel-devel
   sudo rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
   sudo yum install dkms
   sudo yum install hyperv-daemons
   ```

2. Disable the Nouveau kernel driver, which is incompatible with the NVIDIA driver. (Only use the NVIDIA driver on NV or NV3 VMs.) To do this, create a file in `/etc/modprobe.d` named `nouveau.conf` with the following content:

   ```
   blacklist nouveau
   blacklist lbm-nouveau
   ```

3. Reboot the VM, reconnect, and install the latest Linux Integration Services for Hyper-V and Azure. Check if LIS is required by verifying the results of lspci. If all GPU devices are listed as expected, installing LIS isn't required.

   Skip this step if you plan to use CentOS/RHEL 7.8 (or higher versions) as LIS is no longer required for these versions.

   Bash

   ```
   wget https://aka.ms/lis
   tar xvzf lis
   cd LISISO

   sudo ./install.sh
   sudo reboot
   ```

4. Reconnect to the VM and run the `lspci` command. Verify that the NVIDIA M60 card or cards are visible as PCI devices.

5. Download and install the GRID driver:

```Bash
wget -O NVIDIA-Linux-x86_64-grid.run https://go.microsoft.com/fwlink/?linkid=874272
chmod +x NVIDIA-Linux-x86_64-grid.run

sudo ./NVIDIA-Linux-x86_64-grid.run
```

6. When you're asked whether you want to run the nvidia-xconfig utility to update your X configuration file, select **Yes**.

7. After installation completes, copy /etc/nvidia/gridd.conf.template to a new file gridd.conf at location /etc/nvidia/.

```Bash
sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

8. Add two lines to `/etc/nvidia/gridd.conf`:

```
IgnoreSP=FALSE
EnableUI=FALSE
```

9. Remove one line from `/etc/nvidia/gridd.conf` if it's present:

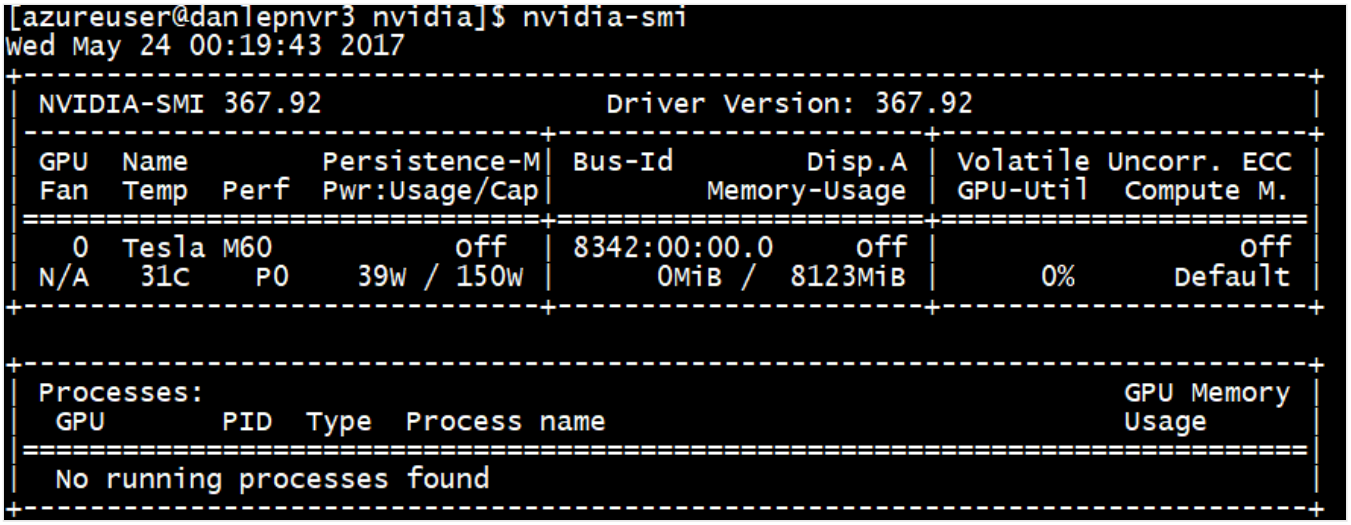
```
FeatureType=0
```

10. Reboot the VM and proceed to verify the installation.

Verify driver installation

To query the GPU device state, SSH to the VM and run the nvidia-smi command-line utility installed with the driver.

If the driver is installed, Nvidia SMI will list the **GPU-Util** as 0% until you run a GPU workload on the VM. Your driver version and GPU details may be different from the ones shown.



X11 server

If you need an X11 server for remote connections to an NV or NVv2 VM, x11vnc is recommended because it allows hardware acceleration of graphics. The BusID of the M60 device must be manually added to the X11 configuration file (usually `etc/X11/xorg.conf`). Add a `"Device"` section similar to the following:

```
Section "Device"
    Identifier     "Device0"
    Driver         "nvidia"
    VendorName     "NVIDIA Corporation"
    BoardName      "Tesla M60"
    BusID          "PCI:0@your-BusID:0:0"
EndSection
```

Additionally, update your `"Screen"` section to use this device.

You can find the decimal BusID by running

Bash

```
nvidia-xconfig --query-gpu-info | awk '/PCI BusID/{print $4}'
```

The BusID can change when a VM gets reallocated or rebooted. Therefore, you may want to create a script to update the BusID in the X11 configuration when a VM is rebooted. For example, create a script named `busidupdate.sh` (or another name you choose) with contents similar to the following:

Bash

```
#!/bin/bash
XCONFIG="/etc/X11/xorg.conf"
OLDBUSID=`awk '/BusID/{gsub(/"/, "", $2); print $2}' ${XCONFIG}`
NEWBUSID=`nvidia-xconfig --query-gpu-info | awk '/PCI BusID/{print $4}'`

if [[ "${OLDBUSID}" == "${NEWBUSID}" ]] ; then
        echo "NVIDIA BUSID not changed - nothing to do"
else
        echo "NVIDIA BUSID changed from \"${OLDBUSID}\" to \"${NEWBUSID}\": Updating
${XCONFIG}"
        sed -e 's|BusID.*|BusID          '\"${NEWBUSID}\"'|' -i ${XCONFIG}
fi
```

Then create an entry for your update script in `/etc/rc.d/rc3.d` so the script is invoked as root on boot.

Troubleshooting

- You can set persistence mode using `nvidia-smi` so the output of the command is faster when you need to query cards. To set persistence mode, run `nvidia-smi -pm 1`. If the VM is restarted, the mode setting goes away. You can always script the mode setting to run upon startup.
- If you updated the NVIDIA CUDA drivers to the latest version and find RDMA connectivity is no longer working, reinstall the RDMA drivers to reestablish that connectivity.
- During installation of LIS, if a certain CentOS/RHEL OS version (or kernel) is not supported for LIS, an error “Unsupported kernel version” is thrown. Report this error along with the OS and kernel versions.
- If jobs are interrupted by ECC errors on the GPU (either correctable or uncorrectable), first check to see if the GPU meets any of Nvidia's RMA criteria for ECC errors. If the GPU is eligible for RMA, contact support about getting it serviced; otherwise, reboot your VM to reattach the GPU as described here. Less invasive methods, such as `nvidia-smi -r`, don't work with the virtualization solution deployed in Azure.

Next steps

- To capture a Linux VM image with your installed NVIDIA drivers, see How to generalize and capture a Linux virtual machine.

Last updated on 12/02/2025

Install NVIDIA GPU drivers on N-series VMs running Windows

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

To take advantage of the GPU capabilities of Azure N-series VMs backed by NVIDIA GPUs, you must install NVIDIA GPU drivers. The NVIDIA GPU Driver Extension installs appropriate NVIDIA CUDA or GRID drivers on an N-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the NVIDIA GPU Driver Extension documentation for supported operating systems and deployment steps.

If you choose to install NVIDIA GPU drivers manually, this article provides supported operating systems, drivers, and installation and verification steps. Manual driver setup information is also available for Linux VMs.

For basic specs, storage capacities, and disk details, see GPU Windows VM sizes.

> ⚠ **Warning**
>
> Installing NVIDIA drivers using methods other than those outlined in this guide may result in failure of the intended driver installation. To ensure proper functionality and support, please follow only the installation steps and use the driver versions specified in this documentation.

Supported operating systems and drivers

NVIDIA Tesla (CUDA) drivers

> ⓘ **Note**
>
> The Azure NVads A10 v5 VMs only support vGPU 17.x or higher driver version. The vGPU driver for the A10 SKU is a unified driver that supports both graphics and compute workloads.

NVIDIA Tesla (CUDA) drivers for all NC* and ND-*series VMs (optional for NV*-series) are generic and not Azure specific. For the latest drivers, visit the NVIDIA website.

> 💡 **Tip**

As an alternative to manual CUDA driver installation on a Windows Server VM, you can deploy an Azure Data Science Virtual Machine image. The DSVM editions for Windows Server 2016 preinstall NVIDIA CUDA drivers, the CUDA Deep Neural Network Library, and other tools.

NVIDIA GRID/vGPU drivers

> **Note**
>
> vGPU18 is available for the NCasT4_v3-series.
>
> vGPU18 is now available for the NVadsA10_v5-series in **Public, China, and Azure Government regions only**.

> **Note**
>
> For Azure NVads A10 v5 VMs we recommend customers to always be on the latest driver version. The latest NVIDIA major driver branch(n) is only backward compatbile with the previous major branch(n-1). For eg, vGPU 17.x is backward compatible with vGPU 16.x only. Any VMs still runnig n-2 or lower may see driver failures when the latest drive branch is rolled out to Azure hosts.
>
> NVs_v3 VMs only support **vGPU 16 or lower** driver version.
>
> Windows server 2019 support is only available till vGPU 16.x.

Microsoft redistributes NVIDIA GRID driver installers for NV, NVv3, and NVads A10 v5-series VMs used as virtual workstations or for virtual applications. Install only these GRID drivers on Azure NV-series VMs, only on the operating systems listed in the following table. These drivers include licensing for GRID Virtual GPU Software in Azure. You don't need to set up a NVIDIA vGPU software license server.

The GRID drivers redistributed by Azure don't work on non-NV series VMs like NCv2, NCv3, ND, and NDv2-series VMs. The one exception is the NCas_T4_V3 VM series where the GRID drivers enable the graphics functionalities similar to NV-series.

The Nvidia extension always installs the latest driver.

For Windows 11 up to and including 25H2, Windows 10 up to and including 22H2, Server 2019, Server 2022, Server 2025:

- GRID 18.6 (573.76) (.exe)

The following links to previous versions are provided to support dependencies on older driver versions.

For Windows Server 2016 1607, 1709:

- GRID 14.1 (512.78) (.exe) is the last supported driver from NVIDIA. The newer 15.x and above don't support Windows Server 2016.

For Windows Server 2012 R2:

- GRID 13.1 (472.39) (.exe)
- GRID 13 (471.68) (.exe)

For links to all previous Nvidia GRID driver versions, visit GitHub.

Driver installation

1. Connect by Remote Desktop to each N-series VM.
2. Download, extract, and install the supported driver for your Windows operating system.

After GRID driver installation on a VM, a restart is required. After CUDA driver installation, a restart is not required.

Verify driver installation

Please note that the NVIDIA Control panel is only accessible with the GRID driver installation. If you have installed CUDA drivers then the NVIDIA control panel will not be visible.

You can verify driver installation in Device Manager.

> **Note**
>
> If you're running Windows 10 build 1903 or higher, dxdiag shows no information in the 'Display' tab. Use the 'Save All Information' option at the bottom and the output file shows the information related to NVIDIA GPU.

To query the GPU device state, run the nvidia-smi command-line utility installed with the driver.

1. Open a command prompt and change to the **C:\Program Files\NVIDIA Corporation\NVSMI** directory.
2. Run `nvidia-smi`. If the driver is installed, Nvidia SMI will list the **GPU-Util** as N/A until you run a GPU workload on the VM.

RDMA network connectivity

RDMA network connectivity can be enabled on RDMA-capable N-series VMs such as NC24r deployed in the same availability set or in a single placement group in a virtual machine scale set. The HpcVmDrivers extension must be added to install Windows network device drivers that enable RDMA connectivity. To add the VM extension to an RDMA-enabled N-series VM, use Azure PowerShell cmdlets for Azure Resource Manager.

To install the latest version 1.1 HpcVMDrivers extension on an existing RDMA-capable VM named myVM in the West US region:

PowerShell

```
Set-AzVMExtension -ResourceGroupName "myResourceGroup" -Location "westus" -VMName
"myVM" -ExtensionName "HpcVmDrivers" -Publisher "Microsoft.HpcCompute" -Type
"HpcVmDrivers" -TypeHandlerVersion "1.1"
```

For more information, see Virtual machine extensions and features for Windows.

The RDMA network supports Message Passing Interface (MPI) traffic for applications running with Microsoft MPI or Intel MPI 5.x.

Next steps

- Developers building GPU-accelerated applications for the NVIDIA Tesla GPUs can also download and install the latest CUDA Toolkit. For more information, see the CUDA Installation Guide.

Last updated on 12/19/2025

Install AMD GPU drivers on NVads V710-series Linux VMs

Applies to: ✔ Linux VMs

> **Important**
>
> To align with inclusive language practices, we've replaced the term "blacklist" with "blocklist" throughout this documentation. This change reflects our commitment to avoiding terminology that might carry unintended negative connotations or perceived racial bias. However, in code snippets and technical references where "blacklist" is part of established syntax or tooling (for example, configuration files, command-line parameters), the original term is retained to preserve functional accuracy. This usage is strictly technical and doesn't imply any discriminatory intent.

NVads V710-series

To leverage the GPU capabilities of Azure's new NVads V710-series virtual machines running Linux, you'll need to install AMD GPU drivers. The AMD GPU driver extension streamlines this process by automating driver installation for NVv710-series VMs. You can manage the extension via the Azure portal, Azure PowerShell, or Azure Resource Manager (ARM) templates. For details on supported operating systems and deployment steps, see AMD GPU Driver Extension documentation.

The marketplace image comes preloaded with the AMD GPU driver, helping accelerate VM setup. This guide explains how to install AMD GPU drivers on Azure NVads V710-series Linux virtual machines (VMs). It covers both automated and manual installation methods specifically for **Ubuntu**.

ROCm

> **Note**
>
> Currently, Azure provides installation instructions for:
>
> - Ubuntu 22.04
> - Ubuntu 24.04
>
> For other Linux distributions, see:

- Quick start installation guide - ROCm installation (Linux)
- ROCm release history - ROCm Documentation

Install the AMD Linux Driver to leverage the full capabilities of the AMD Radeon PRO V710 GPU on an NVv5-V710 GPU Linux instance in Microsoft Azure. The sections that follow provide detailed instructions for installing the Linux driver and running inference workloads using ROCm on this instance type.

Quick start options

Option 1: Use the AMD GPU driver extension

The simplest method is using the AMD GPU Driver Extension, which automates driver installation for NVv710-series VMs. You can deploy this extension through:

- Azure portal
- Azure PowerShell
- Azure Resource Manager templates

Option 2: Use pre-configured marketplace image

A marketplace image is available with pre-installed AMD GPU drivers, allowing for faster VM deployment.

Option 3: Manual installation

Follow these instructions for manual driver installation and configuration.

ROCm driver installation

Prerequisites

System requirements:

- Disk size must exceed 64GB for optimal performance
- Supported distributions: Ubuntu 22.04 or Ubuntu 24.04
- Virtual Function Device ID: 7461 (AMD Radeon PRO V710 GPU)

Step 1: Verify your system

Follow these steps to verify that your GPU card is detected on your system.

1. Check your Linux distribution:

   ```Bash
   cat /etc/*release
   ```

2. Check your kernel version:

   ```Bash
   uname -srmv
   ```

3. Verify your GPU card is detected:

   ```Bash
   sudo lspci -d 1002:7461
   ```

   You should see output similar to:

   ```
   c3:00.0 Display controller: Advanced Micro Devices, Inc. [AMD/ATI] Device 7461
   ```

Step 2: Install the driver

The driver installation commands are slightly different depending on whether you're running Ubuntu 22.04 or 24.04.

For Ubuntu 22.04

```Bash
sudo apt update
sudo apt install "linux-headers-$(uname -r)" "linux-modules-extra-$(uname -r)"
sudo apt install python3-setuptools python3-wheel
sudo usermod -a -G render,video $LOGNAME
wget https://repo.radeon.com/amdgpu-install/7.0.1/ubuntu/jammy/amdgpu-
install_7.0.1.70001-1_all.deb
sudo apt install ./amdgpu-install_7.0.1.70001-1_all.deb
```

```
sudo apt update
sudo apt install amdgpu-dkms rocm
```

For Ubuntu 24.04

Bash

```
sudo apt update
sudo apt install "linux-headers-$(uname -r)" "linux-modules-extra-$(uname -r)"
sudo apt install python3-setuptools python3-wheel
sudo usermod -a -G render,video $LOGNAME # Add the current user to the render and
video groups
wget https://repo.radeon.com/amdgpu-install/7.0.1/ubuntu/noble/amdgpu-
install_7.0.1.70001-1_all.deb
sudo apt install ./amdgpu-install_7.0.1.70001-1_all.deb
sudo apt update
sudo apt install amdgpu-dkms rocm
```

Step 3: Load and verify the driver

Follow these steps to load and verify the driver.

1. Load the driver:

   Bash

   ```
   sudo modprobe amdgpu
   ```

2. Check that the driver loaded successfully:

   Bash

   ```
   sudo dmesg | grep amdgpu
   ```

3. Verify driver status with AMD-SMI:

   Bash

   ```
   amd-smi monitor
   ```

Step 4: Enable automatic loading on reboot

Follow these steps to enable automatic loading on reboot.

1. Search for blocklist entries:

```bash
grep amdgpu /etc/modprobe.d/* -rn
```

2. If the driver is blocklisted, remove the blocklist:

```bash
sudo nano /etc/modprobe.d/blacklist.conf
```

3. Delete the line containing `blacklist amdgpu`, then update initramfs:

```bash
sudo update-initramfs -uk all
```

4. Reboot to apply changes:

```bash
sudo reboot
```

Graphics and ROCm installation

This section covers installing the AMD driver for graphics workloads with ROCm libraries and development tools.

Prerequisites

System requirements:

- Ubuntu 22.04 with kernel 6.5
- Disk size greater than 64GB
- Desktop environment (for graphics workloads, use Ubuntu Desktop ISO)

Pre-installation steps

Complete the following pre-installation steps.

1. Update package list:

```bash
```

```
sudo apt update
```

2. Install Python packages:

```Bash
sudo apt install python3-setuptools python3-wheel
```

3. Add user to required groups:

```Bash
sudo usermod -a -G render,video $LOGNAME
```

4. Install kernel headers:

```Bash
sudo apt install "linux-headers-$(uname -r)" "linux-modules-extra-$(uname -r)"
```

Blocklist default driver

Follow these steps to blocklist the default driver.

1. Check if the driver is already blocklisted:

```Bash
grep amdgpu /etc/modprobe.d/* -rn
```

2. If not blocklisted, add it:

```Bash
sudo vim /etc/modprobe.d/blacklist.conf
```

3. Add this line:

```
blacklist amdgpu
```

4. Apply changes:

Bash

```
sudo update-initramfs -uk all
```

5. Reboot the system:

Bash

```
sudo reboot
```

Install AMD driver with graphics support

Follow these steps to install the AMD driver with graphics support.

1. Upgrade the system:

Bash

```
sudo apt upgrade
```

2. Download the installer:

Bash

```
wget -N -P /tmp/ https://repo.radeon.com/amdgpu-
install/6.1.4/ubuntu/jammy/amdgpu-install_6.1.60104-1_all.deb
```

3. If a previous driver exists, remove it:

Bash

```
sudo amdgpu-uninstall
sudo apt remove amdgpu-install --purge
```

4. Install the new driver:

Bash

```
sudo apt-get install /tmp/amdgpu-install_6.1.60104-1_all.deb
sudo amdgpu-install --usecase=workstation,rocm,amf --opencl=rocr --vulkan=pro --
-no-32 --accept-eula
```

5. Load the driver:

Bash

```
sudo modprobe amdgpu
```

6. Verify installation:

```Bash
sudo dmesg | grep amdgpu
```

To remove a blocklist, see Enable automatic loading on reboot.

X11 remote server configuration

After installing the graphics driver, follow these steps to configure a virtual display with hardware acceleration for remote access.

Step 1: Install required packages

```Bash
sudo apt install net-tools
sudo apt install x11vnc
```

Step 2: Configure GDM3

Follow these steps to configure GDM3.

1. Edit the GDM3 configuration:

```Bash
sudo vim /etc/gdm3/custom.conf
```

2. Modify to include:

```ini
[daemon]
AutomaticLoginEnable=true
AutomaticLogin=your_username
WaylandEnable=false
```

3. Restart GDM3:

Bash

```
sudo systemctl restart gdm3
```

Step 3: Configure X11

Follow these steps to configure X11.

1. Get your GPU's Bus ID:

Bash

```
lspci -d 1002: | awk '{print $1}'
```

2. Convert the hex Bus ID to decimal format. For example, `3a9e:00:00.0` becomes `3841536`.

3. Edit the X configuration file `/usr/share/X11/xorg.conf.d/00-amdgpu.conf`:

```
Section "Device"
    Identifier "Card0"
    Driver "amdgpu"
    BusID "PCI:3841536:0:0"
EndSection

Section "Screen"
    Identifier "Screen0"
    Device "Card0"
    Monitor "Monitor0"
EndSection
```

4. Edit `/usr/share/X11/xorg.conf.d/10-amdgpu.conf`:

```
Section "OutputClass"
    Identifier "Card0"
    MatchDriver "amdgpu"
    Driver "amdgpu"
    Option "PrimaryGPU" "yes"
EndSection
```

5. Reboot and load the driver:

Bash

```
sudo reboot
```

6. After reboot, run the following commands:

```Bash
sudo systemctl stop gdm
sudo modprobe amdgpu
sudo systemctl start gdm
```

Step 4: Start VNC server

To start the server, run the following command.

```Bash
x11vnc --forever -find
```

> ⓘ **Note**
>
> X11 configuration only works with Ubuntu Desktop images, not Server images.

Troubleshooting

Downgrade to kernel 6.5

If you're running kernel 6.8, you can downgrade to 6.5 for compatibility by following these steps.

1. Check loaded kernels:

```Bash
dpkg --list | egrep -i --color 'linux-image|linux-headers|linux-modules' | awk '{ print $2 }'
```

2. Install kernel 6.5:

```Bash
sudo apt install linux-image-6.5.0-1025-azure
```

3. Remove kernel 6.8:

```
Bash
sudo apt purge linux-headers-6.8.0-1025-azure linux-image-6.8.0-1025-azure
linux-modules-6.8.0-1025-azure
```

4. Edit GRUB:

```
Bash
sudo vim /etc/default/grub
```

5. Set kernel 6.5 as default:

```

GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-1025-azure"
```

6. Update GRUB and reboot:

```
Bash
sudo update-grub
sudo reboot
```

7. Verify kernel version:

```
Bash
uname -a
```

Uninstalling the AMD GPU driver

To completely remove the AMD GPU driver, run the following commands:

```
Bash
dkms status
sudo amdgpu-install --uninstall
sudo amdgpu-uninstall
sudo apt autoremove --purge amdgpu-install
sudo reboot
```

Verify removal:

Bash

```
dkms status
```

Last updated on 10/16/2025

Install AMD GPU drivers on N-series VMs running Windows

Applies to: Windows VMs ✔ Flexible scale sets

> **Note**
>
> The AMD GPU Driver Extension installs AMD GPU drivers on a NVads V710 v5 Series, NGads V620 Series and NVv4-series. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the AMD GPU Driver Extension documentation for supported operating systems and deployment steps.

NVads V710 v5 Series

To take advantage of the GPU capabilities of the new Azure NVads V710 v5 series VMs running Windows, AMD GPU drivers must be installed. The AMD GPU Driver Extension installs AMD GPU drivers on a NVads V710 v5-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the AMD GPU Driver Extension documentation for supported operating systems and deployment steps.

If you choose to install AMD GPU drivers manually, this article provides supported operating systems, drivers, and installation and verification steps.

Only GPU drivers published by Microsoft are supported on NVads V710 v5 series VMs. Don't install GPU drivers from any other source.

For basic specs, storage capacities, and disk details, see GPU Windows VM sizes.

Expand table

OS	Driver
Windows 10 22h2 Windows 11 23H2, 24H2, 25h2	25.Q4 (.exe)
Windows Server 2025, Windows 11 EMS	25.Q4 (.exe)

OS	Driver
Windows Server 2022	25.Q4 (.exe)

VM Creation

Create the VMs using CLI. (Azure AMD GPU driver extensions don't support NVads V710 v5 Series during preview)

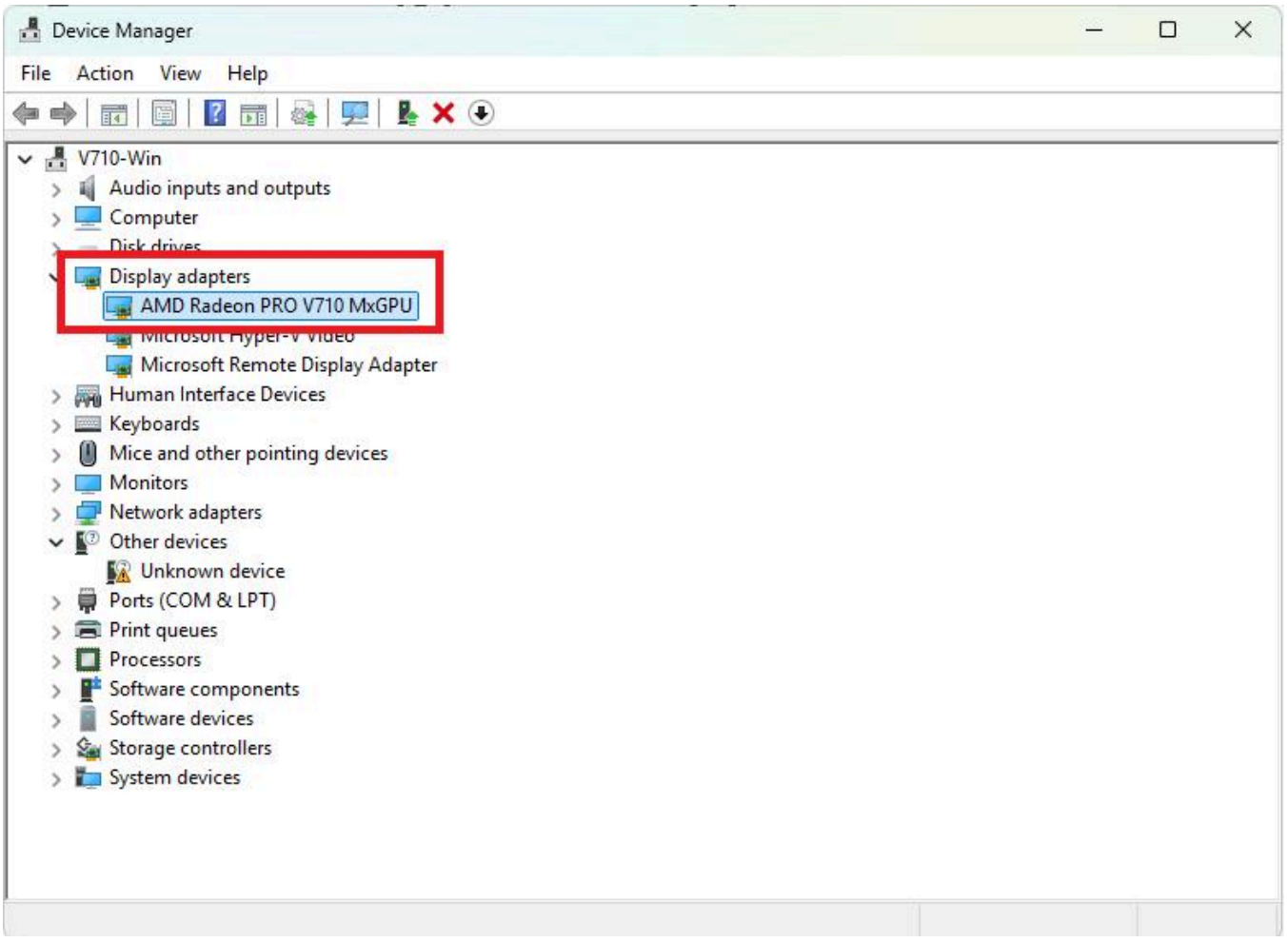
- Review the CLI VM creation documentation.

Driver installation

1. Connect by Remote Desktop to each NVads V710 v5-series VM.
2. Download the EXE file to a local drive.
3. If you need to uninstall the previous driver version, run "setup.exe -factoryresetinstall" from a command line.
4. For a first-time driver installation, double-click or run “setup.exe” from a command line.
5. Reboot the VM.

Verify driver installation

You can verify driver installation in Device Manager. The following example shows successful configuration of the Radeon Pro V710 card on an NVads V710 v5 VM. The exact driver date and version will depend on the driver package released.



NGads V620 Series

The AMD Software: Cloud Edition drivers must be installed to take advantage of the GPU capabilities of Azure NGads V620 Series VMs.

Requirements

Expand table

OS	Driver
Windows 11 64-bit version 21H2, 22H2, 23H2, 24H2 Windows 10 64-bit version 20H2, 21H2, 22H2, 23H2, 24H2	25.1.1
Windows 11 EMS 64-bit 21H2, 22H2, 23H2, 24H2 Windows Server 2022 64-bit	25.1.1
Windows 10 EMS 64-bit 21H2, 22H2, 23H2, 24H2 Windows Server 2019 LTSC (version 1809)	25.1.1

VM Creation

Create the VMs using CLI. (Azure AMD GPU driver extensions don't support NGads V620 Series during preview)

1. Review the CLI VM creation documentation.

Driver installation

1. Connect by Remote Desktop to each NGads V620-series VM
2. Download the EXE file to a local drive
3. If you need to uninstall the previous driver version, run "setup.exe -factoryresetinstall" from a command line
4. For a first-time driver installation, double-click or run “setup.exe” from a command line
5. Reboot the VM

Verify driver installation

1. You can verify driver installation in Device Manager. The following example shows successful configuration of the Radeon Pro V620 card on an Azure NGads V620 VM. The exact driver date and version will depend on the driver package released.

NVv4 Series

To take advantage of the GPU capabilities of the new Azure NVv4 series VMs running Windows, AMD GPU drivers must be installed. The AMD GPU Driver Extension installs AMD GPU drivers on a NVv4-series VM. Install or manage the extension using the Azure portal or tools such as Azure PowerShell or Azure Resource Manager templates. See the AMD GPU Driver Extension documentation for supported operating systems and deployment steps.

If you choose to install AMD GPU drivers manually, this article provides supported operating systems, drivers, and installation and verification steps.

Only GPU drivers published by Microsoft are supported on NVv4 VMs. Don't install GPU drivers from any other source.

For basic specs, storage capacities, and disk details, see GPU Windows VM sizes.

Supported operating systems and drivers

Expand table

OS	Driver
Windows 11 64-bit 21H2, 22H2, 23H2, 24H2 Windows 10 64-bit 21H2, 22H2, 20H2	23Q3 MR 4.1 Hotfix (.exe)
Windows Server 2022, Windows 11 EMS	23.Q3 MR 4.1 Hotfix (.exe)
Windows Server 2019, Windows 10 EMS	23.Q3 MR 4.1 Hotfix (.exe)

Previous supported driver versions for Windows builds up to 1909 are 20.Q4-1 (.exe) and 21.Q2-1 (.exe)

> **Note**
>
> If you use build 1903/1909 then you may need to update the following group policy for optimal performance. These changes are not needed for any other Windows builds.

> [Computer Configuration->Policies->Windows Settings->Administrative Templates->Windows Components->Remote Desktop Services->Remote Desktop Session Host->Remote Session Environment], set the Policy [Use WDDM graphics display driver for Remote Desktop Connections] to Disabled.

Driver installation

> **Note**
>
> Follow these steps if you see "Error 184 - AMD Installer cannot continue due to an unsupported Operating System" error on Windows 10 EMS / Windows 11 EMS.

> Go to C:\AMD\AMD Software Azure NVv4 Guest Driver 23Q3\Packages\Drivers\Display\WT6A_INF Right click and install on the *.inf file.

> For Windows10 EMS: u9397288.inf

> For Windows11 EMS: u2397344.inf

1. Connect by Remote Desktop to each NVv4-series VM.

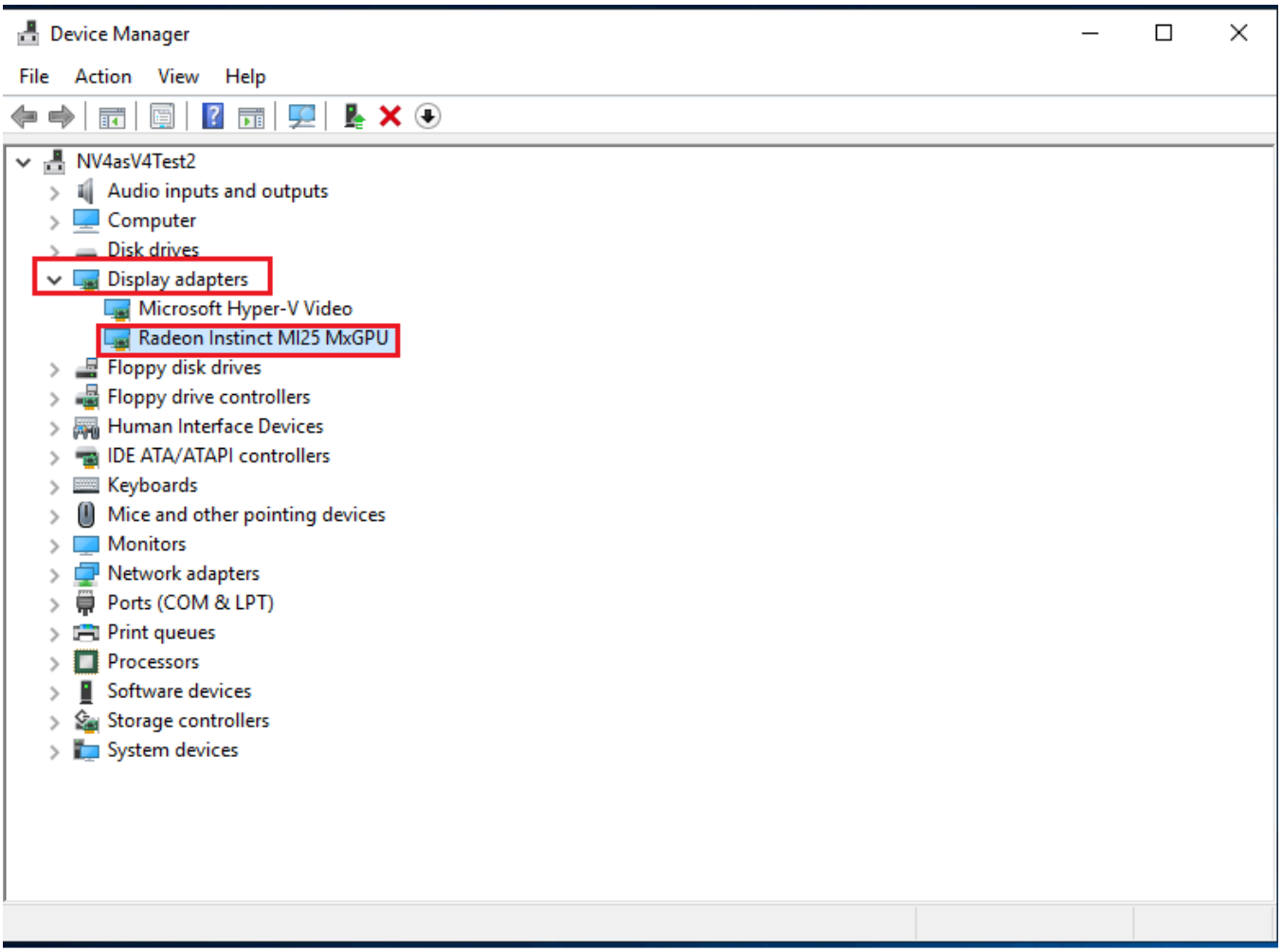
2. If you need to uninstall the previous driver version, download the AMD cleanup utility. Don't use the utility that comes with the previous version of the driver.

3. Download and install the latest driver.

4. Reboot the VM.

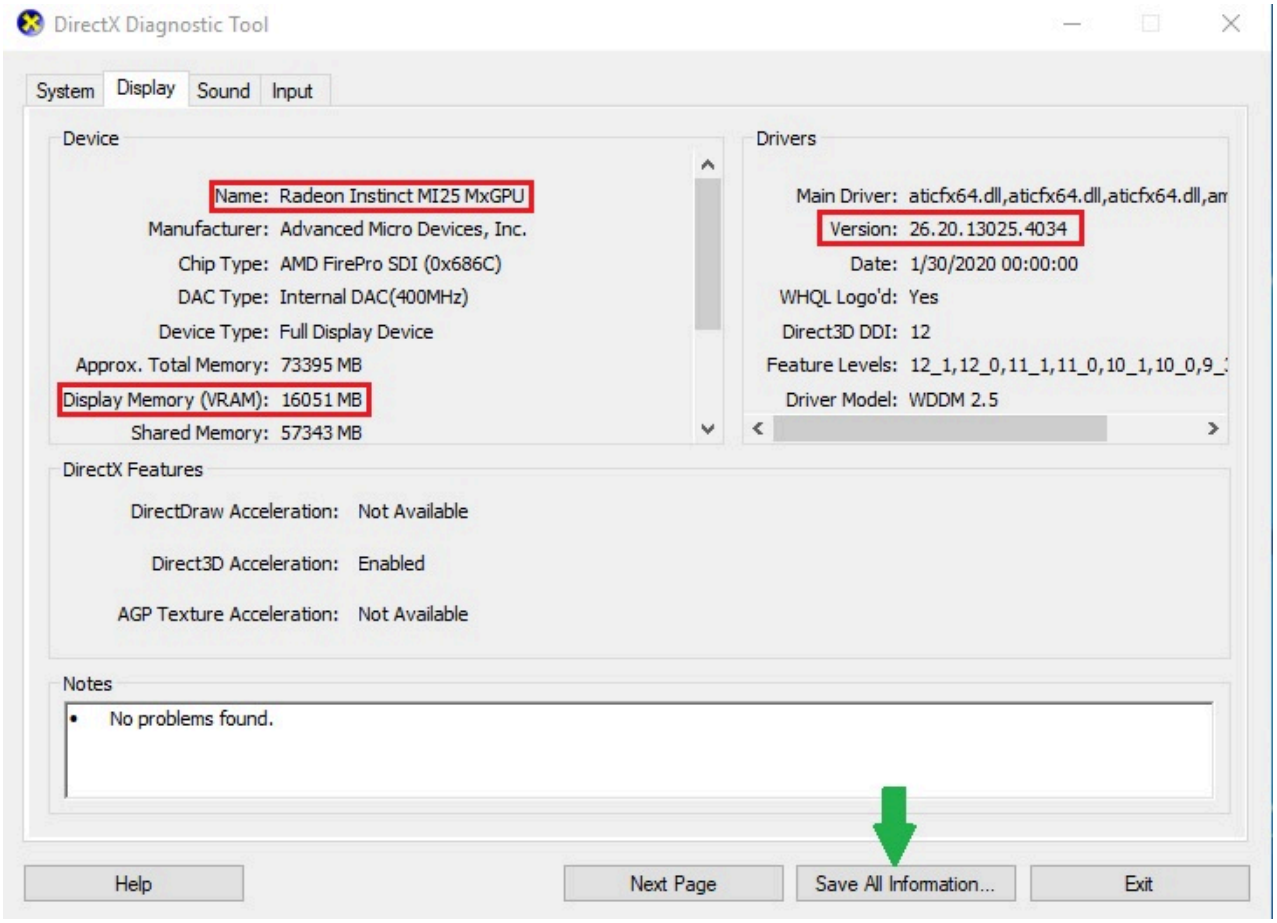
Verify driver installation

1. You can verify driver installation in Device Manager. The following example shows successful configuration of the Radeon Instinct MI25 card on an Azure NVv4 VM.



1. You can use dxdiag to verify the GPU display properties including the video RAM. The following example shows a 1/2 partition of the Radeon Instinct MI25 card on an Azure

NVv4 VM.



2. If you're running Windows 10 build 1903 or higher, dxdiag shows no information in the 'Display' tab. Use the 'Save All Information' option at the bottom and the output file shows the information related to AMD MI25 GPU.

```
---------------
Display Devices
---------------
           Card name: Radeon Instinct MI25 MxGPU
        Manufacturer: Advanced Micro Devices, Inc.
           Chip type: AMD FirePro SDI (0x686C)
            DAC type: Internal DAC(400MHz)
         Device Type: Full Device
          Device Key: Enum\PCI\VEN_1002&DEV_686C&SUBSYS_0C351002&REV_00
       Device Status: 0180600A [DN_DRIVER_LOADED|DN_STARTED|DN_DISABLEABLE|DN_REMOVABLE|DN_NT_ENUMERATOR|DN_NT_DRIVER]
 Device Problem Code: No Problem
Driver Problem Code: Unknown
     Display Memory: 73395 MB
    Dedicated Memory: 16051 MB
       Shared Memory: 57343 MB
        Current Mode: 1920 x 1080 (32 bit) (32Hz)
         HDR Support: Unknown
    Display Topology: Unknown
Display Color Space: DXGI_COLOR_SPACE_RGB_FULL_G22_NONE_P709
     Color Primaries: Red(0.000000,0.000000), Green(0.000000,0.000000), Blue(0.000000,0.000000), White Point(0.000000,0.000000)
   Display Luminance: Min Luminance = 0.000000, Max Luminance = 0.000000, MaxFullFrameLuminance = 0.000000
         Driver Name: C:\windows\System32\DriverStore\FileRepository\c0351510.inf_amd64_072efa16e4548b4c\B351427\aticfx64.dll,
                      C:\windows\System32\DriverStore\FileRepository\c0351510.inf_amd64_072efa16e4548b4c\B351427\aticfx64.dll,
                      C:\windows\System32\DriverStore\FileRepository\c0351510.inf_amd64_072efa16e4548b4c\B351427\aticfx64.dll,
                      C:\windows\System32\DriverStore\FileRepository\c0351510.inf_amd64_072efa16e4548b4c\B351427\amdxc64.dll
Driver File Version: 26.20.13025.4034 (English)
      Driver Version: 26.20.13025.4034
         DDI Version: 12
      Feature Levels: 12_1,12_0,11_1,11_0,10_1,10_0,9_3,9_2,9_1
        Driver Model: WDDM 2.5
Graphics Preemption: Primitive
  Compute Preemption: DMA
           Miracast: Not Supported
Hybrid Graphics GPU: Not Supported
      Power P-states: Not Supported
      Virtualization: Not Supported
          Block List: GPU_PV_HIGH_SECURITY
  Catalog Attributes: Universal:False Declarative:False
   Driver Attributes: Final Retail
   Driver Date/Size: 1/30/2020 12:00:00 AM, 1979552 bytes
        WHQL Logo'd: Yes
     WHQL Date Stamp: Unknown
   Device Identifier: {D7B71EE2-2B2C-11CF-707C-D41EBBC2D735}
           Vendor ID: 0x1002
           Device ID: 0x686C
           SubSys ID: 0x0C351002
         Revision ID: 0x0000
  Driver Strong Name: oem2.inf:cb0ae414494e7940:ati2mtag_R7500DS:26.20.13025.4034:pci\ven_1002&dev_686c&rev_00
      Rank Of Driver: 00D12000
         Video Accel: Unknown
         DXVA2 Modes:
      Deinterlace Caps: n/a
        D3D9 Overlay: Not Supported
             DXVA-HD: Not Supported
        DDraw Status: Not Available
          D3D Status: Enabled
          AGP Status: Not Available
       MPO MaxPlanes: 1
            MPO Caps: Not Supported
         MPO Stretch: Not Supported
     MPO Media Hints: Not Supported
         MPO Formats: Not Supported
    PanelFitter Caps: Not Supported
PanelFitter Stretch: Not Supported
```

Last updated on 03/18/2026

Migration Guide for GPU Compute Workloads in Azure

Article • 03/17/2025

As more powerful GPUs become available in the marketplace and in Microsoft Azure datacenters, we recommend re-assessing the performance of your workloads and considering migrating to newer GPUs.

For the same reason, as well as to maintain a high-quality and reliable service offering, Azure periodically retires the hardware that powers older VM sizes. The first group of GPU products to be retired in Azure are the original NC, NC v2 and ND-series VMs, powered by NVIDIA Tesla K80, P100, and P40 datacenter GPU accelerators respectively. These products will be retired on August 31st 2023, and the oldest VMs in this series launched in 2016.

Since then, GPUs have made incredible strides alongside the entire deep learning and HPC industry, typically exceeding a doubling in performance between generations. Since the launch of NVIDIA K80, P40, and P100 GPUs, Azure has shipped multiple newer generations and categories of VM products geared at GPU-accelerated compute and AI, based around NVIDIA's T4, V100, and A100 GPUs, and differentiated by optional features such as InfiniBand-based interconnect fabrics. These are all options we encourage customers to explore as migration paths.

In most cases, the dramatic increase in performance offered by newer generations of GPUs lowers overall TCO by decreasing the duration of job, for burstable jobs- or reducing the quantity of overall GPU-enabled VMs required to cover a fixed-size demand for compute resources, even though costs per GPU-hour may vary. In addition to these benefits, customers may improve Time-to-Solution via higher-performing VMs, and improve the health and supportability of their solution by adopting newer software, CUDA runtime, and driver versions.

Migration vs. Optimization

Azure recognizes that customers have a multitude of requirements that may dictate the selection of a specific GPU VM product, including GPU architectural considerations, interconnects, TCO, Time to Solution, and regional availability based on compliance locality or latency requirements- and some of these even change over time.

At the same time, GPU acceleration is a new and rapidly evolving area.

Thus, there is no true one-size fits-all guidance for this product area, and a migration is a perfect time to re-evaluate potentially dramatic changes to a workload- like moving from a clustered deployment model to a single large 8-GPU VM or vice versa, leveraging reduced precision datatypes, adopting features like Multi-Instance GPU, and much more.

These sorts of considerations- when made the context of already dramatic per-generation GPU performance increases, where a feature such as the addition of TensorCores can increase performance by an order of magnitude, are extremely workload-specific.

Combining migration with application re-architecture can yield immense value and improvement in cost and time-to-solution.

However, these sorts of improvements are beyond the scope of this document, which aims to focus on direct equivalency classes for generalized workloads that may be run by customers today, to identify the most similar VM options in both price *and* performance per GPU to existing VM families undergoing retirement.

Thus, this document assumes that the user may not have any insight or control over workload-specific properties like the number of required VM instances, GPUs, interconnects, and more.

Recommended Upgrade Paths

NC-Series VMs featuring NVIDIA K80 GPUs

The NC (v1)-Series VMs are Azure's oldest GPU-accelerated compute VM type, powered by 1 to 4 NVIDIA Tesla K80 datacenter GPU accelerators paired with Intel Xeon E5-2690 v3 (Haswell) processors. Once a flagship VM type for demanding AI, ML, and HPC applications, they remained a popular choice late into the product lifecycle (particularly via NC-series promotional pricing) for users who valued having a very low absolute cost per GPU-hour over GPUs with higher throughput-per-dollar.

Today, given the relatively low compute performance of the aging NVIDIA K80 GPU platform, in comparison to VM series featuring newer GPUs, a popular use case for the NC-series is real-time inference and analytics workloads, where an accelerated VM must be available in a steady state to serve request from applications as they arrive. In these cases the volume or batch size of requests may be insufficient to benefit from more performant GPUs. NC VMs are also popular for developers and students learning about, developing for, or experimenting with GPU acceleration, who need an inexpensive

cloud-based CUDA deployment target upon which to iterate that doesn't need to perform to production levels.

In general, NC-Series customers should consider moving directly across from NC sizes to NC T4 v3 sizes, Azure's new GPU-accelerated platform for light workloads powered by NVIDIA Tesla T4 GPUs.

Expand table

Current VM Size	Target VM Size	Difference in Specification
Standard_NC6 Standard_NC6_Promo	Standard_NC4as_T4_v3 or Standard_NC8as_T4	CPU: Intel Haswell vs AMD Rome GPU count: 1 (same) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 4 (-2) or 8 (+2) Memory GiB: 16 (-40) or 56 (same) Temp Storage (SSD) GiB: 180 (-160) or 360 (+20) Max data disks: 8 (-4) or 16 (+4) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC12 Standard_NC12_Promo	Standard_NC16as_T4_v3	CPU: Intel Haswell vs AMD Rome GPU count: 1 (-1) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 16 (+4) Memory GiB: 110 (-2) Temp Storage (SSD) GiB: 360 (-320) Max data disks: 48 (+16) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC24 Standard_NC24_Promo	Standard_NC64as_T4_v3*	CPU: Intel Haswell vs AMD Rome GPU count: 4 (same) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 64 (+40) Memory GiB: 440 (+216) Temp Storage (SSD) GiB: 2880 (+1440) Max data disks: 32 (-32) Accelerated Networking: Yes (+) Premium Storage: Yes (+)

Current VM Size	Target VM Size	Difference in Specification
Standard_NC24r Standard_NC24r_Promo	Standard_NC64as_T4_v3*	CPU: Intel Haswell vs AMD Rome GPU count: 4 (same) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 64 (+40) Memory GiB: 440 (+216) Temp Storage (SSD) GiB: 2880 (+1440) Max data disks: 32 (-32) Accelerated Networking: Yes (+) Premium Storage: Yes (+) InfiniBand interconnect: No

NC v2-Series VMs featuring NVIDIA Tesla P100 GPUs

The NC v2-series virtual machines are a flagship platform originally designed for AI and Deep Learning workloads. They offered excellent performance for Deep Learning training, with per-GPU performance roughly 2x that of the original NC-Series and are powered by NVIDIA Tesla P100 GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. Like the NC and ND -Series, the NC v2-Series offers a configuration with a secondary low-latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

In general, NCv2-Series customers should consider moving directly across to NC A100 v4 sizes, Azure's new GPU-accelerated platform powered by NVIDIA Ampere A100 PCIe GPUs.

Expand table

Current VM Size	Target VM Size	Difference in Specification
Standard_NC6s_v2	Standard_NC24ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 1 (same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generation) GPU memory (GiB per GPU): 80 (+64) vCPU: 24 (+18) Memory GiB: 220 (+108) Temp Storage (SSD) GiB: 1123 (+387) Max data disks: 12 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)

Current VM Size	Target VM Size	Difference in Specification
Standard_NC12s_v2	Standard_NC48ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 2 (same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generations) GPU memory (GiB per GPU): 80 (+64) vCPU: 48 (+36) Memory GiB: 440 (+216) Temp Storage (SSD) GiB: 2246 (+772) Max data disks: 24 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC24s_v2	Standard_NC96ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 4 (same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generations) GPU memory (GiB per GPU): 80 (+64) vCPU: 96 (+72) Memory GiB: 880 (+432) Temp Storage (SSD) GiB: 4492 (+1544) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC24rs_v2	Standard_NC96ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 4 (Same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generations) GPU memory (GiB per GPU): 80 (+64) vCPU: 96 (+72) Memory GiB: 880 (+432) Temp Storage (SSD) GiB: 4492 (+1544) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+) InfiniBand interconnect: No (-)

ND-Series VMs featuring NVIDIA Tesla P40 GPUs

The ND-series virtual machines are a midrange platform originally designed for AI and Deep Learning workloads. They offered excellent performance for batch inferencing via improved single-precision floating point operations over their predecessors and are powered by NVIDIA Tesla P40 GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. Like the NC and NC v2-Series, the ND-Series offers a configuration with a secondary low-

latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

Expand table

Current VM Size	Target VM Size	Difference in Specification
Standard_ND6	Standard_NC4as_T4_v3 or Standard_NC8as_T4_v3	CPU: Intel Broadwell vs AMD Rome GPU count: 1 (same) GPU generation: NVIDIA Pascal vs. Turing (+1 generation) GPU memory (GiB per GPU): 16 (-8) vCPU: 4 (-2) or 8 (+2) Memory GiB: 16 (-40) or 56 (-56) Temp Storage (SSD) GiB: 180 (-552) or 360 (-372) Max data disks: 8 (-4) or 16 (+4) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_ND12	Standard_NC16as_T4_v3	CPU: Intel Broadwell vs AMD Rome GPU count: 1 (-1) GPU generation: NVIDIA Pascal vs. Turing (+1 generations) GPU memory (GiB per GPU): 16 (-8) vCPU: 16 (+4) Memory GiB: 110 (-114) Temp Storage (SSD) GiB: 360 (-1,114) Max data disks: 48 (+16) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_ND24	Standard_NC64as_T4_v3*	CPU: Intel Broadwell vs AMD Rome GPU count: 4 (same) GPU generation: NVIDIA Pascal vs. Turing (+1 generations) GPU memory (GiB per GPU): 16 (-8) vCPU: 64 (+40) Memory GiB: 440 (same) Temp Storage (SSD) GiB: 2880 (same) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_ND24r	Standard_ND96amsr_A100_v4	CPU: Intel Broadwell vs AMD Rome GPU count: 8 (+4) GPU generation: NVIDIA Pascal vs. Ampere (+2 generation)

Current VM Size	Target VM Size	Difference in Specification
		GPU memory (GiB per GPU): 80 (+56) vCPU: 96 (+72) Memory GiB: 1900 (+1452) Temp Storage (SSD) GiB: 6400 (+3452) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+) InfiniBand interconnect: Yes (Same)

Migration Steps

General Changes

1. Choose a series and size for migration. Leverage the pricing calculator for further insights.
2. Get quota for the target VM series
3. Resize the current N* series VM size to the target size. This may also be a good time to update the operating system used by your Virtual Machine image, or adopt one of the HPC images with drivers pre-installed as your starting point.

> **Important**
>
> Your VM image may have been produced with an older version of the CUDA runtime, NVIDIA driver, and (if applicable, for RDMA-enabled sizes only) Mellanox OFED drivers than your new GPU VM series requires, which can be updated by following the instructions in the Azure Documentation.

Breaking Changes

Select target size for migration

After assessing your current usage, decide what type of GPU VM you need. Depending on the workload requirements you have few different choices.

> **Note**

A best practice is to select a VM size based on both cost and performance. The recommendations in this guide are based on a general-purpose, one-to-one comparison of performance metrics and the nearest match in another VM series. Before deciding on the right size, get a cost comparison using the Azure Pricing Calculator.

Important

All legacy NC, NC v2 and ND-Series sizes are available in multi-GPU sizes, including 4-GPU sizes with and without InfiniBand interconnect for scale-out, tightly-coupled workloads that demand more compute power than a single 4-GPU VM, or a single K80, P40, or P100 GPU can supply respectively. Although the recommendations above offer a straightforward path forward, users of these sizes should consider achieving their performance goals with more powerful NVIDIA V100 GPU-based VM series like the NC v3-Series and ND v2-series, which typically enable the same level of workload performance at lower costs and with improved manageability by providing considerably greater performance per GPU and per VM before multi-GPU and multi-node configurations are required, respectively.

Get quota for the target VM family

Follow the guide to request an increase in vCPU quota by VM family. Select the target VM size you have selected for migration.

Resize the current virtual machine

You can resize the virtual machine.

Next steps

For a full list of GPU enabled virtual machine sizes, see GPU - accelerated compute overview

Feedback

Was this page helpful? Yes No

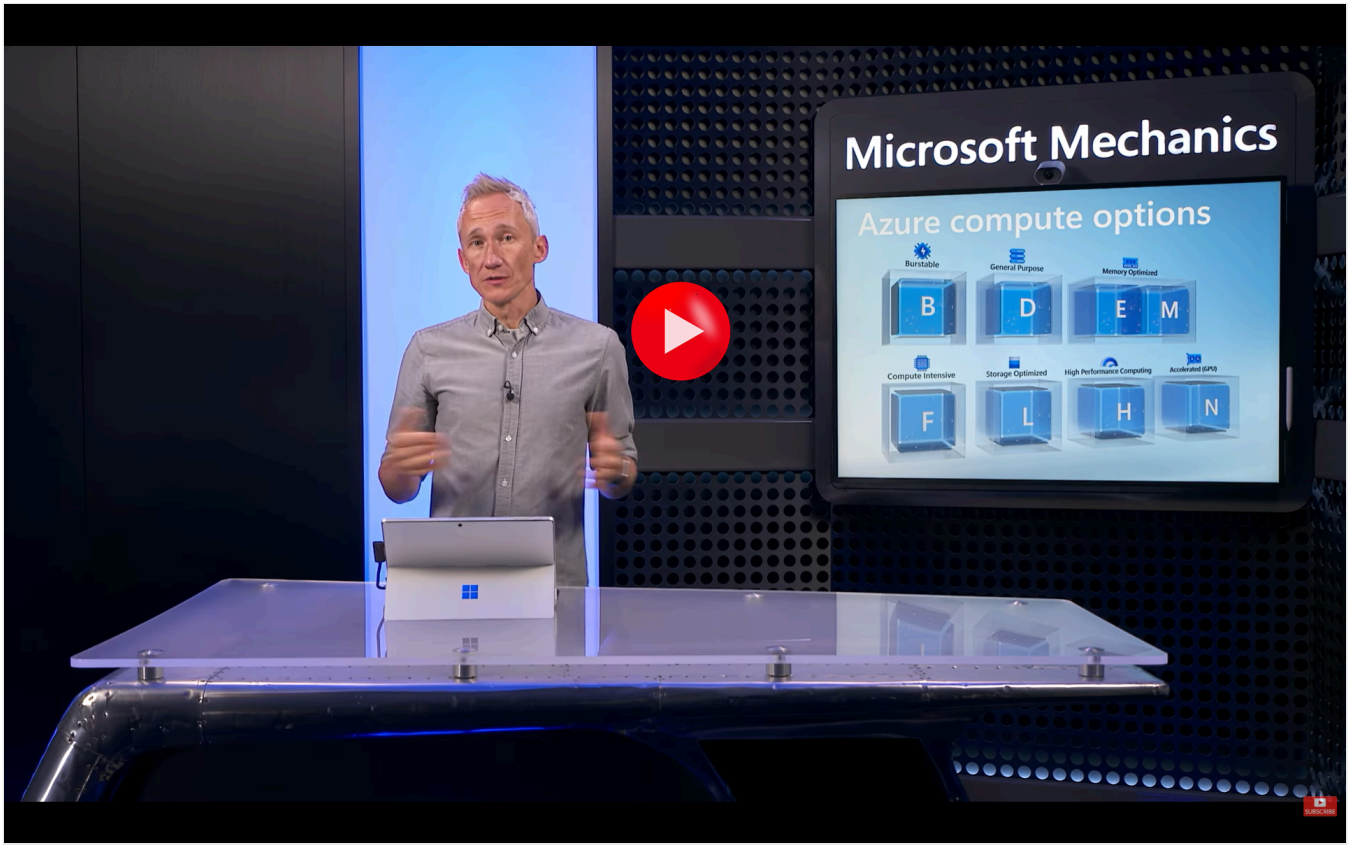
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

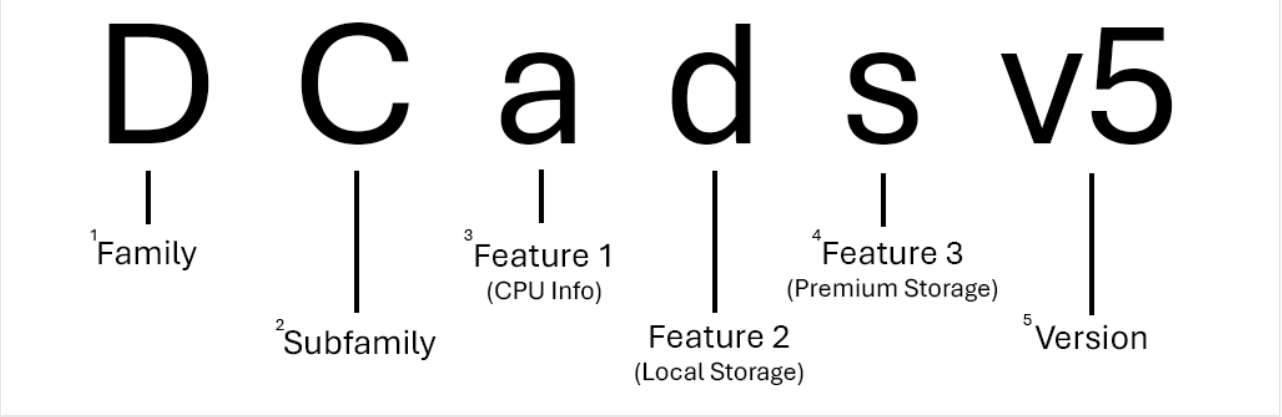
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

NMads MA35D series

Summarize this article for me

The NMads MA35D-Series virtual machines are Azure's first SKU to offer specialized hardware (Xilinx MA35D "Supernova") accelerated VM optimized for batch and real-time video transcoding workloads. The VM series are powered by 4th generation AMD EPYC™ Genoa processors. It offers 1 ASIC video processing unit (VPU) with 8GB of memory in addition to 16 vCPUs, 32GB of RAM, 76GB of temporary storage, and 4Gbps of network bandwidth.

Compared with existing general-purpose CPU or GPU based solutions, the NMads MA35D-Series VM SKU provides higher throughout and lower latency while maintaining a lower TCO for customers. The VM series presents a huge opportunity to save on infrastructure costs and improve performance and efficiency on video transcoding. Additionally, the VM provides access to modern codecs such as AV1 to further improve efficiency for video processing. It is the ideal choice for running your video transcoding workloads on the cloud.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	192 vCPUs	AMD EPYC (Genoa) [x86-64]
Memory	768 GiB	
Local Storage	3 Disk	2TB E1.S NVMe per disk
Remote Storage	8-16 Disks	
Network	5 NICs	
Accelerators	10 MA35D cards/20 VPUs	AMD Xilinx Alveo MA35D

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NM16ads_MA35D	16	32

VM Basics resources

- Check vCPU quotas

Table definitions

Expected network bandwidth is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth

Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance depends on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see Optimize network throughput for Azure virtual machines.

To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
VM Generation 2 Support	Supported

Feature name	Support status
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Temporary local NVMe disks	Supported
Nested Virtualization	Not Supported

Other size information

General purpose

Memory optimized

Storage optimized

GPU optimized

High performance compute

Previous generations

You can use the pricing calculator to estimate your Azure VMs costs.

For more information on disk types, see What disk types are available in Azure?

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Check out Azure Dedicated Hosts for physical servers able to host one or more virtual machines assigned to one Azure subscription.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

'NP' family storage optimized VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

> **Note**
>
> Azure NP-series VMs are scheduled for retirement on May 31, 2027. For more information, see NP-series virtual machine migration guidance

The 'NP' subfamily of VM size series are one of Azure's storage-optimized VM instances. They're designed for workloads that require high disk throughput and I/O, such as databases, big data applications, and data warehousing. High disk throughput and large local disk storage capacities on L-series VMs support applications and services that benefit from low latency and high sequential read and write speeds. This makes them well suited for handling tasks like large-scale log processing, real-time big data analytics, and scenarios involving large databases that perform frequent disk operations, ensuring efficient performance for storage-heavy applications.

Workloads and use cases

Real-Time Data Processing: NP-family VMs excel in environments where data needs to be processed in real time with minimal latency, such as in financial trading, real-time analytics, and network data processing.

Custom AI and Machine Learning: NP-family VMs are suitable for accelerating AI and machine learning inference tasks, where the FPGA can be programmed to execute specific algorithms sometimes faster than typical CPU or GPU-based solutions.

Genomics and Life Sciences: NP-family VMs can significantly speed up genomic sequencing tasks and other life sciences applications that benefit from custom hardware acceleration.

Video Transcoding and Streaming: FPGAs can be used to accelerate video processing tasks such as transcoding and real-time video streaming, optimizing performance and reducing processing times.

Signal Processing: NP-family VMs are ideal for applications in telecommunications and signal processing where rapid manipulation and analysis of signals are necessary.

Database Acceleration: NP-family VMs can enhance database operations, especially for custom search operations and large-scale database queries, by offloading these tasks to the FPGA.

Series in family

NP-series

The NP-series virtual machines are powered by Xilinx U250 FPGAs for accelerating workloads including machine learning inference, video transcoding, and database search & analytics. NP-series VMs are also powered by Intel Xeon 8171M (Skylake) CPUs with all core turbo clock speed of 3.2 GHz.

View the full np-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	10 - 40 vCPUs	Intel Xeon 8171M (Skylake) [x86-64]
Memory	168 - 672 GiB	
Local Storage	1 Disk	736 - 2948 GiB
Remote Storage	8 - 32 Disks	
Network	1 - 4 NICs	7500 - 30000 Mbps
Accelerators	1 - 4 FPGAs	AMD Alveo U250 FPGA (64GB)

Previous-generation NP family series

For older sizes, see previous generation sizes.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/14/2026

NP sizes series

The NP-series virtual machines are powered by Xilinx U250 FPGAs for accelerating workloads including machine learning inference, video transcoding, and database search & analytics. NP-series VMs are also powered by Intel Xeon 8171M (Skylake) CPUs with all core turbo clock speed of 3.2 GHz.

> **Note**
>
> Azure NP-series virtual machines are scheduled for retirement on May 31, 2027. For more information, see NP-series virtual machine migration guidance.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	10 - 40 vCPUs	Intel Xeon 8171M (Skylake) [x86-64]
Memory	168 - 672 GiB	
Local Storage	1 Disk	736 - 2948 GiB
Remote Storage	8 - 32 Disks	
Network	1 - 4 NICs	7500 - 30000 Mbps
Accelerators	1 - 4 FPGAs	AMD Alveo U250 FPGA (64GB)

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_NP10s	10	168
Standard_NP20s	20	336
Standard_NP40s	40	672

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Not Supported
Generation 1 VMs	Supported
Accelerated Networking	Not Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Frequently asked questions

Q: What's the difference between Xilinx U250 and the AMD Alveo U250?

A: AMD Acquired Xilinx and renamed their FPGA line to Alveo. They are identical and use the same drivers, but the original Xilinx page redirects to AMD's new site.

Q: How to request quota for NP VMs?

A: Follow this page Increase VM-family vCPU quotas. NP VMs are available in East US, West US2, SouthCentral US, West Europe, SouthEast Asia, Japan East, and Canada Central.

Q: What version of Vitis should I use?

A: Xilinx recommends Vitis 2022.1, you can also use the Development VM marketplace options (Vitis 2022.1 Development VM for Ubuntu 18.04, Ubuntu 20.04, and CentOS 7.8)

Q: Do I need to use NP VMs to develop my solution?

A: No, you can develop on-premises and deploy to the cloud. Make sure to follow the attestation documentation to deploy on NP VMs.

Q: What shell version is supported and how can I get the development files?

A: The FPGAs in Azure NP VMs support Xilinx Shell 2.1 (gen3x16-xdma-shell_2.1). See Xilinx Page Xilinx/Azure with Alveo U250 to get the development shell files.

Q: Which file returned from attestation should I use when programming my FPGA in an NP VM?

A: Attestation returns two xclbins, **design.bit.xclbin** and **design.azure.xclbin**. Use **design.azure.xclbin**.

Q: Where should I get all the XRT / Platform files?

A: Visit Xilinx's Microsoft-Azure site for all files.

Q: What Version of XRT should I use?

A: xrt_202210.2.13.479

Q: What is the target deployment platform?

A: Use the following platforms.

- xilinx-u250-gen3x16-xdma-platform-2.1-3_all
- xilinx-u250-gen3x16-xdma-validate_2.1-3005608.1

Q: Which platform should I target for development?

A: xilinx-u250-gen3x16-xdma-2.1-202010-1-dev_1-2954688_all

Q: What are the supported Operating Systems?

A: Xilinx and Microsoft validated Ubuntu 18.04 LTS, Ubuntu 20.04 LTS, and CentOS 7.8.

> Xilinx has created the following marketplace images to simplify the deployment of these VMs:
>
> - Xilinx Alveo U250 2022.1 Deployment VM Ubuntu18.04
> - Xilinx Alveo U250 2022.1 Deployment VM Ubuntu20.04
> - Xilinx Alveo U250 2022.1 Deployment VM CentOS7.8

Q: Can I deploy my Own Ubuntu / CentOS VMs and install XRT / Deployment Target Platform?

A: Yes.

Q: If I deploy my own Ubuntu18.04 VM then what are the required packages and steps?

A: Follow the guidance in Xilinx XRT documentation Xilinx XRT documentation

> Install the following packages.
>
> - xrt_202210.2.13.479_18.04-amd64-xrt.deb
> - xrt_202210.2.13.479_18.04-amd64-azure.deb
> - xilinx-u250-gen3x16-xdma-platform-2.1-3_all_18.04.deb.tar.gz
> - xilinx-u250-gen3x16-xdma-validate_2.1-3005608.1_all.deb

Q: If I deploy my own Ubuntu20.04 VM then what are the required packages and steps?

A: Follow the guidance in Xilinx XRT documentation Xilinx XRT documentation

> Install the following packages.
>
> - xrt_202210.2.13.479_20.04-amd64-xrt.deb
> - xrt_202210.2.13.479_20.04-amd64-azure.deb
> - xilinx-u250-gen3x16-xdma-platform-2.1-3_all_18.04.deb.tar.gz
> - xilinx-u250-gen3x16-xdma-validate_2.1-3005608.1_all.deb

Q: If I deploy my own CentOS7.8 VM then what are the required packages and steps?

A: Follow the guidance in Xilinx XRT documentation Xilinx XRT documentation

> Install the following packages.
>
> - xrt_202210.2.13.479_7.8.2003-x86_64-xrt.rpm
> - xrt_202210.2.13.479_7.8.2003-x86_64-azure.rpm
> - xilinx-u250-gen3x16-xdma-platform-2.1-3.noarch.rpm.tar.gz
> - xilinx-u250-gen3x16-xdma-validate-2.1-3005608.1.noarch.rpm

Q: What are the differences between on-premises FPGAs and NP VMs?

A:
- Regarding XOCL/XCLMGMT:
On Azure NP VMs, only the role endpoint (Device ID 5005), which uses the XOCL driver, is present.

In on-premises FPGAs, both the management endpoint (Device ID 5004) and role endpoint (Device ID 5005), which use the XCLMGMT and XOCL drivers respectively, are present.

- Regarding XRT:
On Azure NP VMs, the XDMA 2.1 platform only supports Host_Mem(SB).
To enable Host_Mem(SB) (up to 1-Gb RAM): sudo xbutil host_mem --enable --size 1g
To disable Host_Mem(SB): sudo xbutil host_mem --disable

Starting on XRT2021.1:

On-premises FPGA in Linux exposes M2M data transfer.
This feature isn't supported in Azure NP VMs.

Q: Can I run xbmgmt commands?

A: No, on Azure VMs there's no management support directly from the Azure VM.

Q: Do I need to load a PLP?

A: No, the PLP is loaded automatically for you, so there's no need to load via xbmgmt commands.

Q: Does Azure support different PLPs?

A: Not at this time. We only support the PLP provided in the deployment platform packages.

Q: How can I query the PLP information?

A: Need to run xbutil query and look at the lower portion.

Q: Do Azure NP VMs support FPGA bitstreams with Networking GT Kernel connections?

A: No. The FPGA Attestation service performs a series of validations on a design checkpoint file and generates an error if the user's application contains connections to the FPGA card's QSFP networking ports.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/13/2026

FPGA attestation for Azure NP-Series VMs (Preview)

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The FPGA Attestation service performs a series of validations on a design checkpoint file (called a "netlist") generated by the Xilinx toolset and produces a file that contains the validated image (called a "bitstream") that can be loaded onto the Xilinx U250 FPGA card in an NP series VM.

> **Note**
>
> - The Azure NP-series FPGA Attestation Service will close to new preview sign-ups on May 1, 2026.
> - The attestation service will remain available to previously approved users until June 1, 2026. Bitstreams that have already been attested will continue to function on NP-series virtual machines until the platform retirement date.
> - This change is related to the upcoming retirement of NP-series virtual machines. For more information, see NP-series virtual machine migration guidance.
> - See the Preview Terms Of Use | Microsoft Azure for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.
> - Sign up for the public preview.

News

The current attestation service is using Vitis 2021.1 from Xilinx, on Sept 26th 2022, we'll be moving to Vitis 2022.1. The change should be transparent to most users. Once your designs are "attested" using Vitis 2022.1, you should be moving to XRT2022.1. Xilinx published new marketplace images based on XRT 2022.1. Please note that current designs already attested on Vitis 2020.2 or 2021.1, works on the current deployment marketplace images and new images based on XRT2022.1

As part of the move to 2021.1, Xilinx introduced a new DRC that might affect some designs previously working on Vitis 2020.2 regarding BUFCE_LEAF failing attestation, for more details

here: Xilinx AR 75980 UltraScale/UltraScale+ BRAM: CLOCK_DOMAIN = Common Mode skew checks.

Prerequisites

You need an Azure subscription and an Azure Storage account. The subscription gives you access to Azure and the storage account is used to hold your netlist and output files of the attestation service.

We provide PowerShell and Bash scripts to submit attestation requests. The scripts use Azure CLI, which can run on Windows and Linux. PowerShell can run on Windows, Linux, and macOS.

Azure CLI download (required)

PowerShell for Windows, Linux, and macOS download (only for PowerShell scripts)

You need to have your tenant and subscription ID authorized to submit to the attestation service. Visit https://aka.ms/AzureFPGAAttestationPreview to request access.

Building your design for attestation

The preferred Xilinx toolset for building designs is Vitis 2022.1. Netlist files that were created with an earlier version of the toolset and are still compatible with 2022.1 can be used. Ensure you've loaded the correct shell to build against. The currently supported version is `xilinx_u250_gen3x16_xdma_2_1_202010_1`. The support files can be downloaded from the Xilinx Alveo lounge.

You must include the following argument to Vitis (v++ cmd line) to build an `xclbin` file that contains a netlist instead of a bitstream.

```
--advanced.param compiler.acceleratorBinaryContent=dcp
```

Logging into Azure

Before performing any operations with Azure, you must log into Azure and set the subscription that is authorized to call the service. Use the `az login` and `az account set -s <Sub ID or Name>` commands for this purpose. Further information about this process is documented here: Sign in with Azure CLI. Use either the **sign in interactively** or **sign in with credentials** option on the command line.

Creating a storage account and blob container

Your netlist file must be uploaded to an Azure storage blob container for access by the attestation service.

For more information on creating the account, a container, and uploading your netlist as a blob to that container, see Quickstart: Create, download, and list blobs with Azure CLI.

You can also use the Azure portal for this as well.

Upload your netlist file to Azure blob storage

There are several ways to copy the file; an example using the az storage upload cmdlet is shown below. The az commands run on both Linux and Windows. You can choose any name for the "blob" name but make sure to retain the `xclbin` extension.

```
az storage blob upload --account-name <storage account to receive netlist> --container-name <blob container name> --name <blob filename> --file <local file with netlist>
```

Running the attestation scripts

To run the scripts, you will need to provide the name of your storage account, the name of the blob container where the netlist file is stored and the name of the netlist file. Also, you need to create a Service shared access signature (SAS) that grants read/write access to your container (not the netlist). This SAS is used by the attestation service to make a local copy of your netlist file and to write back the resulting output files of the validation process to your container.

An overview of shared access signatures is available here with specific information about the Service SAS available here. The Service SAS page includes an important caution about protecting the generated SAS. Read the caution to understand the need to keep the SAS protected from malicious or unintended use.

You can generate a SAS for your container using the az storage container generate-sas cmdlet. Specify an expiry time in UTC format that is at least a few hours past the time of submission; around 6 hours should be more than adequate.

If you wish to use virtual directories, you must include the directory hierarchy as part of the container argument. For example, if you have a container named “netlists” and have a virtual directory named “image1” that contains the netlist blob, you would specify “netlists/image1” as the container name. Add any extra directory names to specify a more detailed hierarchy.

PowerShell

PowerShell

```
$sas=$(az storage container generate-sas --account-name <storage acct name> --name
<blob container name> --https-only --permissions rwc --expiry <e.g., 2021-01-
07T17:00Z> --output tsv)

.\Validate-FPGAImage.ps1 -StorageAccountName <storage acct name> -Container <blob
container name> -BlobContainerSAS $sas -NetlistName <netlist blob filename>
```

Bash

Bash

```
sas=az storage container generate-sas --account-name <storage acct name> --name
<blob container name> --https-only --permissions rwc --expiry <2021-01-07T17:00Z> -
-output tsv

validate-fpgaimage.sh --storage-account <storage acct name> --container <blob
container name> --netlist-name <netlist blob filename> --blob-container-sas $sas
```

Checking on the status of your submission

The Attestation service will return the orchestration ID of your submission. The submission scripts automatically start monitoring the submission by polling for completion. The orchestration ID is the primary way for us to review what happened to your submission so please keep that in case you have an issue. As reference points, attestation takes about 30 minutes to complete for a small netlist file (300MB in size); a 1.6GB file took an hour.

You can call the Monitor-Validation.ps1 script at any time to get status and results of attestation, providing the orchestration ID as an argument:

`.\Monitor-Validation.ps1 -OrchestrationId <orchestration ID>`

Alternatively, you can submit HTTP post request to the attestation service endpoint:

`https://fpga-attestation.azurewebsites.net/api/ComputeFPGA_HttpGetStatus`

The request body should contain your Subscription ID, Tenant ID, and orchestration ID of your attestation request:

JSON

```
{
  "OrchestrationId": "<orchestration ID>",
  "ClientSubscriptionId": "<your subscription ID>",
```

```
  "ClientTenantId": "<your tenant ID>"
}
```

Post validation steps

The service will write its output back to your container. If the validation pass succeeds, your container will have the original netlist file (abc.xclbin), a file with the bitstream (abc.bit.xclbin), a file that identifies the private location of your stored bitstream (abc.azure.xclbin) and four log files: one for the startup process (abc-log.txt) and one each for the three parallel phases that perform the validation. These are named *logPhaseX.txt where X is a number for the phase. The azure.xclbin is used on your VM to signal the uploading of your validated image to the U250.

If validation failed, an error-*.txt file is written indicating which step failed. Also check the log files if the error log indicates that attestation failed. When contacting us for support, please be sure to include all these files as part of the support request along with the orchestration ID.

You can use the Azure portal to create your container as well as uploading your netlist and downloading the bitstream and log files. Submitting an attestation request and monitoring its progress through the portal is not supported at this time and must be done through scripts as described above.

Last updated on 04/14/2026

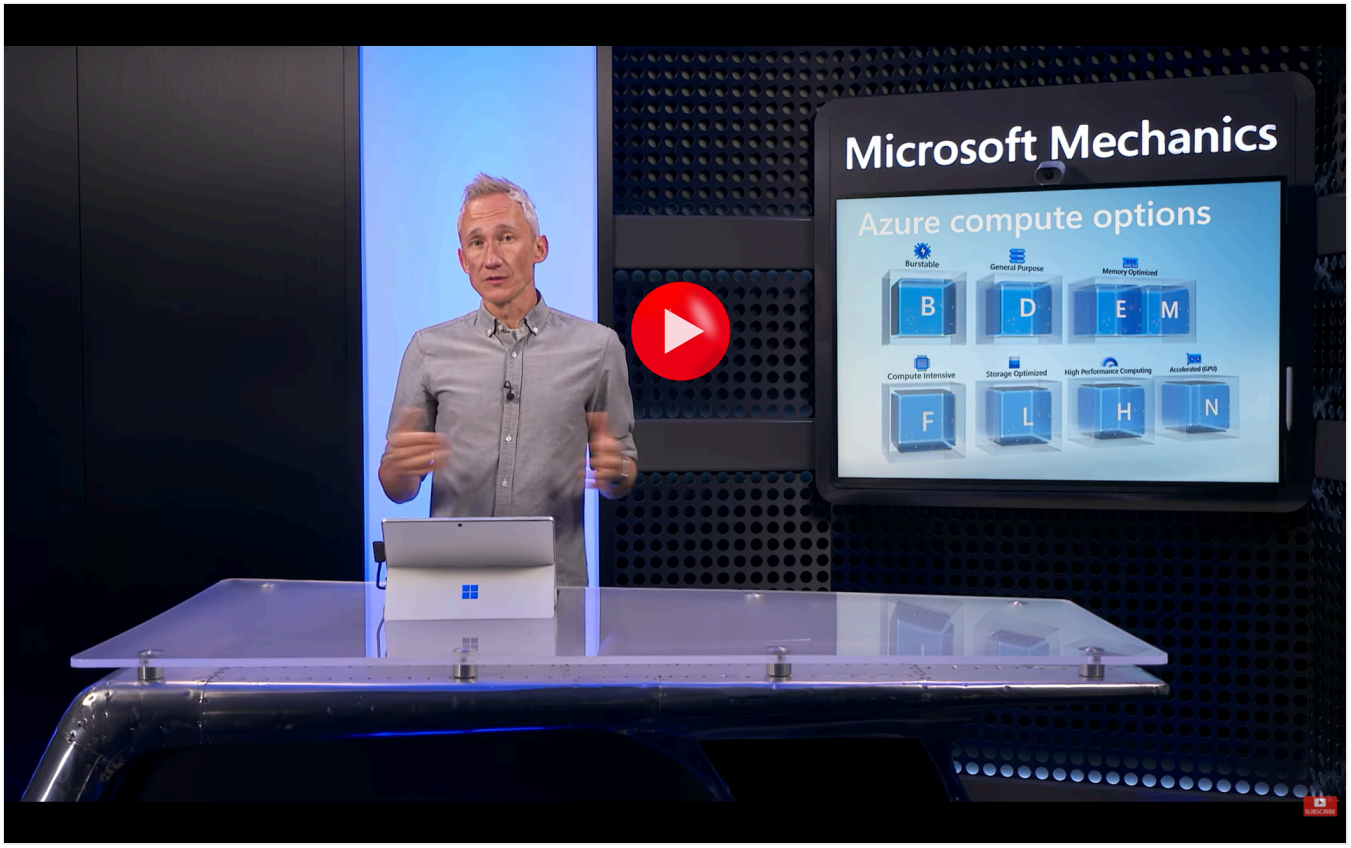
Sizes for virtual machines in Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machine (VM) sizes are designed to provide a wide range of options for hosting your servers and their workloads in the cloud. Sizes are categorized into different families and types, each optimized for specific purposes. Users can choose the most suitable VM size based on their requirements, such as CPU, memory, storage, and network bandwidth.

This article describes what sizes are, gives an overview of the available sizes and shows different options for Azure virtual machine instances you can use to run your apps and workloads.



VM size and series naming

Azure VM sizes follow specific naming conventions to denote varying features and specifications. Each character in the name represents different aspects of the VM. These include the VM family, number of vCPUs, and extra features like premium storage or included accelerators.

VM naming is further broken down into the 'Series' name and the 'Size' name. Size names include extra characters representing the number of vCPUs, type of storage, etc.

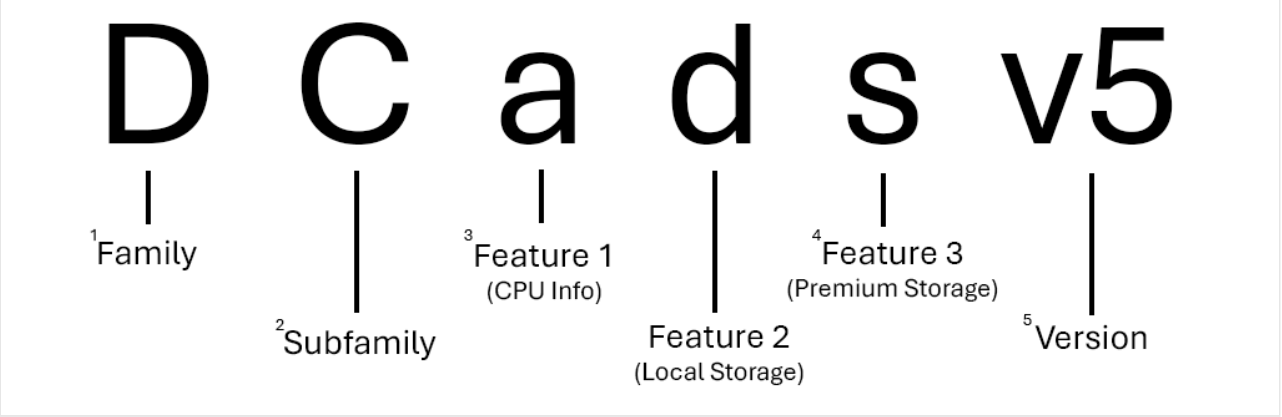
Expand table

Category	Description	Links
Type	Basic categorization by intended workload.	General purpose Compute optimized Memory optimized Storage optimized GPU accelerated FPGA accelerated
Series	Group of sizes with similar hardware and features.	Enter the 'Series' tab here.
Size	Specific VM configuration, including vCPUs, memory, and accelerators.	Enter the 'Size' tab here.

Name structure breakdown

Series

Here's a breakdown of a 'General purpose, **DCads_v5**-series' size series.



[1] Most families are represented using one letter, but others such as GPU sizes (`ND-series`, `NV-series`, etc.) use two.

[2] Most subfamilies are represented with a single upper case letter, but others (such as `Ebsv5-series`) are still considered subfamilies of their parent family due to feature differences.

[3] If no feature letter for a CPU is listed, the series uses Intel x86-64 CPUs. If the CPU is AMD, it's listed as `a`. If the CPU is ARM based (Microsoft Cobalt or Ampere Altra), it's listed as `p`.

[4] There can be any number of extra features in a size name. There could be none (`Dv5-series`) or there could be three (`Dplds_v6-series`).

[5] Version numbers only appear in the size name if there are multiple versions of the same

series. If you're using the first version of a series (`HB-series`, `B-series`, etc.) it's often not included in the size name.

> **Note**
>
> Not all sizes will have subfamilies, support accelerators, or specify the CPU vendor. For more information on VM size naming conventions, see Azure VM sizes naming conventions.

List of VM size families by type

This section contains a list of all current generation size series with tabs dedicated to each size family. Each group has a 'Series List' column with a linked list of all available size series, These links will bring you to the family page for that series, where you can find detailed information on each size in that series or go to the series' page for a list of sizes in that series.

To learn more about a size family, select the 'family' tab under each type section. There you can read a summary on the family, see the workloads it's recommended for, and view the full family page with specifications for all series in that family.

General purpose

General purpose VM sizes provide balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

Family list

Expand table

Family	Workloads	Series List
A-family	Entry-level economical	Av2-series Previous-gen A-family series
B-family	Burstable	Bsv2-series Basv2-series Bpsv2-series Previous-gen B-family series
D-family	Enterprise-grade applications Relational databases	Dsv7 and Ddsv7 series Dlsv7 and Dldsv7 series

Family	Workloads	Series List
	In-memory caching Data analytics	Dasv7 and Dadsv7 series Dalsv7 and Daldsv7 series Dpsv6 and Dplsv6-series Dpdsv6 and Dpldsv6-series Dasv6 and Dadsv6-series Dalsv6 and Daldsv6-series Dsv6 and Ddsv6-series Dlsv6 and Dldsv6-series Dpsv5 and Dpdsv5-series Dplsv5 and Dpldsv5-series Dlsv5 and Dldsv5-series Dv5 and Dsv5-series Ddv5 and Ddsv5-series Dasv5 and Dadsv5-series Previous-gen D-family series
DC-family	D-family with confidential computing	DCasv5 and DCadsv5-series DCas_cc_v5 and DCads_cc_v5-series DCesv6-series DCedsv6-series DCasv6 and DCadsv6-series DCsv3 and DCdsv3-series Previous-gen DC-family

Compute optimized

Compute optimized VM sizes have a high CPU-to-memory ratio. These sizes are good for medium traffic web servers, network appliances, batch processes, and application servers.

Family list

List of compute optimized VM size families:

Expand table

Family	Workloads	Series List
F-family	Medium traffic web servers Network appliances Batch processes Application servers	Fasv7 and Fadsv7 series Famsv7 and Famdsv7 series Falsv7 and Faldsv7 series Fasv6, Falsv6, and Famsv6-series Fsv2-series Previous-gen F-family

Family	Workloads	Series List
FX-family	Electronic Design Automation (EDA) Large memory relational databases Medium to large caches In-memory analytics	FX-series FXmsv2-series FXmdsv2-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Memory optimized

Memory optimized VM sizes offer a high memory-to-CPU ratio that is great for relational database servers, medium to large caches, and in-memory analytics.

Family list

List of memory optimized VM sizes with links to each series' family page section:

Expand table

Family	Workloads	Series List
E-family	Relational databases Medium to large caches In-memory analytics	Esv7 and Edsv7-series Easv7 and Eadsv7 series Esv6 and Edsv6-series Epsv6 and Epdsv6-series Easv6 and Eadsv6-series Ev5 and Esv5-series Edv5 and Edsv5-series Easv5 and Eadsv5-series Epsv5 and Epdsv5-series Previous-gen families
Eb-family	E-family with High remote storage performance	Ebsv6 and Ebdsv6-series Ebdsv5 and Ebsv5-series
EC-family	E-family with confidential computing	ECasv6 and ECadsv6-series ECasv5 and ECadsv5-series ECas_cc_v5 and ECads_cc_v5-series
M-family	Extremely large databases Large amounts of memory	Mbsv3 and Mbdsv3-series Msv3 and Mdsv3 Medium Memory series Msv3 and Mdsv3 High Memory series Mdsv3 Very High Memory series

Family	Workloads	Series List
		Msv2 and Mdsv2-series Msv2 High Memory Series M-series
Other families	Older generation memory optimized sizes	Dnsv6 and Dndsv6-series Dnlsv6 and Dnldsv6-series Ensv6 and Endsv6-series Previous-gen families

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

Storage optimized

Storage optimized virtual machine (VM) sizes offer high disk throughput and IO, and are ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases. Examples include Cassandra, MongoDB, Cloudera, and Redis.

Family list

List of storage optimized VM size families:

Expand table

Family	Workloads	Series List
L-family	High disk throughput and IO Big Data SQL and NoSQL databases Data warehousing Large transactional databases	Lsv4-series Lasv4-series Laosv4-series Lsv3-series Lasv3-series Previous-gen L-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

GPU accelerated

GPU optimized VM sizes are specialized virtual machines available with single, multiple, or fractional GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.

Family list

List of GPU optimized VM size families:

Expand table

Family	Workloads	Series List
NC-family	Compute-intensive Graphics-intensive Visualization	NC_RTXPRO6000BSE_v6-series NCads_H100_v5-series NCCads_H100_v5-series NCv3-series NCasT4_v3-series NC_A100_v4-series
ND-family	Large memory compute-intensive Large memory graphics-intensive Large memory visualization	ND GB300-v6-series ND GB200-v6-series ND_MI300X_v5-series ND-H200-v5-series ND-H100-v5-series NDm_A100_v4-series ND_A100_v4-series
NG-family	Virtual Desktop (VDI) Cloud gaming	NGads V620-series
NV-family	Virtual desktop (VDI) Single-precision compute Video encoding and rendering	NV-series NVv3-series NVv4-series NVadsA10_v5-series NVads V710 v5-series Previous-gen NV-family

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

FPGA accelerated

FPGA optimized VM sizes are specialized virtual machines available with single or multiple FPGAs. These sizes are designed for compute-intensive workloads. This article provides information about the number and type of FPGAs, vCPUs, data disks, and NICs. Storage throughput and network bandwidth are also included for each size in this grouping.

Family list

List of field programmable gate array accelerated VM size families:

Expand table

Family	Workloads	Series List
NP-family	Machine learning inference Video transcoding Database search and analytics	NP-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

High performance compute

Azure High Performance Compute VMs are optimized for various HPC workloads such as computational fluid dynamics, finite element analysis, frontend and backend EDA, rendering, molecular dynamics, computational geoscience, weather simulation, and financial risk analysis.

Family list

List of high performance computing optimized VM size families:

Expand table

Family	Workloads	Series List
HB-family	High memory bandwidth Fluid Dynamics Weather modeling	HBv2-series HBv3-series HBv4-series HBv5-series
HC-family	High density compute Finite element analysis Molecular dynamics Computational chemistry	HC-series
HX-family	Large memory capacity Electronic Design Automation (EDA)	HX-series

To learn more about a specific size family or series, select the tab for that family and scroll to find your desired size series.

List available machine sizes by region.

The Azure CLI can be leveraged to identify which machine sizes are available for a particular region using the command:

```
az vm list-skus --location <region> --output table
```

Learn platform sizes content

- For information about pricing of the various sizes, see the pricing pages for Linux or Windows.
- Want to change the size of your VM? See Change the size of a VM.
- For availability of VM sizes in Azure regions, see Products available by region.
- To see general limits on Azure VMs, see Azure subscription and service limits, quotas, and constraints.
- For more information on how Azure names its VMs, see Azure virtual machine sizes naming conventions.

REST API

For information on using the REST API to query for VM sizes, see the following:

- List available virtual machine sizes for resizing
- List available virtual machine sizes for a subscription
- List available virtual machine sizes in an availability set

Benchmark scores

Learn more about compute performance for Linux VMs using the CoreMark benchmark scores.

Learn more about compute performance for Windows VMs using the SPECInt benchmark scores.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/09/2026

'HB' sub-family storage optimized VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'HB' sub-family of VM size series are one of Azure's high-performance computing (HPC) optimized H-family VM instances. They're designed for compute-intensive workloads, such as computational fluid dynamics, finite element analysis, and large-scale scientific simulations. HB-series VMs, equipped with high-performance AMD EPYC processors and fast memory, deliver exceptional CPU and memory bandwidth. They're ideal for applications that need extensive computational resources to handle large-scale calculations and data processing. They're well-suited for industries such as engineering, scientific research, and data analysis. In these fields, processing speed and accuracy are essential for maintaining productivity and driving innovation.

Workloads and use cases

Computational Fluid Dynamics (CFD): HB family VMs are ideal for simulations in fields like aerospace, automotive design, and manufacturing, where fluid dynamics calculations are intensive.

Finite Element Analysis (FEA): HB family VMs are suitable for engineering analyses that simulate physical phenomena, requiring intensive computational power to model complex systems and materials.

Weather Forecasting: HB family VMs can handle the massive datasets and complex simulations required for high-resolution weather modeling and forecasting.

Seismic Processing: HB family VMs are in the oil and gas industry, HB family VMs can process seismic data to help map and understand subsurface structures.

Scientific Research: HB family VMs support a wide range of scientific research that requires large-scale mathematical modeling, including physics and computational chemistry simulations.

Genomics and Bioinformatics: HB family VMs are also used in life sciences for genomic analysis, where large amounts of data need to be processed quickly to decode genetic information.

Series in family

HBv2-series

HBv2-series VMs are optimized for applications that are driven by memory bandwidth, such as fluid dynamics, finite element analysis, and reservoir simulation. HBv2 VMs feature 120 AMD EPYC 7V12 processor cores, 4 GB of RAM per CPU core, and no simultaneous multithreading. Each HBv2 VM provides up to 350 GB/s of memory bandwidth, and up to 4 teraFLOPS of FP64 compute. HBv2-series VMs feature 200 Gb/sec Mellanox HDR InfiniBand. These VMs are connected in a non-blocking fat tree for optimized and consistent RDMA performance. These VMs support Adaptive Routing and the Dynamic Connected Transport (DCT, in addition to standard RC and UD transports). These features enhance application performance, scalability, and consistency, and their usage is recommended.

View the full hbv2-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 120 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
L3 Cache	1536 MB	
Memory	456 GB	350 GB/s
Local Storage	1 Temp Disk 1 NVMe Disk	480 GiB 960 GiB
Remote Storage	8 Disks	
Network	8 vNICs 1 InfiniBand HDR NIC	40 Gb/s 200 Gb/s
Accelerators	None	

HBv3-series

HBv3-series Virtual Machines (VMs) are designed for very demanding computing tasks. They help with things like studying how liquids move, analyzing structures, predicting weather, and processing earthquake data. They also support oil reservoir modeling and testing computer chip designs. HBv3 VMs feature up to 120 AMD EPYC™ 7V73X (Milan-X) CPU cores, 448 GB of RAM, and no simultaneous multithreading. HBv3-series VMs also provide 350 GB/sec of memory bandwidth (amplified up to 630 GB/s), up to 96 MB of L3 cache per core (1536 MB total per VM), up to 7 GB/s of block device SSD performance, and clock frequencies up to 3.5 GHz. All HBv3-series VMs feature 200 Gb/sec High Data Rate (HDR) InfiniBand from NVIDIA

Networking to enable supercomputer-scale Message Pass Interface (MPI) workloads. These VMs are connected in a nonblocking fat tree for optimized and consistent RDMA performance. The HDR InfiniBand fabric also supports Adaptive Routing and Dynamic Connected Transport (DCT), along with standard Reliable Connection (RC) and Unreliable Datagram (UD) transports. These features improve performance, make scaling easier, and keep results consistent.

View the full hbv3-series page.

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 120 vCPUs	AMD EPYC 7V73X (Milan-X) [x86-64]
L3 Cache	1536 MB	
Memory	448 GB	350 GB/s
Local Storage	1 Temp Disk 2 NVMe Disks	480 GiB 960 GiB
Remote Storage	32 Disks	
Network	8 vNICs 1 InfiniBand HDR NIC	40 Gb/s 200 Gb/s
Accelerators	None	

HBv4-series

HBv4-series VMs are optimized for many high-performance computing (HPC) workloads. These include:

- Computational fluid dynamics
- Finite element analysis
- Frontend and backend Electronic Design Automation (EDA)
- Rendering and graphics work
- Molecular dynamics
- Computational geoscience
- Weather simulation
- Financial risk analysis

HBv4 VMs feature up to 176 AMD EPYC™ 9V33X ("Genoa-X") CPU cores with AMD's 3D V-Cache, clock frequencies up to 3.7 GHz, and no simultaneous multithreading. HBv4-series VMs also provide 768 GB of RAM, 2304 MB L3 cache. Each VM has 2304 MB of L3 cache that

delivers up to 5.7 TB/s of bandwidth. This boosts DRAM bandwidth of 780 GB/s, giving an average of 1.2 TB/s of effective memory speed for many workloads. The VMs also provide up to 12 GB/s (reads) and 7 GB/s (writes) of block device SSD performance.

All HBv4-series VMs feature 400 Gb/s NDR InfiniBand from NVIDIA Networking to enable supercomputer-scale MPI workloads. These VMs are connected in a non-blocking fat tree for optimized and consistent RDMA performance. NDR continues to support features like Adaptive Routing and the Dynamically Connected Transport (DCT). This newest generation of InfiniBand also brings greater support for offload of MPI collectives, optimized real-world latencies due to congestion control intelligence, and enhanced adaptive routing capabilities. These features enhance application performance, scalability, and consistency, and their usage is recommended.

View the full hbv4-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	24 - 176 vCPUs	AMD EPYC 9V33X (Genoa-X) [x86-64]
L3 Cache	2304 MB	
Memory	768 GB	780 GB/s
Local Storage	1 Temp Disk 2 NVMe Disks	480 GiB 1800 GiB
Remote Storage	32 Disks	
Network	8 vNICs 1 InfiniBand NDR NIC	80 Gb/s 400 Gb/s
Accelerators	None	

HBv5-series

HBv5-series VMs are optimized for the most memory bandwidth-intensive HPC applications, including:

- Computational fluid dynamics
- Automotive and aerospace simulations
- Weather modeling
- Energy research
- Molecular dynamics simulations

- Computer-aided engineering and other HPC workloads

HBv5 VMs feature 6.7 TB/s of memory bandwidth across 432 GB of high-bandwidth memory (HBM) and up to 368 4th Generation AMD EPYC™ processor cores with 4 GHz boost frequencies, 3.5 GHz base frequencies, and no simultaneous multithreading.

Each HBv5-series VM also includes 14.3 TiB of local NVMe SSD storage with up to 50 GB/s (reads) and 30 GB/s (writes) of block device performance.

All HBv5-series VMs feature 800 Gb/s per node (4 x 200 Gb/s CX-7 NIC) of InfiniBand connectivity from NVIDIA Networking to enable supercomputer-scale MPI workloads. These VMs are connected in a nonblocking fat tree for optimized and consistent RDMA performance.

The InfiniBand NICs support features like Adaptive Routing, Dynamically Connected Transport (DCT), hardware acceleration of MPI collectives, and congestion control. These features enhance application performance, scalability, and consistency, and their usage is recommended for optimal performance and cost efficiency.

View the full hbv5-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	48 - 368 vCPUs	4th Generation AMD EPYC [x86-64]
L3 Cache	1.5 GB	
Memory	450 GB	6.7 TB/s
Local Storage	1 Temp Disk 8 NVMe Disks	480 GiB 14.304 TiB (1.9 TB/Disk)
Remote Storage	32 Disks	
Network	8 vNICs 4 InfiniBand NICs	180 Gb/s 800 Gb/s (4 x 200 Gb/s)
Accelerators	None	

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 11/25/2025

HBv2 series virtual machine overview

> **ⓘ Important**
>
> **HBv2-series VMs are scheduled for retirement on May 31, 2027.** This applies to all HBv2 sizes: Standard_HB120rs_v2, Standard_HB120-96rs_v2, Standard_HB120-64rs_v2, Standard_HB120-32rs_v2, and Standard_HB120-16rs_v2. After this date, HBv2 VMs will be set to a deallocated state, stop working, stop incurring billing charges, and lose SLA and support. Plan your migration to current-generation HPC alternatives before the retirement date.

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets.

Maximizing high performance compute (HPC) application performance on AMD EPYC requires a thoughtful approach memory locality and process placement. Below we outline the AMD EPYC architecture and our implementation of it on Azure for HPC applications. We use the term **pNUMA** to refer to a physical NUMA domain, and **vNUMA** to refer to a virtualized NUMA domain.

Physically, an HBv2-series server is 2 * 64-core EPYC 7V12 CPUs for a total of 128 physical cores. Simultaneous Multithreading (SMT) is disabled on HBv2. These 128 cores are divided into 16 sections (8 per socket), each section containing 8 processor cores. Azure HBv2 servers also run the following AMD BIOS settings:

Output

```
Nodes per Socket (NPS) = 2
L3 as NUMA = Disabled
NUMA domains within VM OS = 4
C-states = Enabled
```

As a result, the server boots with 4 NUMA domains (2 per socket). Each domain is 32 cores in size. Each NUMA has direct access to 4 channels of physical DRAM operating at 3,200 MT/s.

To provide room for the Azure hypervisor to operate without interfering with the VM, we reserve 8 physical cores per server.

VM topology

We reserve these 8 hypervisor host cores symmetrically across both CPU sockets, taking the first 2 cores from specific Core Complex Dies (CCDs) on each NUMA domain, with the remaining cores for the HBv2-series VM. The CCD boundary isn't equivalent to a NUMA boundary. On HBv2, a group of four consecutive (4) CCDs is configured as a NUMA domain, both at the host server level and within a guest VM. Thus, all HBv2 VM sizes expose 4 NUMA domains that appear to an OS and application. 4 uniform NUMA domains, each with different number of cores depending on the specific HBv2 VM size.

Process pinning works on HBv2-series VMs because we expose the underlying silicon as-is to the guest VM. We strongly recommend process pinning for optimal performance and consistency.

Hardware specifications

Expand table

Hardware Specifications	HBv2-series VM
Cores	120 (SMT disabled)
CPU	AMD EPYC 7V12
CPU Frequency (non-AVX)	~3.1 GHz (single + all cores)
Memory	4 GB/core (480 GB total)
Local Disk	960 GiB NVMe (block), 480 GB SSD (page file)
Infiniband	200 Gb/s HDR Mellanox ConnectX-6
Network	50 Gb/s Ethernet (40 Gb/s usable) Azure second Gen SmartNIC

Software specifications

Expand table

Software Specifications	HBv2-series VM
Max MPI Job Size	36000 cores (300 VMs in a single virtual machine scale set with singlePlacementGroup=true)
MPI Support	HPC-X, Intel MPI, OpenMPI, MVAPICH2, MPICH, Platform MPI
Additional Frameworks	UCX, libfabric, PGAS

Software Specifications	HBv2-series VM
Azure Storage Support	Standard and Premium Disks (maximum 8 disks)
OS Support for SRIOV RDMA	RHEL 7.9+, Ubuntu 18.04+, SLES 12 SP5+, WinServer 2016+
Orchestrator Support	CycleCloud, Batch, AKS; cluster configuration options

> **Note**
>
> Windows Server 2012 R2 isn't supported on HBv2 and other VMs with more than 64 (virtual or physical) cores. For more information, see Supported Windows guest operating systems for Hyper-V on Windows Server.

Availability and purchasing

> **Important**
>
> **Reserved Instance purchase end date**: 1-year and 3-year Reserved Instance purchases for HBv2-series VMs ended on April 2, 2026. Avoid new long-term RI commitments on HBv2-series. Existing RIs remain valid until their expiration date.

- **Pay-as-you-go**: HBv2-series VMs remain available on a pay-as-you-go basis until retirement on **May 31, 2027**.
- **Plan migration**: Begin planning and testing migrations to current-generation HPC alternatives ahead of the retirement date. If you have active RIs for HBv2, consider exchanging them for supported HPC series RIs or trading them in for an Azure Savings Plan for compute. For more information, see Exchange and refund Azure reservations.

Migrate to current-generation HPC VMs

HBv2-series VMs will be retired on May 31, 2027. Microsoft recommends migrating to the following current-generation HPC VM series:

Expand table

Recommended series	Key characteristics
HBv5-series	4th Gen AMD EPYC (Zen4), up to 368 cores, HBM3 memory, NDR InfiniBand – best for memory bandwidth-intensive workloads
HX-series	AMD EPYC 9004, up to 176 cores, large L3 cache – suited for memory-capacity-intensive HPC workloads
HBv4-series	4th Gen AMD EPYC (Genoa), up to 176 cores, NDR InfiniBand – strong all-round HPC replacement
HBv3-series	3rd Gen AMD EPYC (Milan), up to 120 cores, HDR InfiniBand – comparable generation step-up from HBv2

Before migrating, validate your workloads on the target series:

- **MPI and RDMA**: Confirm compatibility with the target InfiniBand fabric (HDR vs. NDR) and your MPI library version.
- **Memory bandwidth**: Run representative benchmarks to confirm performance parity or improvement on the target series.
- **Application testing**: Complete end-to-end testing of your HPC applications on the target series before production migration.

Next steps

- For migration to current-generation HPC VMs, see the Migrate to current-generation HPC VMs section in this article.
- For more information about AMD EPYC architecture and multi-chip architectures, see the HPC Tuning Guide for AMD EPYC Processors.
- For latest announcements on HPC workload examples, and performance results see Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.
- For more information on the current-generation HBv5 series, see HBv5-series overview.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/14/2026

HBv2 sizes series

> **Important**
>
> **HBv2-series VMs are scheduled for retirement on May 31, 2027.** This applies to all HBv2 sizes: Standard_HB120rs_v2, Standard_HB120-96rs_v2, Standard_HB120-64rs_v2, Standard_HB120-32rs_v2, and Standard_HB120-16rs_v2. After this date, HBv2 VMs will be set to a deallocated state, stop working, stop incurring billing charges, and lose SLA and support. Plan your migration to current-generation HPC alternatives before the retirement date. For migration guidance, see the Migration guidance section below.

HBv2-series VMs are optimized for applications that are driven by memory bandwidth, such as fluid dynamics, finite element analysis, and reservoir simulation. HBv2 VMs feature 120 AMD EPYC 7V12 processor cores, 4 GB of RAM per CPU core, and no simultaneous multithreading. Each HBv2 VM provides up to 350 GB/s of memory bandwidth, and up to 4 teraFLOPS of FP64 compute. HBv2-series VMs feature 200 Gb/sec Mellanox HDR InfiniBand. These VMs are connected in a non-blocking fat tree for optimized and consistent RDMA performance. These VMs support Adaptive Routing and the Dynamic Connected Transport (DCT, in addition to standard RC and UD transports). These features enhance application performance, scalability, and consistency, and their usage is recommended.

> **Note**
>
> HBv2-series is scheduled for retirement on May 31, 2027. For new HPC architecture planning, consider current-generation alternatives such as HBv5-series, HX-series, HBv4-series, or HBv3-series VMs.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 120 vCPUs	AMD EPYC 7V12 (Rome) [x86-64]
L3 Cache	1536 MB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Memory	456 GB	350 GB/s
Local Storage	1 Temp Disk 1 NVMe Disk	480 GiB 960 GiB
Remote Storage	8 Disks	
Network	8 vNICs 1 InfiniBand HDR NIC	40 Gb/s 200 Gb/s
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (MB)	Memory Bandwidth (GB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HB120rs_v2	120	456	512	350	2.45	3.3	3.1
Standard_HB120-96rs_v2	96	456	512	350	2.45	3.3	3.1
Standard_HB120-64rs_v2	64	456	512	350	2.45	3.3	3.1
Standard_HB120-32rs_v2	32	456	512	350	2.45	3.3	3.1
Standard_HB120-16rs_v2	16	456	512	350	2.45	3.3	3.1

VM Basics resources

- Check vCPU quotas

Lifecycle and procurement

> **Important**
>
> **Reserved Instance purchase end date**: 1-year and 3-year Reserved Instance purchases for HBv2-series VMs ended on April 2, 2026. New long-term RI commitments are no longer available for this series. Existing RIs remain valid until their expiration.

- **Pay-as-you-go**: HBv2-series VMs remain available on a pay-as-you-go basis until the retirement date of **May 31, 2027**.
- **Plan migration**: Begin migrating workloads to current-generation HPC alternatives before May 31, 2027. After this date, HBv2 VMs will be deallocated automatically.

For migration recommendations, see the Migration guidance section.

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Backend Network	InfiniBand HDR

Migration guidance

HBv2-series VMs will be retired on May 31, 2027. Microsoft recommends migrating to the following current-generation HPC VM series before the retirement date:

Expand table

Recommended series	Key characteristics
HBv5-series	4th Gen AMD EPYC (Zen4), up to 368 cores, HBM3 memory, NDR InfiniBand – best for memory bandwidth-intensive workloads
HX-series	AMD EPYC 9004, up to 176 cores, large L3 cache – suited for memory-capacity-intensive HPC workloads
HBv4-series	4th Gen AMD EPYC (Genoa), up to 176 cores, NDR InfiniBand – strong all-round HPC replacement
HBv3-series	3rd Gen AMD EPYC (Milan), up to 120 cores, HDR InfiniBand – comparable generation step-up from HBv2

When selecting a replacement size, consider the following:

- **MPI and RDMA requirements**: Verify that your MPI library and InfiniBand fabric (HDR vs. NDR) are supported on the target series.
- **Memory bandwidth**: Benchmark your workload on the target series to confirm performance parity or improvement.
- **Workload compatibility**: Test application correctness and performance on the target series before production migration.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.

For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/14/2026

HBv2-series virtual machine sizes

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Performance testing was conducted across multiple HBv2-series size VMs. The following table summarizes key findings from these tests.

Expand table

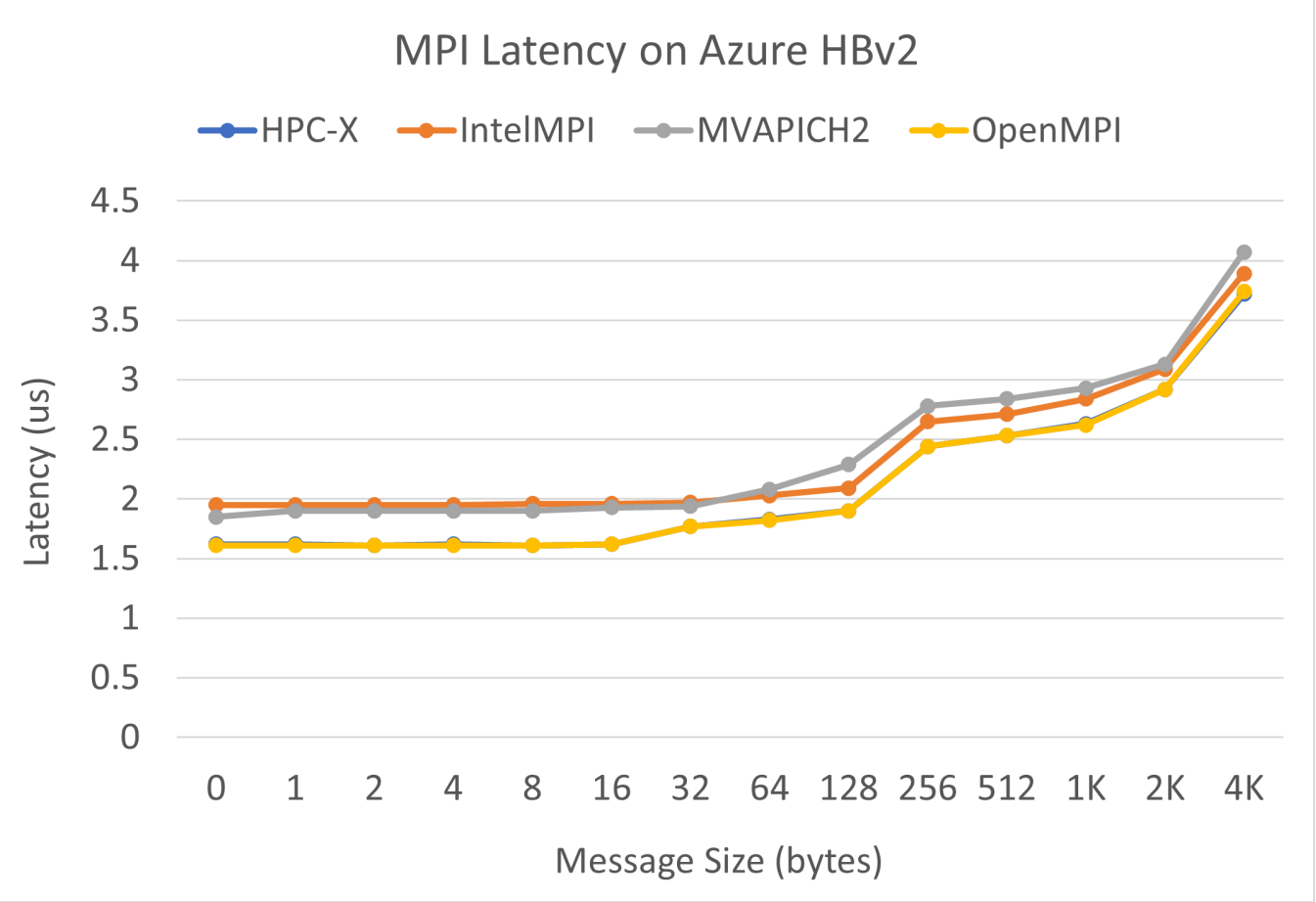
Workload	HBv2
STREAM Triad	350 GB/s (21-23 GB/s per CCX)
High-Performance Linpack (HPL)	4 TeraFLOPS (Rpeak, FP64), 8 TeraFLOPS (Rmax, FP32)
RDMA latency & bandwidth	1.2 microseconds, 190 Gb/s
FIO on local NVMe SSD	2.7 GB/s reads, 1.1 GB/s writes; 102k IOPS reads, 115 IOPS writes
IOR on 8 * Azure Premium SSD (P40 Managed Disks, RAID0)**	1.3 GB/s reads, 2.5 GB/writes; 101k IOPS reads, 105k IOPS writes

MPI latency

MPI latency test from the OSU Micro-Benchmark (OMB) suite is run. Sample scripts are on GitHub.

Bash

```
./bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=[INSERT CORE #]
./osu_latency
```

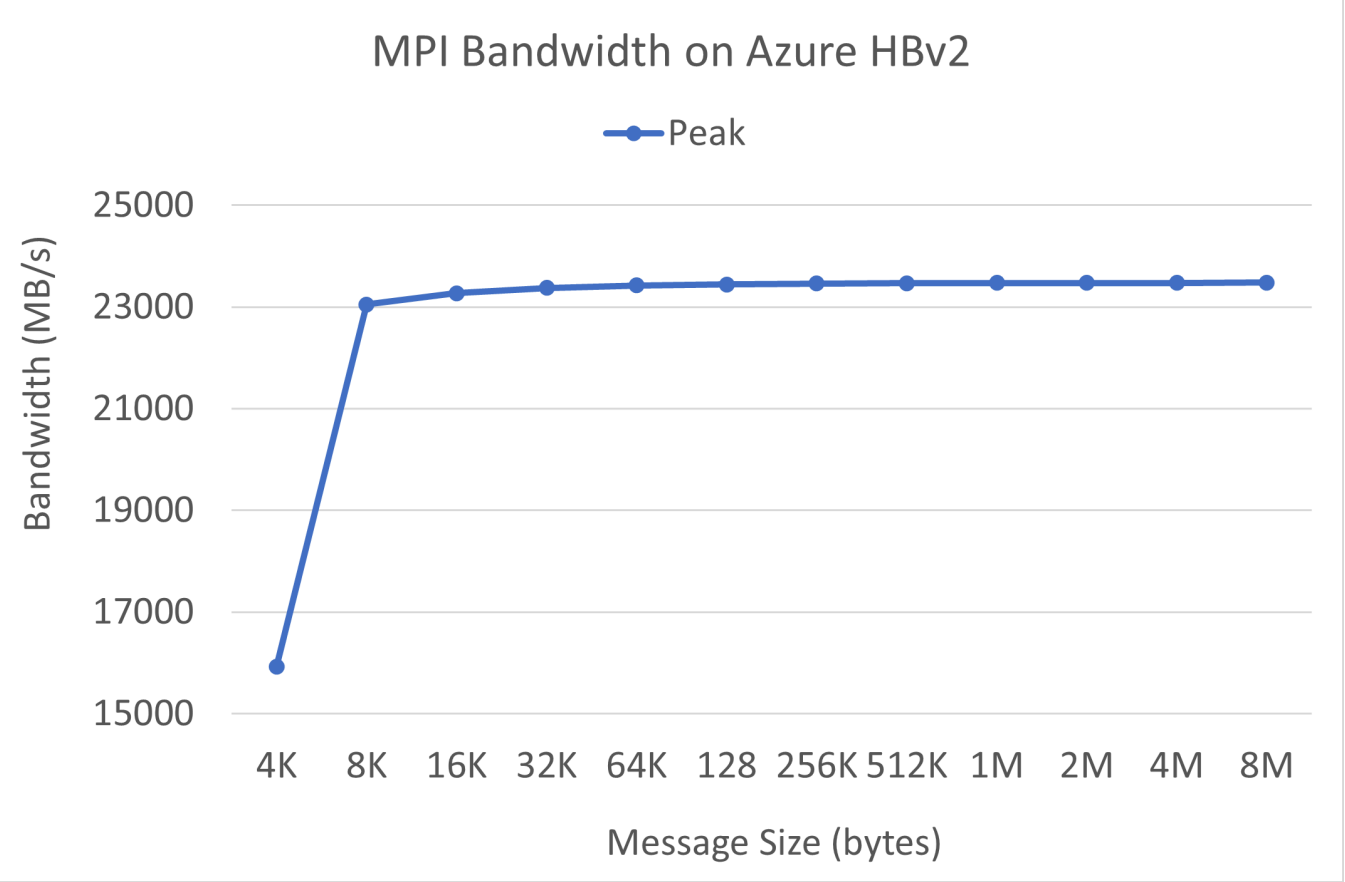


MPI bandwidth

MPI bandwidth test from the OSU microbenchmark suite is run. Sample scripts are on GitHub.

Bash

```
./mvapich2-2.3.install/bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=
[INSERT CORE #] ./mvapich2-2.3/osu_benchmarks/mpi/pt2pt/osu_bw
```



Mellanox Perftest

The Mellanox Perftest package has many InfiniBand tests such as latency (ib_send_lat) and bandwidth (ib_send_bw). The following command serves as an example.

Bash

```
numactl --physcpubind=[INSERT CORE #]  ib_send_lat -a
```

Next steps

- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/11/2026

HBv3-series virtual machine overview

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

An HBv3-series server features 2 * 64-core EPYC 7V73X CPUs for a total of 128 physical "Zen3" cores with AMD 3D V-Cache. Simultaneous Multithreading (SMT) is disabled on HBv3. These 128 cores are divided into 16 sections (8 per socket), each section containing 8 processor cores with uniform access to a 96 MB L3 cache. Azure HBv3 servers also run the following AMD BIOS settings:

Output

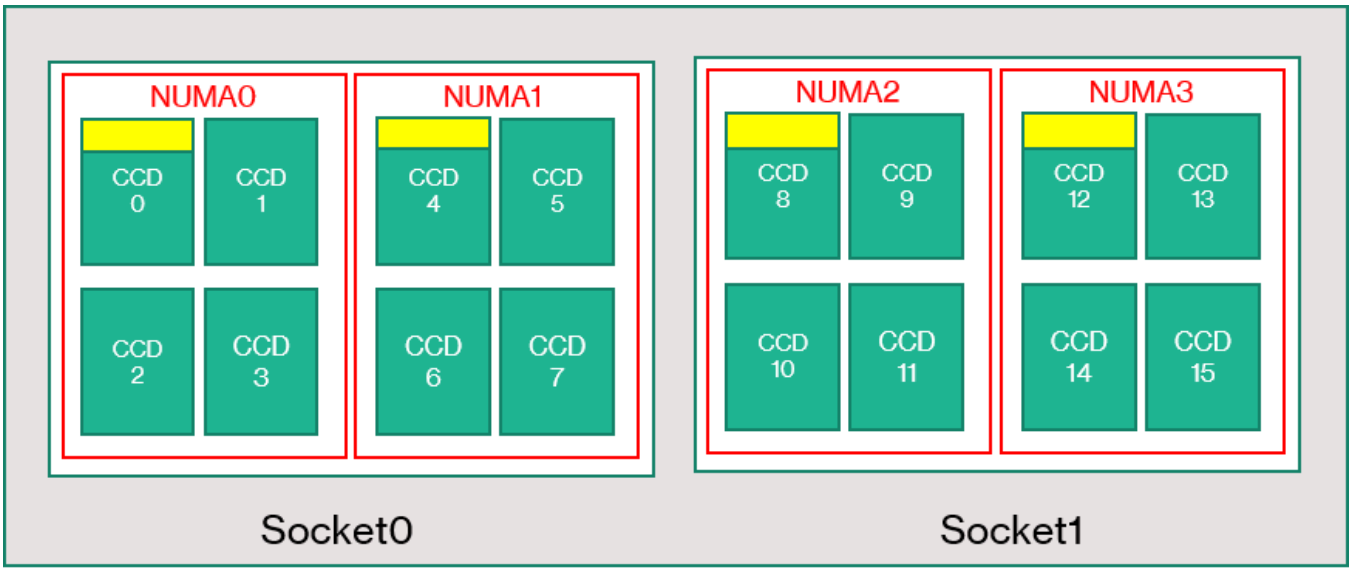
```
Nodes per Socket (NPS) = 2
L3 as NUMA = Disabled
NUMA domains within VM OS = 4
C-states = Enabled
```

As a result, the server boots with 4 NUMA domains (2 per socket). Each domain is 32 cores in size. Each NUMA has direct access to 4 channels of physical DRAM operating at 3,200 MT/s.

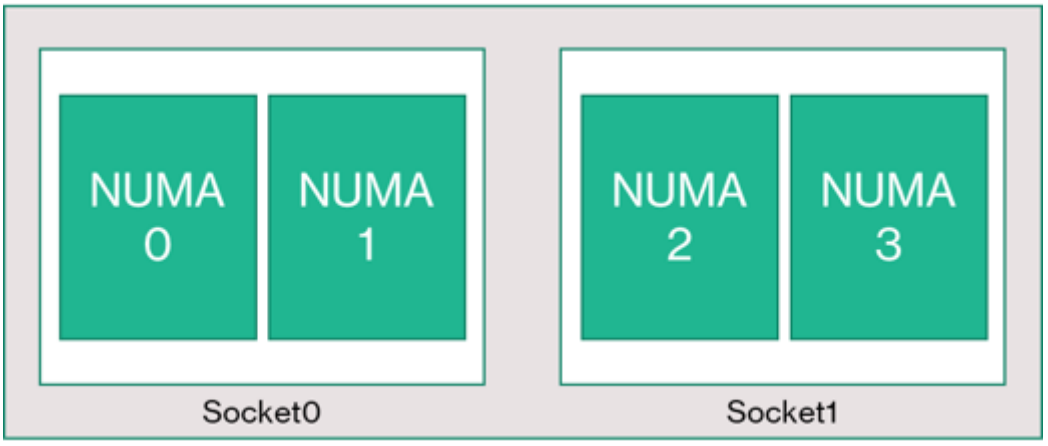
To provide room for the Azure hypervisor to operate without interfering with the VM, we reserve 8 physical cores per server.

VM topology

The following diagram shows the topology of the server. We reserve these 8 hypervisor host cores (yellow) symmetrically across both CPU sockets, taking the first 2 cores from specific Core Complex Dies (CCDs) on each NUMA domain, with the remaining cores for the HBv3-series VM (green).



The CCD boundary isn't equivalent to a NUMA boundary. On HBv3, a group of four consecutive (4) CCDs is configured as a NUMA domain, both at the host server level and within a guest VM. Thus, all HBv3 VM sizes expose 4 NUMA domains that appear to an OS and application as shown. 4 uniform NUMA domains, each with different number of cores depending on the specific HBv3 VM size.



Each HBv3 VM size is similar in physical layout, features, and performance of a different CPU from the AMD EPYC 7003-series, as follows:

Expand table

HBv3-series VM size	NUMA domains	Cores per NUMA domain	Similarity with AMD EPYC
Standard_HB120rs_v3	4	30	Dual-socket EPYC 7773X
Standard_HB120-96rs_v3	4	24	Dual-socket EPYC 7643
Standard_HB120-64rs_v3	4	16	Dual-socket EPYC 7573X
Standard_HB120-32rs_v3	4	8	Dual-socket EPYC 7373X
Standard_HB120-16rs_v3	4	4	Dual-socket EPYC 72F3

> **Note**
>
> The constrained cores VM sizes only reduce the number of physical cores exposed to the VM. All global shared assets (RAM, memory bandwidth, L3 cache, GMI and xGMI connectivity, InfiniBand, Azure Ethernet network, local SSD) stay constant. This capability allows a customer to pick a VM size best tailored to a given set of workload or software licensing needs.

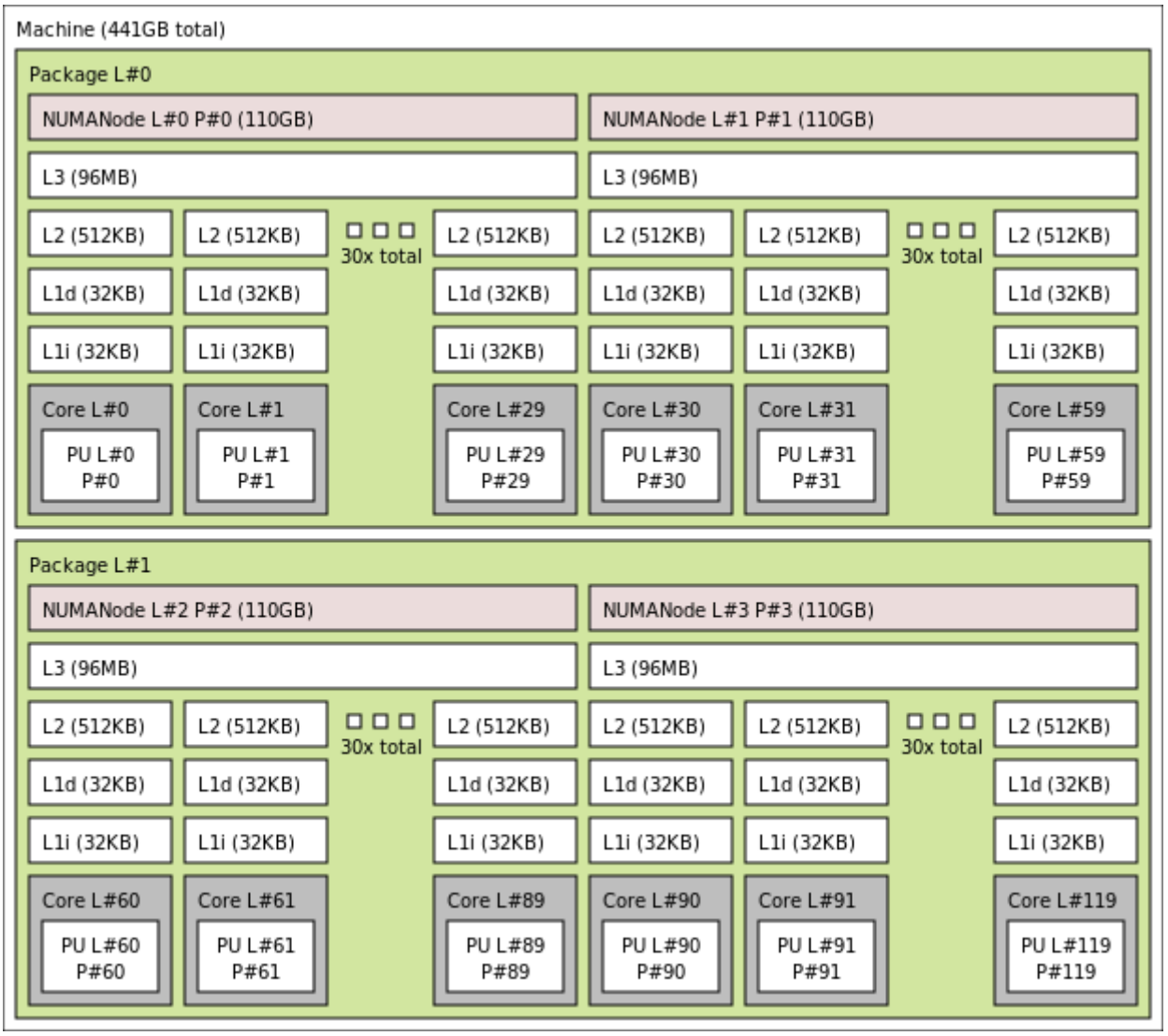
The virtual NUMA mapping of each HBv3 VM size is mapped to the underlying physical NUMA topology. The hardware topology is shown directly, without potentially misleading abstraction.

The exact topology for the various HBv3 VM size appears as follows using the output of lstopo :

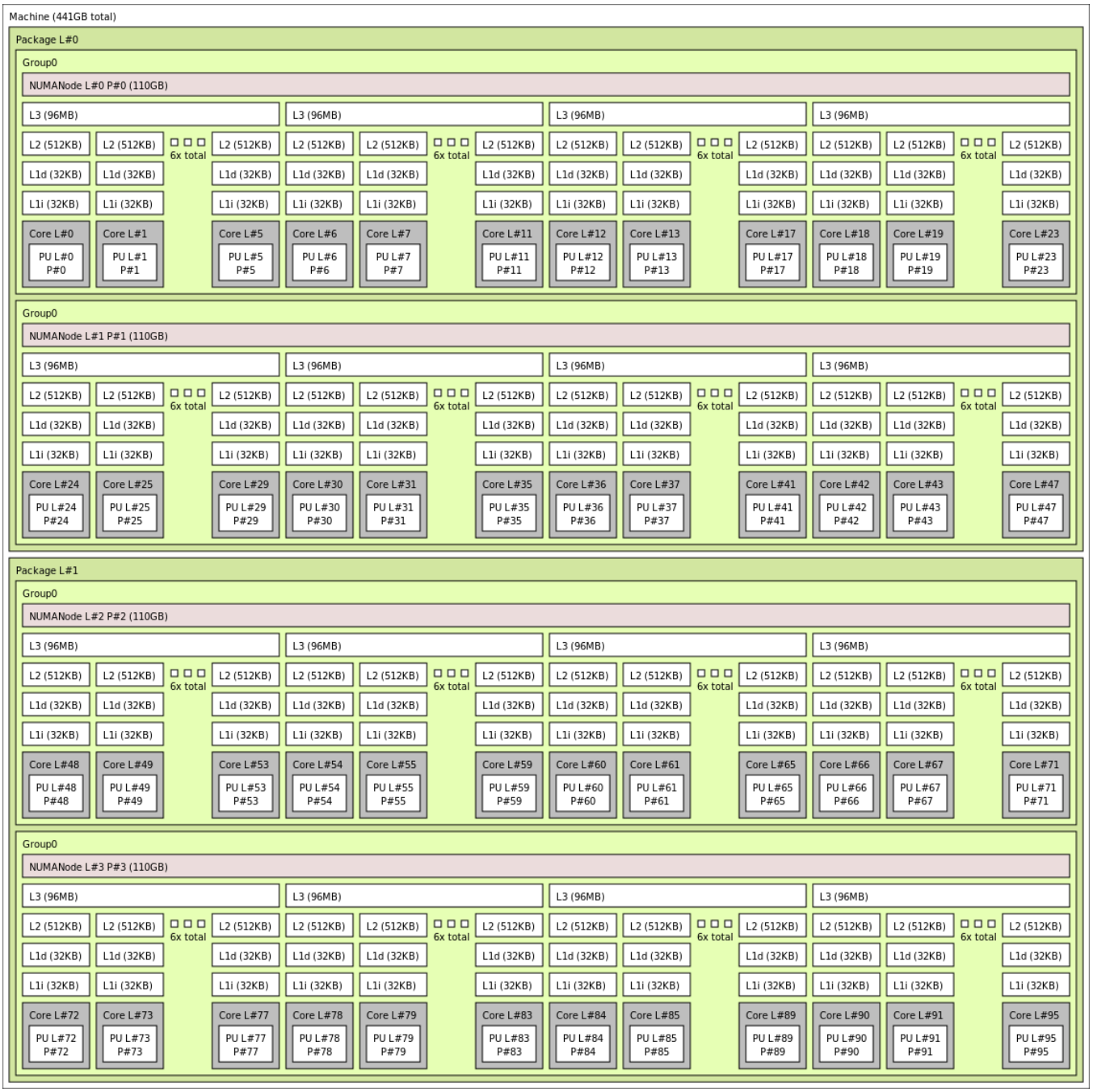
Bash

```
lstopo-no-graphics --no-io --no-legend --of txt
```

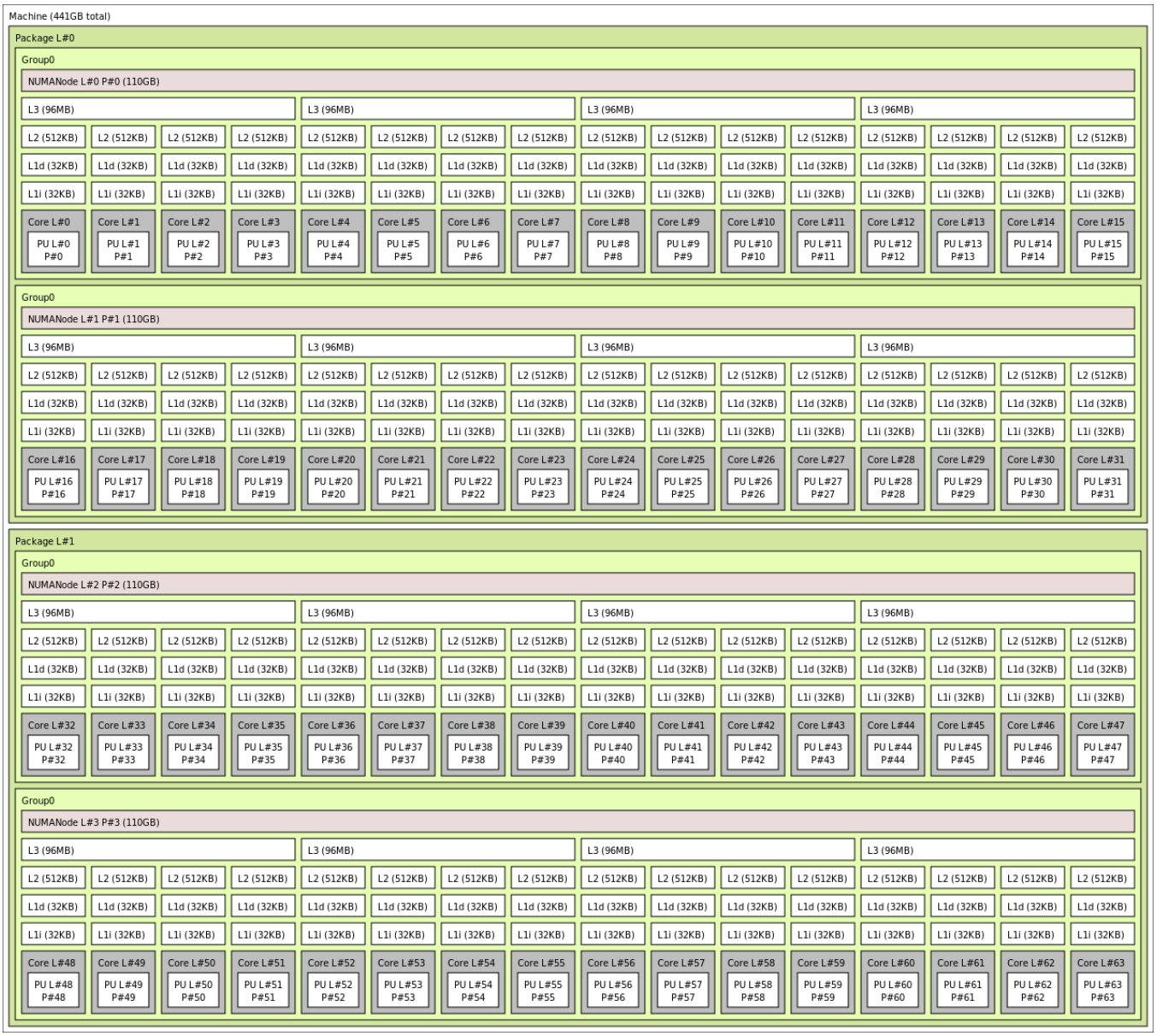
▼ Click to view lstopo output for Standard_HB120rs_v3



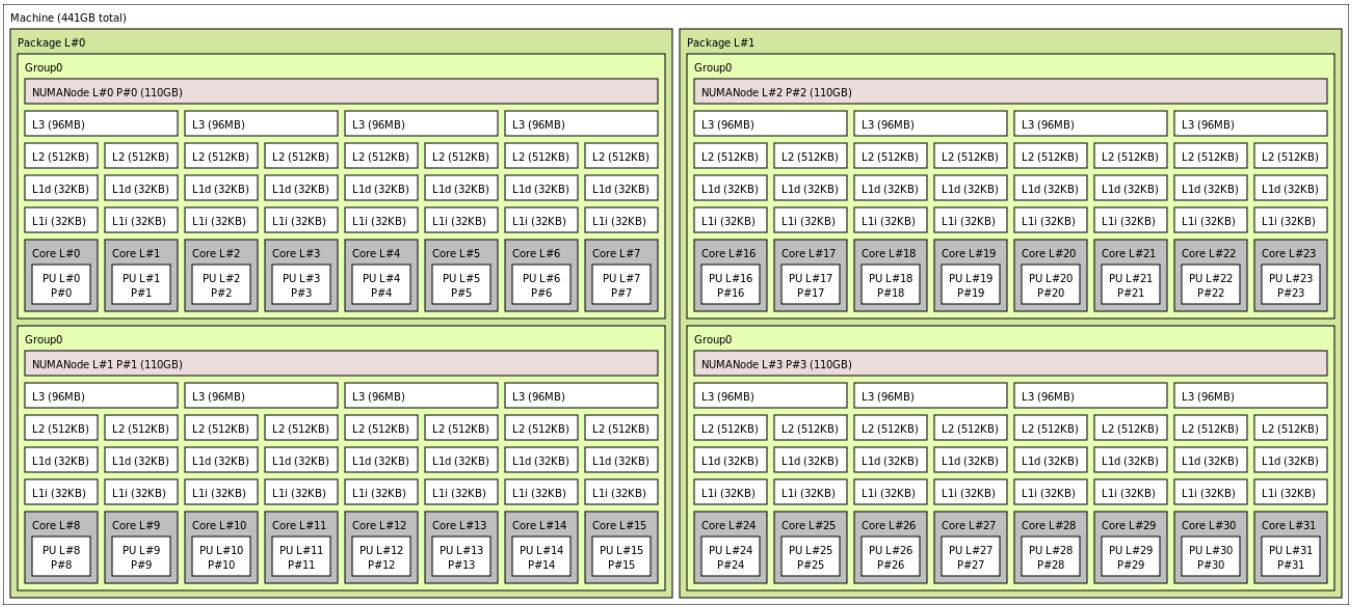
▼ Click to view lstopo output for Standard_HB120rs-96_v3



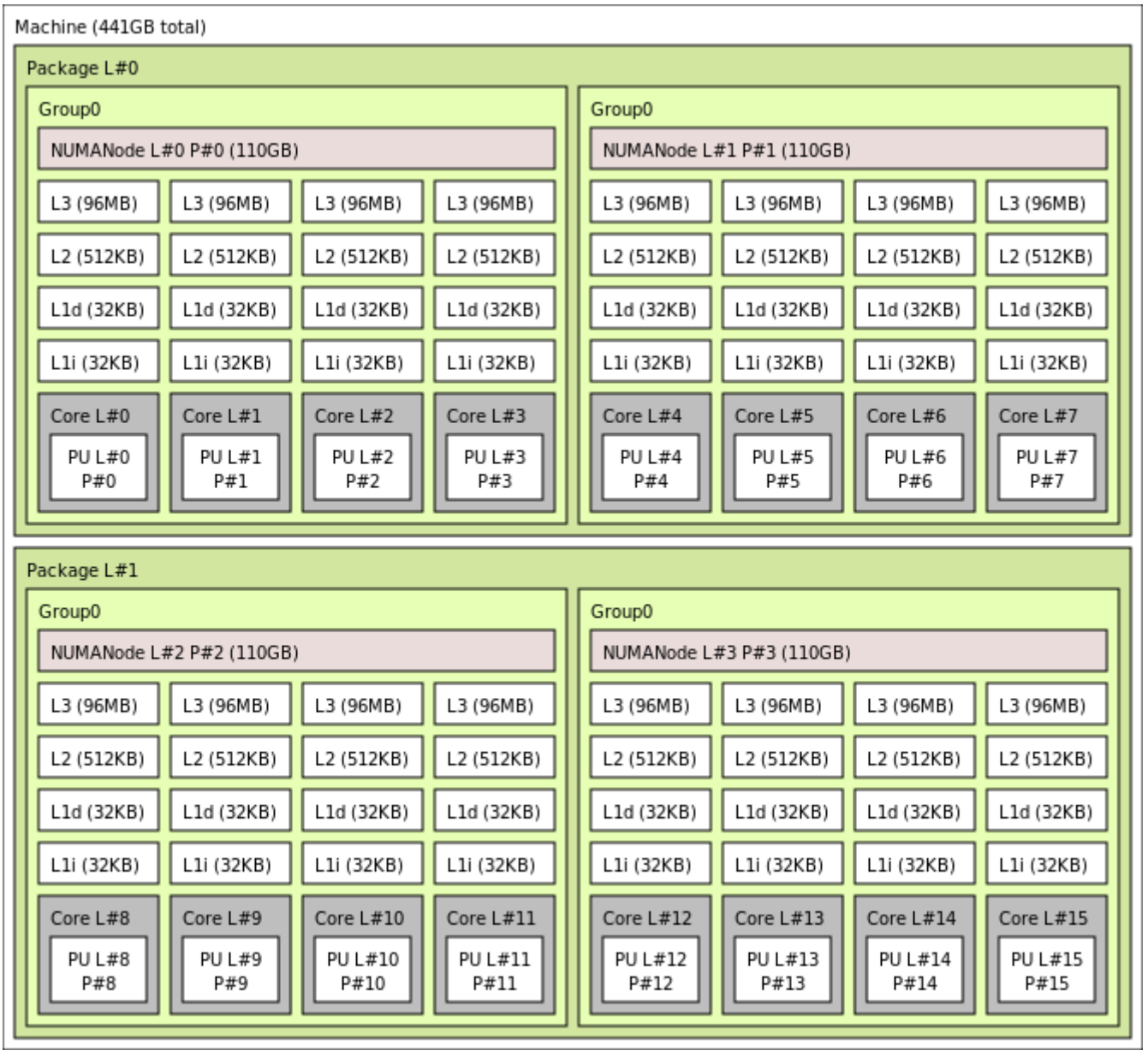
▼ Click to view lstopo output for Standard_HB120rs-64_v3



▼ Click to view lstopo output for Standard_HB120rs-32_v3



▼ Click to view lstopo output for Standard_HB120rs-16_v3



InfiniBand networking

HBv3 VMs also feature NVIDIA Mellanox HDR InfiniBand network adapters (ConnectX-6) operating at up to 200 Gigabits/sec. The NIC is passed through to the VM via SRIOV, enabling network traffic to bypass the hypervisor. As a result, customers load standard Mellanox OFED drivers on HBv3 VMs as they would a bare metal environment.

HBv3 VMs support Adaptive Routing, the Dynamic Connected Transport (DCT, along with standard RC and UD transports), and hardware-based offload of MPI collectives to the onboard processor of the ConnectX-6 adapter. These features enhance application performance, scalability, and consistency, and usage of them is recommended.

Temporary storage

HBv3 VMs feature 3 physically local SSD devices. One device is preformatted to serve as a page file and it appeared within your VM as a generic "SSD" device.

Two other, larger SSDs are provided as unformatted block NVMe devices via NVMeDirect. As the block NVMe device bypasses the hypervisor, it has higher bandwidth, higher IOPS, and lower latency per IOP.

When paired in a striped array, the NVMe SSD provides up to 7 GB/s reads and 3 GB/s writes, and up to 186,000 IOPS (reads) and 201,000 IOPS (writes) for deep queue depths.

Hardware specifications

Expand table

Hardware specifications	HBv3-series VMs
Cores	120, 96, 64, 32, or 16 (SMT disabled)
CPU	AMD EPYC 7V73X
CPU Frequency (non-AVX)	3.0 GHz (all cores), 3.5 GHz (up to 10 cores)
Memory	448 GB (RAM per core depends on VM size)
Local Disk	2 * 960 GB NVMe (block), 480 GB SSD (page file)
Infiniband	200 Gb/s Mellanox ConnectX-6 HDR InfiniBand
Network	50 Gb/s Ethernet (40 Gb/s usable) Azure second Gen SmartNIC

Software specifications

Expand table

Software specifications	HBv3-series VMs
Max MPI Job Size	36,000 cores (300 VMs in a single Virtual Machine Scale Set with singlePlacementGroup=true)
MPI Support	HPC-X, Intel MPI, OpenMPI, MVAPICH2, MPICH
Additional Frameworks	UCX, libfabric, PGAS
Azure Storage Support	Standard and Premium Disks (maximum 32 disks)
OS Support for SRIOV RDMA	RHEL 7.9+, Ubuntu 18.04+, SLES 15.4, WinServer 2016+

Software specifications	HBv3-series VMs
Recommended OS for Performance	Windows Server 2019+
Orchestrator Support	Azure CycleCloud, Azure Batch, AKS; cluster configuration options

> **Note**
>
> Windows Server 2012 R2 isn't supported on HBv3 and other VMs with more than 64 (virtual or physical) cores. For more details, see Supported Windows guest operating systems for Hyper-V on Windows Server.

> **Important**
>
> This document references a release version of Linux that's nearing or at, End of Life(EOL). Consider updating to a more current version.

Next steps

- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/11/2026

HBv3 sizes series

Summarize this article for me

HBv3-series Virtual Machines (VMs) are designed for very demanding computing tasks. They help with things like studying how liquids move, analyzing structures, predicting weather, and processing earthquake data. They also support oil reservoir modeling and testing computer chip designs. HBv3 VMs feature up to 120 AMD EPYC™ 7V73X (Milan-X) CPU cores, 448 GB of RAM, and no simultaneous multithreading. HBv3-series VMs also provide 350 GB/sec of memory bandwidth (amplified up to 630 GB/s), up to 96 MB of L3 cache per core (1536 MB total per VM), up to 7 GB/s of block device SSD performance, and clock frequencies up to 3.5 GHz. All HBv3-series VMs feature 200 Gb/sec High Data Rate (HDR) InfiniBand from NVIDIA Networking to enable supercomputer-scale Message Pass Interface (MPI) workloads. These VMs are connected in a nonblocking fat tree for optimized and consistent RDMA performance. The HDR InfiniBand fabric also supports Adaptive Routing and Dynamic Connected Transport (DCT), along with standard Reliable Connection (RC) and Unreliable Datagram (UD) transports. These features improve performance, make scaling easier, and keep results consistent.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	16 - 120 vCPUs	AMD EPYC 7V73X (Milan-X) [x86-64]
L3 Cache	1536 MB	
Memory	448 GB	350 GB/s
Local Storage	1 Temp Disk 2 NVMe Disks	480 GiB 960 GiB
Remote Storage	32 Disks	
Network	8 vNICs 1 InfiniBand HDR NIC	40 Gb/s 200 Gb/s
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (MB)	Memory Bandwidth (GB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HB120rs_v3	120	448	1536	350	1.9	3.5	3
Standard_HB120-96rs_v3	96	448	1536	350	1.9	3.5	3
Standard_HB120-64rs_v3	64	448	1536	350	1.9	3.5	3
Standard_HB120-32rs_v3	32	448	1536	350	1.9	3.5	3
Standard_HB120-16rs_v3	16	448	1536	350	1.9	3.5	3

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported

Feature name	Support status
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Backend Network	InfiniBand HDR

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.

For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 03/10/2026

HBv3-series virtual machine performance

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Performance expectations using common HPC microbenchmarks are as follows:

Expand table

Workload	HBv3
STREAM Triad	330-350 GB/s (amplified up to 630 GB/s)
High-Performance Linpack (HPL)	4 TF (Rpeak, FP64), 8 TF (Rpeak, FP32) for 120-core VM size
RDMA latency & bandwidth	1.2 microseconds (1 byte), 192 GB/s (one-way)
FIO on local NVMe SSDs (RAID0)	7 GB/s reads, 3 GB/s writes; 186k IOPS reads, 201k IOPS writes

Process pinning

Process pinning works well on HBv3-series VMs because we expose the underlying silicon as-is to the guest VM. We strongly recommend process pinning for optimal performance and consistency.

MPI latency

The MPI latency test from the OSU microbenchmark suite can be executed as shown. Sample scripts are on GitHub.

Bash

```
./bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=[INSERT CORE #]
./osu_latency
```

MPI bandwidth

The MPI bandwidth test from the OSU Micro-Benchmark (OMB) suite can be executed as shown. Sample scripts are on GitHub.

Bash

```
./mvapich2-2.3.install/bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=
[INSERT CORE #] ./mvapich2-2.3/osu_benchmarks/mpi/pt2pt/osu_bw
```

Mellanox Perftest

The Mellanox Perftest package has many InfiniBand tests such as latency (ib_send_lat) and bandwidth (ib_send_bw). The following command serves as an example.

Bash

```
numactl --physcpubind=[INSERT CORE #]  ib_send_lat -a
```

Next steps

- Learn about scaling MPI applications.
- Review the performance and scalability results of HPC applications on the HBv3 VMs at the TechCommunity article.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/11/2026

HBv4-series virtual machine overview

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

An HBv4-series server features 2 * 96-core EPYC 9V33X CPUs for a total of 192 physical "Zen4" cores with AMD 3D-V Cache. Simultaneous Multithreading (SMT) is disabled on HBv4. These 192 cores are divided into 24 sections (12 per socket), each section containing eight processor cores with uniform access to a 96 MB L3 cache. Azure HBv4 servers also run the following AMD BIOS settings:

Bash

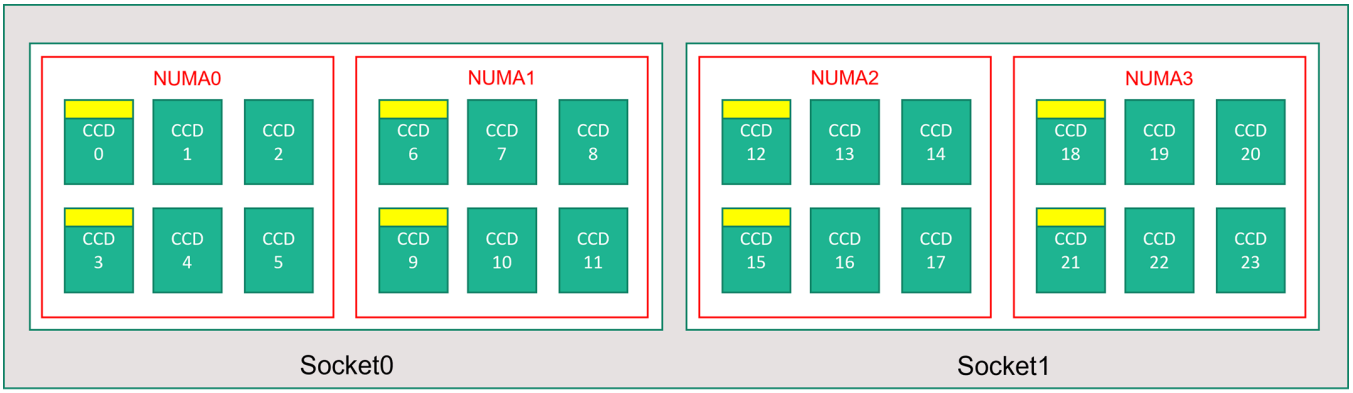
```
Nodes per Socket (NPS) = 2
L3 as NUMA = Disabled
NUMA domains within VM OS = 4
C-states = Enabled
```

As a result, the server boots with four NUMA domains (2 per socket) each 48-cores in size. Each NUMA has direct access to six channels of physical DRAM.

To provide room for the Azure hypervisor to operate without interfering with the VM, we reserve 16 physical cores per server.

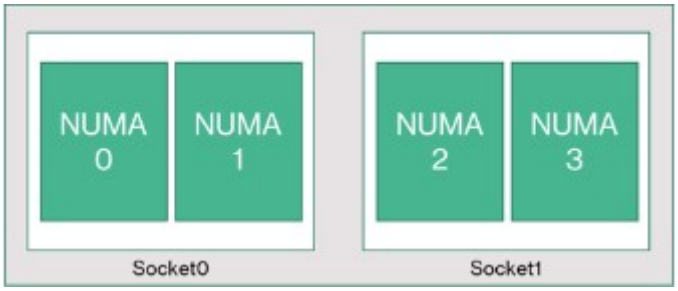
VM topology

The following diagram shows the topology of the server. We reserve these 16 hypervisor host cores (yellow) symmetrically across both CPU sockets, taking the first two cores from specific Core Complex Dies (CCDs) in each NUMA domain, with the remaining cores for the HBv4-series VM (green).



The CCD boundary is different from a NUMA boundary. On HBv4, a group of six (6) consecutive CCDs is configured as a NUMA domain, both at the host server level and within a guest VM.

Thus, all HBv4 VM sizes expose four uniform NUMA domains that appear to an OS and application as shown below, each with different number of cores depending on the specific HBv4 VM size.



Each HBv4 VM size is similar in physical layout, features, and performance of a different CPU from the AMD EPYC 9V33X, as follows:

Expand table

HBv4-series VM size	NUMA domains	Cores per NUMA domain	Similarity with AMD EPYC
Standard_HB176rs_v4	4	44	Dual-socket EPYC 9V33X
Standard_HB176-144rs_v4	4	36	Dual-socket EPYC 9V33X
Standard_HB176-96rs_v4	4	24	Dual-socket EPYC 9V33X
Standard_HB176-48rs_v4	4	12	Dual-socket EPYC 9V33X
Standard_HB176-24rs_v4	4	6	Dual-socket EPYC 9V33X

> **Note**
>
> - The constrained cores VM sizes only reduce the number of physical cores exposed to the VM. All global shared assets (RAM, memory bandwidth, L3 cache, GMI and xGMI connectivity, InfiniBand, Azure Ethernet network, local SSD) stay constant. This allows a customer to pick a VM size best tailored to a given set of workload or software licensing needs.
> - We are currently experiencing a core mapping issue on the HBv4-series parent size Standard_HB176rs_v4.

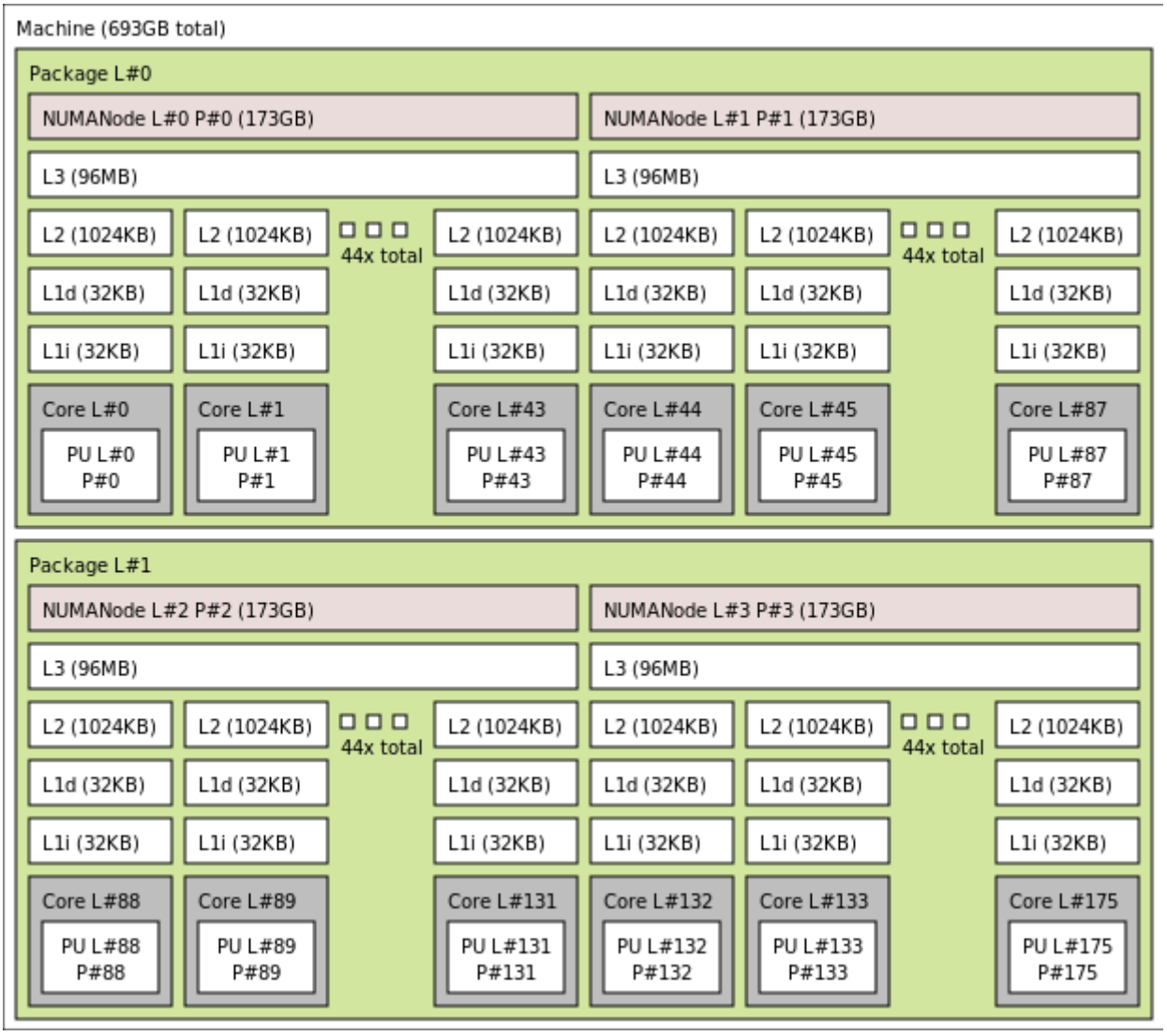
The virtual NUMA mapping of each HBv4 VM size is mapped to the underlying physical NUMA topology. There's no potential misleading abstraction of the hardware topology.

The exact topology for the various HBv4 VM size appears as follows using the output of lstopo:

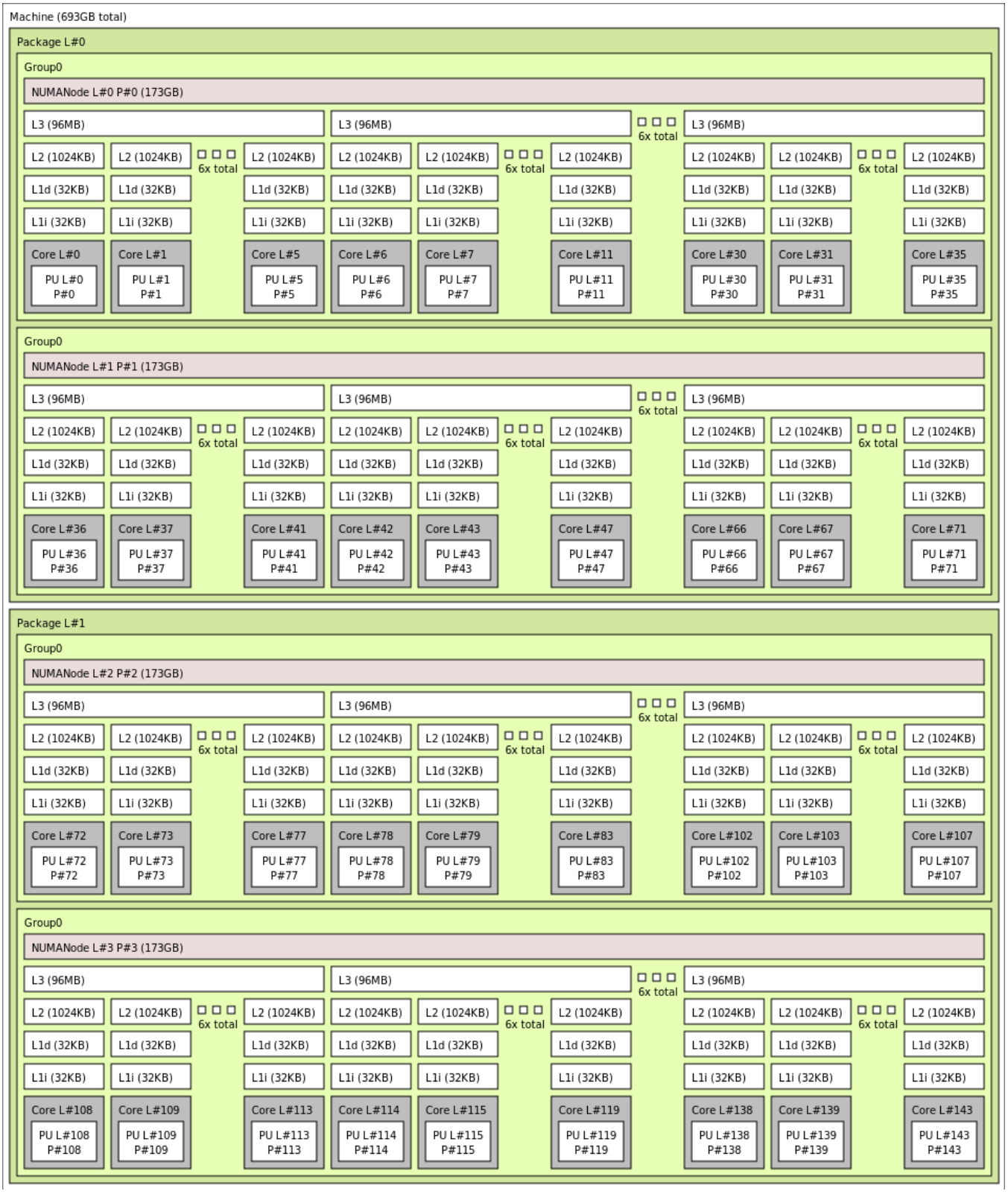
Bash

```
lstopo-no-graphics --no-io --no-legend --of txt
```

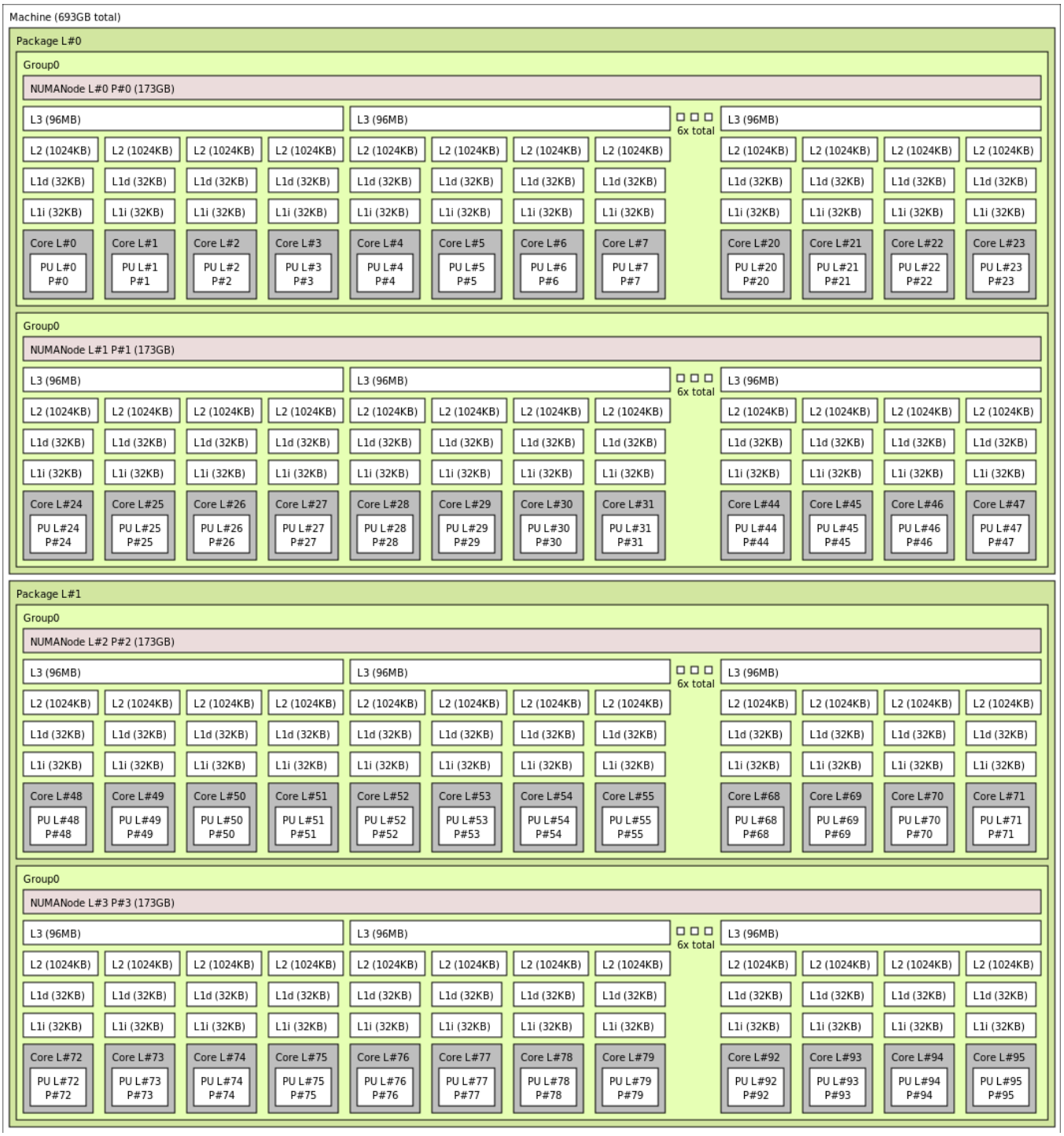
▼ Select to view lstopo output for Standard_HB176rs_v4



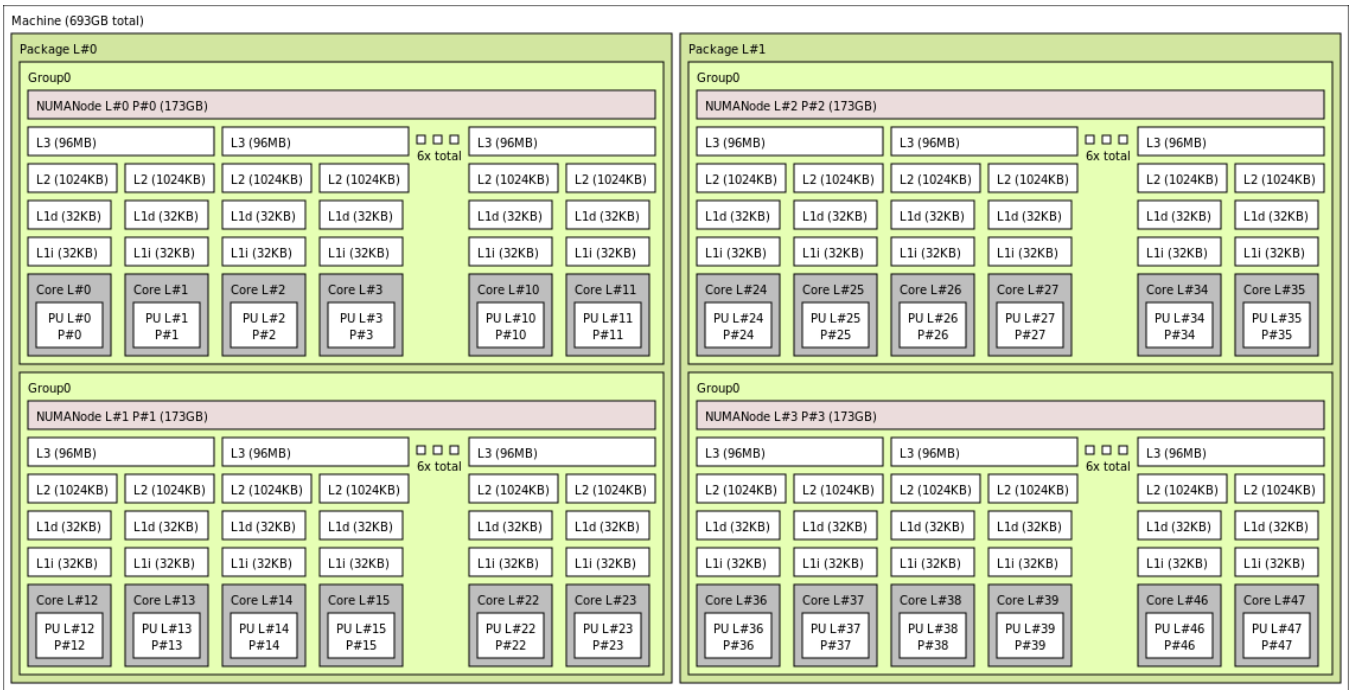
▼ Select to view lstopo output for Standard_HB176-144rs_v4



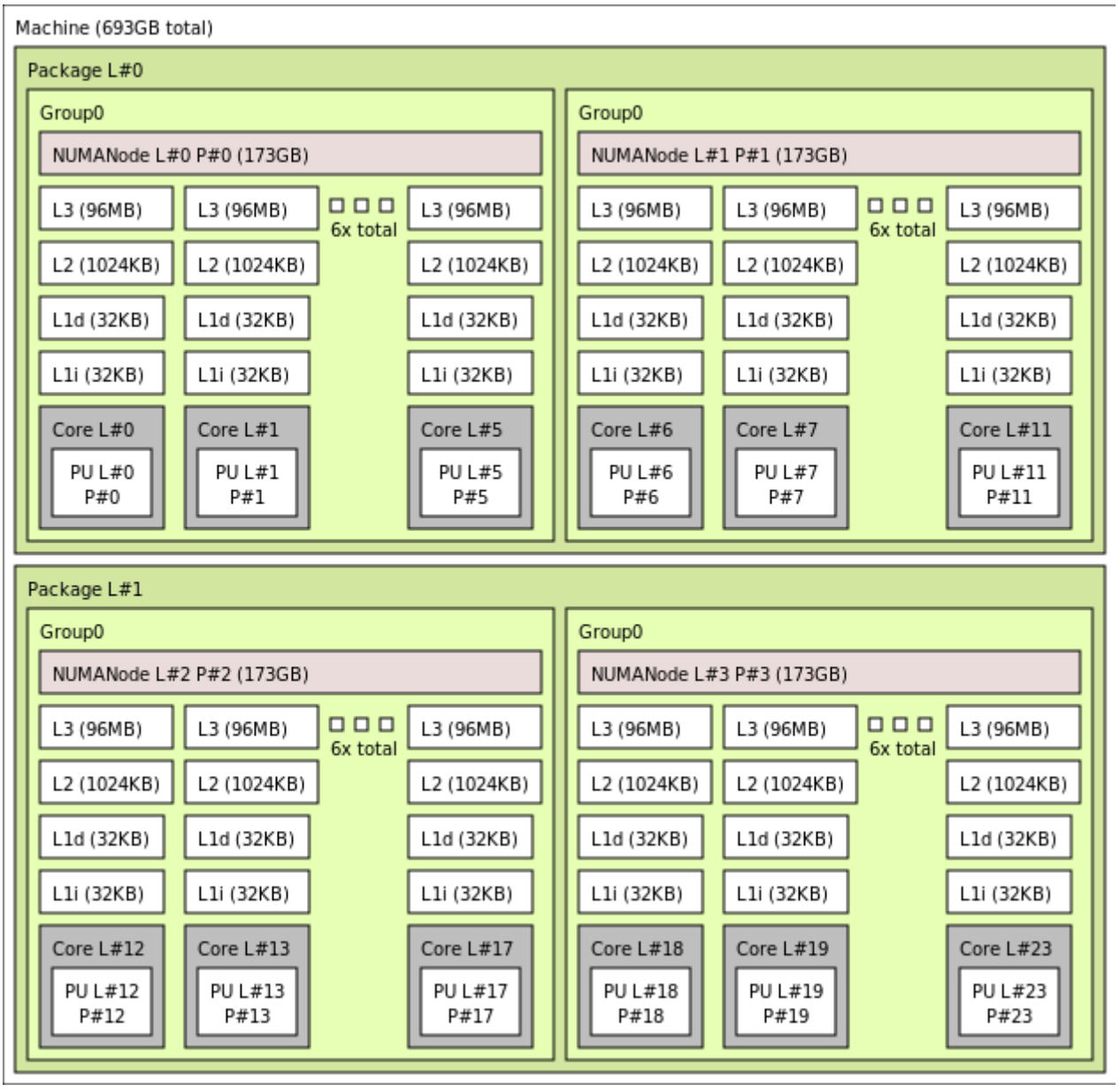
▼ Select to view lstopo output for Standard_HB176-96rs_v4



▼ Select to view lstopo output for Standard_HB176-48rs_v4



▼ Select to view lstopo output for Standard_HB176-24rs_v4



InfiniBand networking

HBv4 VMs also feature NVIDIA Mellanox NDR InfiniBand network adapters (ConnectX-7) operating at up to 400 Gigabits/sec. The NIC is passed through to the VM via SRIOV, enabling network traffic to bypass the hypervisor. As a result, customers load standard Mellanox OFED drivers on HBv4 VMs as they would a bare metal environment.

HBv4 VMs support Adaptive Routing, Dynamic Connected Transport (DCT, in addition to the standard RC and UD transports), and hardware-based offload of MPI collectives to the onboard processor of the ConnectX-7 adapter. These features enhance application performance, scalability, and consistency, and usage of them is recommended.

Temporary storage

HBv4 VMs feature three physically local SSD devices. One device is preformatted to serve as a page file and it appeared within your VM as a generic "SSD" device.

Two other, larger SSDs are provided as unformatted block NVMe devices via NVMeDirect. As the block NVMe device bypasses the hypervisor, it has higher bandwidth, higher IOPS, and lower latency per IOP.

When paired in a striped array, the NVMe SSD provides up to 12 GB/s reads and 7 GB/s writes, and up to 186,000 IOPS (reads) and 201,000 IOPS (writes) for deep queue depths.

Hardware specifications

Expand table

Hardware specifications	HBv4-series VMs
Cores	176, 144, 96, 48, or 24 (SMT disabled)
CPU	AMD EPYC 9V33X
CPU Frequency (non-AVX)	2.55 GHz base, 3.7 GHz peak boost
Memory	768 GB (RAM per core depends on VM size)
Local Disk	2 * 1.8-TB NVMe (block), 480 GB SSD (page file)
InfiniBand	400 Gb/s Mellanox ConnectX-7 NDR InfiniBand
Network	100 Gb/s Ethernet (80 Gb/s usable) Azure second Gen SmartNIC

Software specifications

Expand table

Software specifications	HBv4-series VMs
Max MPI Job Size	52,800 cores (300 VMs in a single virtual machine scale set with singlePlacementGroup=true)
MPI Support	HPC-X (2.13 or higher), Intel MPI (2021.7.0 or higher), OpenMPI (4.1.3 or higher), MVAPICH2 (2.3.7 or higher), MPICH (4.1 or higher)
Additional Frameworks	UCX, libfabric, PGAS, or other InfiniBand based runtimes
Azure Storage Support	Standard and Premium Disks (maximum 32 disks), Azure NetApp Files, Azure Files, Azure HPC Cache, Azure Managed Lustre File System
Supported and Validated OS	AlmaLinux 8.6, 8.7, Ubuntu 20.04+
Recommended OS for Performance	AlmaLinux HPC 8.7, Ubuntu-HPC 20.04+
Orchestrator Support	Azure CycleCloud, Azure Batch, AKS; cluster configuration options

Note

- These VMs support only Generation 2.
- Official kernel-level support from AMD starts with RHEL 8.6 and AlmaLinux 8.6, which is a derivative of RHEL.
- Windows Server 2012 R2 isn't supported on HBv4 and other VMs with more than 64 (virtual or physical) cores. For more information, see Supported Windows guest operating systems for Hyper-V on Windows Server. Windows Server 2022 is required for 144 and 176 core sizes, Windows Server 2016 also works for 24, 48, and 96 core sizes, Windows Server works for only 24 and 48 core sizes.

Important

Recommended image URN: almalinux:almalinux-hpc:8_7-hpc-gen2:8.7.2023060101, To deploy this image over Azure CLI, ensure the following parameters are included **--plan 8_7-hpc-gen2 --product almalinux-hpc --publisher almalinux**. For scaling tests please use the recommended URN along with the new HPC-X tarball.

> **Note**
>
> - NDR support is added in UCX 1.13 or later. Older UCX versions will report the above runtime error. UCX Error: Invalid active speed `[1677010492.951559] [updsb-vm-0:2754 :0] ib_iface.c:1549 UCX ERROR Invalid active_speed on mlx5_ib0:1: 128`.
> - Ibstat shows low speed (SDR): Older Mellanox OFED (MOFED) versions don't support NDR and it may report slower IB speeds. Use MOFED versions MOFED 5.6-1.0.3.3 or above.

Next steps

- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/10/2026

HBv4 sizes series

Summarize this article for me

HBv4-series VMs are optimized for many high-performance computing (HPC) workloads. These include:

- Computational fluid dynamics
- Finite element analysis
- Frontend and backend Electronic Design Automation (EDA)
- Rendering and graphics work
- Molecular dynamics
- Computational geoscience
- Weather simulation
- Financial risk analysis

HBv4 VMs feature up to 176 AMD EPYC™ 9V33X ("Genoa-X") CPU cores with AMD's 3D V-Cache, clock frequencies up to 3.7 GHz, and no simultaneous multithreading. HBv4-series VMs also provide 768 GB of RAM, 2304 MB L3 cache. Each VM has 2304 MB of L3 cache that delivers up to 5.7 TB/s of bandwidth. This boosts DRAM bandwidth of 780 GB/s, giving an average of 1.2 TB/s of effective memory speed for many workloads. The VMs also provide up to 12 GB/s (reads) and 7 GB/s (writes) of block device SSD performance.

All HBv4-series VMs feature 400 Gb/s NDR InfiniBand from NVIDIA Networking to enable supercomputer-scale MPI workloads. These VMs are connected in a non-blocking fat tree for optimized and consistent RDMA performance. NDR continues to support features like Adaptive Routing and the Dynamically Connected Transport (DCT). This newest generation of InfiniBand also brings greater support for offload of MPI collectives, optimized real-world latencies due to congestion control intelligence, and enhanced adaptive routing capabilities. These features enhance application performance, scalability, and consistency, and their usage is recommended.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	24 - 176 vCPUs	AMD EPYC 9V33X (Genoa-X) [x86-64]
L3 Cache	2304 MB	

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Memory	768 GB	780 GB/s
Local Storage	1 Temp Disk 2 NVMe Disks	480 GiB 1800 GiB
Remote Storage	32 Disks	
Network	8 vNICs 1 InfiniBand NDR NIC	80 Gb/s 400 Gb/s
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (MB)	Memory Bandwidth (GB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HB176rs_v4	176	768	2304	780	2.55	3.7	3.7
Standard_HB176-144rs_v4	144	768	2304	780	2.55	3.7	3.7
Standard_HB176-96rs_v4	96	768	2304	780	2.55	3.7	3.7
Standard_HB176-48rs_v4	48	768	2304	780	2.55	3.7	3.7
Standard_HB176-24rs_v4	24	768	2304	780	2.55	3.7	3.7

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Backend Network	InfiniBand NDR

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

HBv4-series virtual machine performance

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Performance expectations using common HPC microbenchmarks are as follows:

Expand table

Workload	HBv4
STREAM Triad	750-780 GB/s of DDR5, up to 5.7 TB/s of 3D V-Cache bandwidth
High-Performance Linpack (HPL)	Up to 7.6 TF (Rpeak, FP64) for 144-core VM size
RDMA latency & bandwidth	< 2 microseconds (1 byte), 400 Gb/s (one-way)
FIO on local NVMe SSDs (RAID0)	12 GB/s reads, 7 GB/s writes; 186k IOPS reads, 201k IOPS writes

Process pinning

Process pinning works well on HBv4-series VMs because we expose the underlying silicon as-is to the guest VM. We strongly recommend process pinning for optimal performance and consistency.

Memory bandwidth test

The STREAM memory test can be run using the scripts in this GitHub repository.

Bash

```
git clone https://github.com/Azure/woc-benchmarking
cd woc-benchmarking/apps/hpc/stream/
sh build_stream.sh
sh stream_run_script.sh $PWD “hbrs_v4”
```

Compute performance test

The HPL benchmark can be run using the script in this GitHub repository.

Bash

```
git clone https://github.com/Azure/woc-benchmarking
cd woc-benchmarking/apps/hpc/hpl
sh hpl_build_script.sh
sh hpl_run_scr_hbv4.sh $PWD
```

MPI latency

The MPI latency test from the OSU microbenchmark suite can be executed as shown. Sample scripts are on GitHub.

Bash

```
module load mpi/hpcx
mpirun -np 2 --host $src,$dst --map-by node -x LD_LIBRARY_PATH
$HPCX_OSU_DIR/osu_latency
```

MPI bandwidth

The MPI bandwidth test from the OSU microbenchmark suite can be executed per below. Sample scripts are on GitHub.

Bash

```
module load mpi/hpcx
mpirun -np 2 --host $src,$dst --map-by node -x LD_LIBRARY_PATH $HPCX_OSU_DIR/osu_bw
```

[!NOTE] Define source(src) and destination(dst).

Mellanox Perftest

The Mellanox Perftest package has many InfiniBand tests such as latency (ib_send_lat) and bandwidth (ib_send_bw). An example command is below.

Console

```
numactl --physcpubind=[INSERT CORE #]  ib_send_lat -a
```

> **Note**
>
> The NUMA node affinity for InfiniBand NIC is NUMA0.

Next steps

- Learn about scaling MPI applications.
- Review the performance and scalability results of HPC applications on the HBv4 VMs at the TechCommunity article.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure HPC Microsoft Community Hub.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/10/2026

HBv5-series virtual machine overview

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

An HBv5-series server features 4 * 96-core 4th Generation AMD EPYC™ CPUs for a total of 384 physical "Zen4" cores with Simultaneous Multithreading (SMT) disabled. These 384 cores are divided into 48 Core Chiplet Dies (CCDs) sections (12 per socket), and each CCD containing eight processor cores with uniform access to a 32 MB L3 cache. Azure HBv5 servers also run the following AMD BIOS settings:

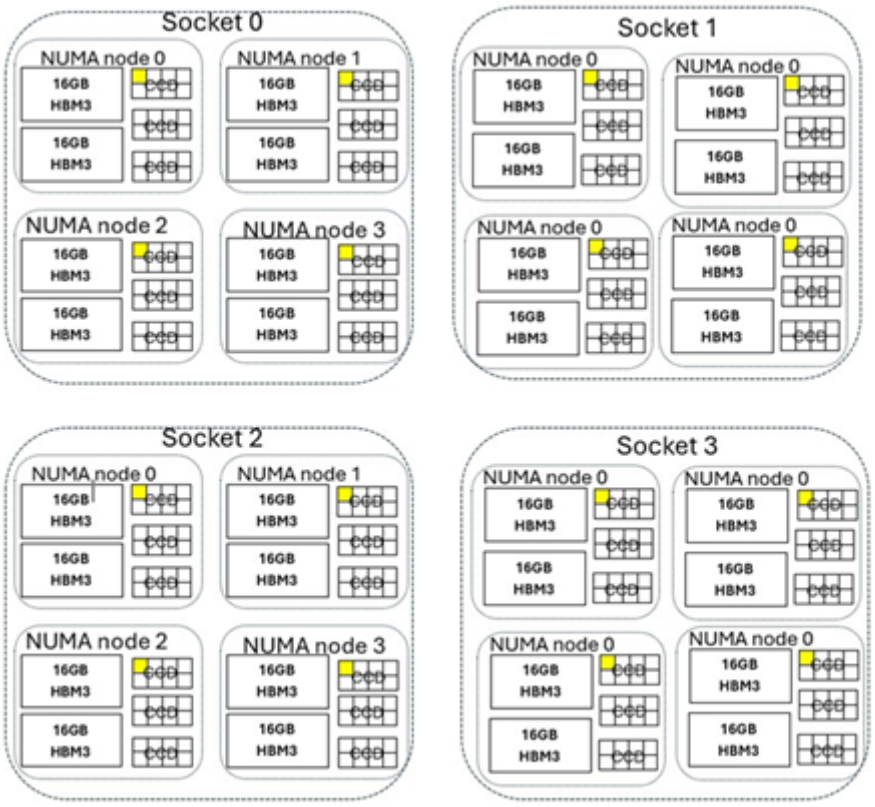
```bash
Nodes per Socket (NPS) = 4
L3 as NUMA = Disabled
Total NUMA domains within VM OS = 16 (4 per socket)
NUMA domains within VM OS = 4
C-states = Enabled
Determinism Mode = Power
```

As a result, the server boots with 16 NUMA domains (4 per socket) each 24-cores in size. Each NUMA has direct access two 16GB HBM3 modules.

To provide room for the Azure hypervisor to operate without interfering with the VM, we reserve 4 physical cores per NUMA and 16 per server.

VM topology

The following diagram shows the topology of the server. We reserve these 16 hypervisor host cores (yellow), taking the first core from specific Core Complex Dies (CCDs) in each NUMA domain, with the remaining cores for the HBv5-series VM (green).



The CCD boundary is different from a NUMA boundary. On HBv5, a group of six (6) consecutive CCDs is configured as a NUMA domain, both at the host server level and within a guest VM. Thus, all HBv5 VM sizes expose four uniform NUMA domains that appear to an OS and application as shown beneath, each with different number of cores depending on the specific HBv5 VM size.

Each HBv5 VM size is similar in physical layout, features, and performance of a different CPU from the AMD EPYC 9V33X, as follows:

Expand table

HBv5-series VM size	NUMA domains	Cores per NUMA domain
Standard_HB368rs_v5	16	23
Standard_HB368-336rs_v5	16	21
Standard_HB368-288rs_v5	16	18
Standard_HB368-240rs_v5	16	15
Standard_HB368-192rs_v5	16	12
Standard_HB368-144rs_v5	16	9
Standard_HB368-96rs_v5	16	6
Standard_HB368-48rs_v5	16	3

> **Note**
>
> - The constrained cores VM sizes only reduce the number of physical cores exposed to the VM. All global shared assets (RAM, memory bandwidth, L3 cache, GMI and xGMI connectivity, InfiniBand, Azure Ethernet network, local SSD) stay constant with the parent VM size. It allows the customer to pick a VM size best tailored to a given set of workload or software licensing needs.
> - Four CPU cores/CCD are needed to saturate memory bandwidth. This requirement means those Standard_HB368-144rs_v5, Standard_HB368-96rs_v5, and Standard_HB368-48rs_v5 VM sizes can't achieve full memory bandwidth of the server.

The virtual NUMA mapping of each HBv5 VM size is mapped to the underlying physical NUMA topology. There's no potential misleading abstraction of the hardware topology.

The exact topology for the various HBv5 VM size appears as follows using the output of lstopo :

Bash

```
lstopo-no-graphics --no-io --no-legend --of txt
```

▼ Select to view lstopo output for Standard_HB368rs_v5



▼ Select to view lstopo output for Standard_HB368-336rs_v5



▼ Select to view lstopo output for Standard_HB368-288rs_v5



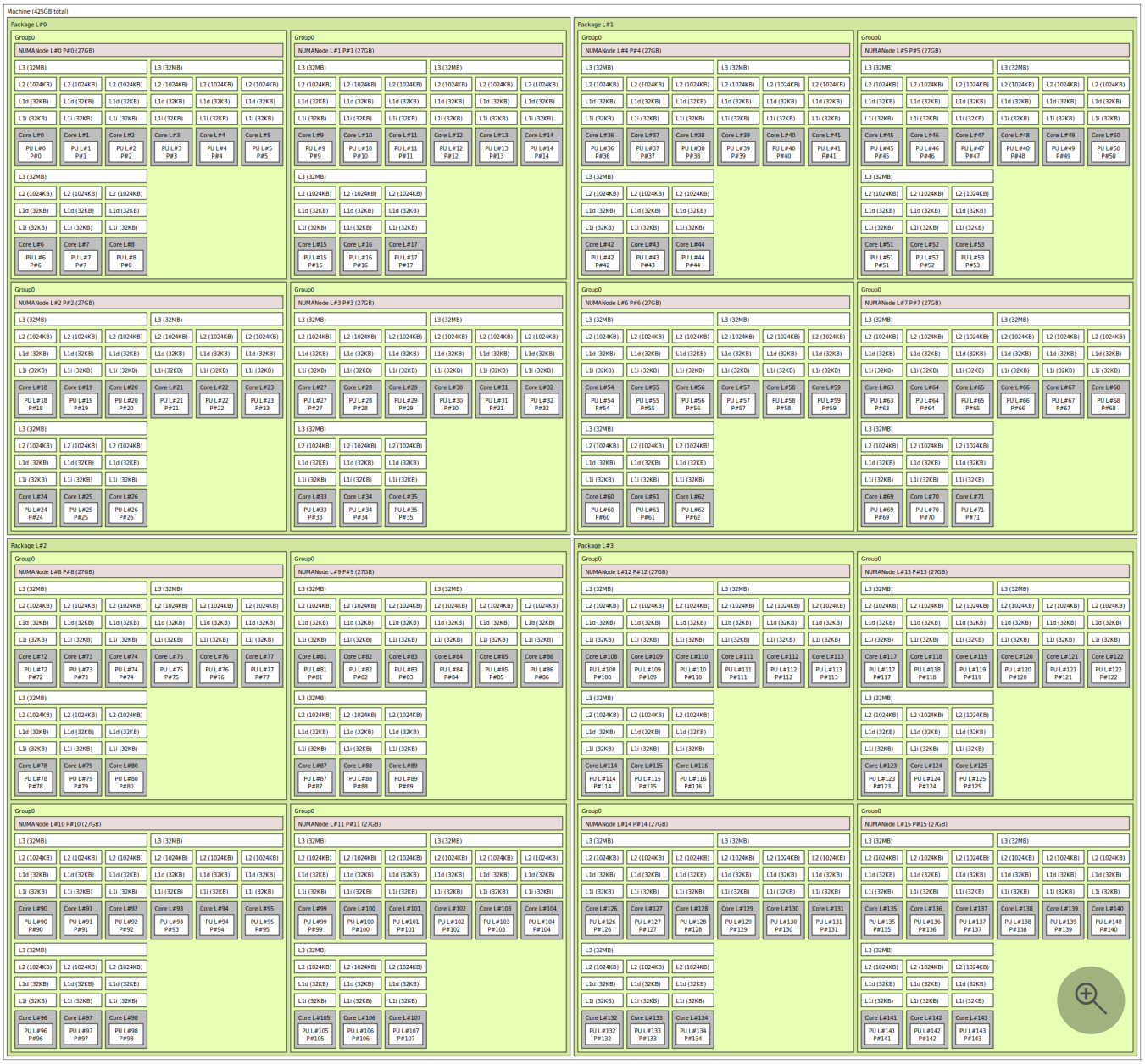
▼ Select to view lstopo output for Standard_HB368-240rs_v5



▼ Select to view lstopo output for Standard_HB368-192rs_v5



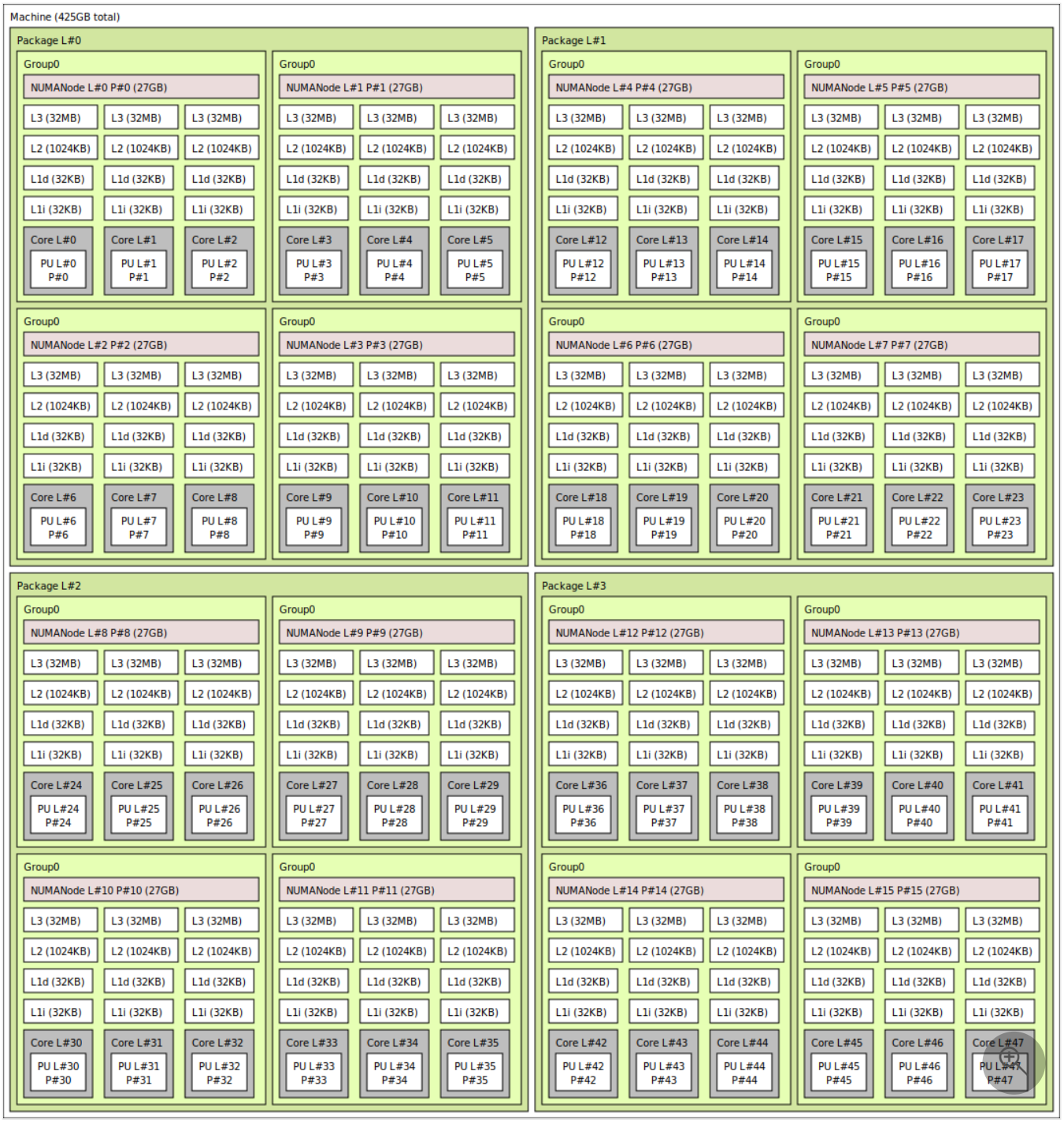
▼ Select to view lstopo output for Standard_HB368-144rs_v5



▼ Select to view lstopo output for Standard_HB368-96rs_v5



▼ Select to view lstopo output for Standard_HB368-48rs_v5



InfiniBand networking

HBv5 VMs also feature 4 NVIDIA Quantum-2 CX7 InfiniBand (NDR) adapters each operating at up to 200 Gigabits/sec for a total of 800 Gigabits/sec per VM. One NIC connects to each of the 4 CPUs on each VM. The NIC is passed through to the VM via SRIOV, enabling network traffic to bypass the hypervisor. As a result, customers load standard Mellanox OFED drivers on HBv5 VMs as they would a bare metal environment.

HBv5 VMs support Adaptive Routing, Dynamic Connected Transport (DCT, in addition to the standard RC and UD transports), and hardware-based offload of MPI collectives to the onboard processor of the ConnectX-7 adapter. These features enhance application performance, scalability, and consistency, and usage of them is recommended.

Best practices for InfiniBand configuration

- Use a validated VM image. Choose an image with tested drivers and NDR InfiniBand-ready software:
  - **Recommended**: AlmaLinux 8.10 from the AlmaLinux HPC shared image gallery
    The Azure HPC team provides access to AlmaLinux 8.10 from the AlmaLinux HPC shared image gallery, and this image is built using Azure HPC VM Image scripts.
  - **Also supported**: Azure HPC marketplace images (Ubuntu-HPC 18.04, Ubuntu-HPC 20.04)

- Select the optimal InfiniBand transport protocol
  - For *smaller-scale jobs*:

    Bash

    ```
    UCX_TLS=rc,sm
    ```

  - For *larger-scale jobs*:

    Bash

    ```
    UCX_TLS=dc,sm UCX_MAX_RNDV_RAILS=1
    ```

- Disable multi-rail for multi-node jobs

  Bash

  ```
  export UCX_MAX_RNDV_RAILS=1
  ```

- Use the latest stable versions
  - UCX: **1.14.0 rc4** (as of September 2025)
  - HPC-X MPI: **hpcx-v2.18-gcc-mlnx_ofed-redhat8-cuda12-x86_64** (as of September 2025)

Best Practices for running MPI Jobs on HBv5

- Apply the *hpc-compute* tuned profile, which is optimized for HPC workloads:

  Bash

  ```
  sudo dnf install -y tuned
  sudo systemctl enable --now tuned
  ```

```
sudo tuned-adm profile hpc-compute
```

- Enable Transparent Huge Pages (THP) to improve memory efficiency for large allocations:

```Bash
echo madvise | sudo tee /sys/kernel/mm/transparent_hugepage/enabled
echo madvise | sudo tee /sys/kernel/mm/transparent_hugepage/defrag
```

- Enable NUMA balancing for better memory locality:

```Bash
echo 1 | sudo tee /proc/sys/kernel/numa_balancing
```

- Drop caches before running jobs as this reduces variability and improves consistency:

```Bash
echo 3 | sudo tee /proc/sys/vm/drop_caches
```

- Use HPCX MPI library on Azure HPC-optimized images:

```Bash
module load mpi/hpcx
```

- MPI process placement and core affinity
  - Pin MPI processes to physical cores
  - Avoid oversubscription
  - Use symmetric rank distribution per CCD (1–7 ranks per CCD)
    - Each HBv5 VM has **48 CCDs**, so supported configurations include:
      `48, 96, 144, 192, 288, 336 ranks per VM`
  - Example command:

```Bash
mpirun -np <number_of_ranks> \
  --map-by ppr:<ranks_per_CCD>:l3cache \
  --rank-by slot \
  --bind-to core \
  <other_options> <executable> <arguments>
```

Optimal configuration depends on workload. Symmetric rank distribution usually performs best, but some workloads can benefit from using all 368 cores per VM. Benchmark multiple

configurations to determine the best setting.

(Topology reference: 16 NUMA regions, 48 CCDs per VM.)



- For multi VM jobs at scale, disable multi rail in UCX, using:

```bash
export UCX_MAX_RNDV_RAILS=1
```

Temporary storage

HBv5 VMs feature 9 physically local NVMe SSD devices. One device is preformatted to serve as a page file and appears in your VM as a generic *SSD* device. 8 other, larger SSDs are provided as unformatted block NVMe devices.

As the block NVMe device bypasses the hypervisor, it has higher bandwidth, higher IOPS, and lower latency per IOP. When paired in a striped array, the NVMe SSD is expected to provide up to 50 GB/s of read bandwidth and 30 GB/s of write bandwidth for large block sizes.

Combined, the 8 NVMe devices provide 15 TiB of total local storage per VM.

Hardware specifications

Expand table

Hardware specifications	HBv5-series VMs
Cores	368, 336, 288, 240, 192, 144, 96, 48 (SMT disabled)
CPU	AMD EPYC 9004-series (EPYC 9V64H)
CPU Frequency (non-AVX)	3.5 GHz base, 4 GHz peak boost(FMAX)
Memory	432 GB (RAM per core depends on VM size)
Local Disk	8 * 1.8 TB NVMe (block), 480 GB SSD (page file)
InfiniBand	4 * 200 Gb/s NVIDIA ConnectX-7 NDR InfiniBand
Network	180 Gb/s Azure Accelerated Networking

Software specifications

Expand table

Software specifications	HBv5-series VMs
Max MPI Job Size	HPC-X (2.18 or higher)*, OpenMPI (4.1.3 or higher), MVAPICH2 (2.3.7 or higher), MPICH (4.1 or higher)
MPI Support	HPC-X (2.13 or higher), Intel MPI (2021.7.0 or higher), OpenMPI (4.1.3 or higher), MVAPICH2 (2.3.7 or higher), MPICH (4.1 or higher)
Additional Frameworks	UCX, libfabric, PGAS, or other InfiniBand based runtimes
Azure Storage Support	Standard and Premium Disks (maximum 32 disks), Azure NetApp Files, Azure Files, Azure HPC Cache, Azure Managed Lustre File System (Preview)
Supported and Validated OS	AlmaLinux 8.10, Red Hat Enterprise Linux 8.10, Ubuntu 22.04+ and 24.04
Recommended OS for Performance	AlmaLinux HPC 8.10 (recommended image URN: almalinux:almalinux-hpc:8_10-hpc-gen2:latest), for scaling test, uses the URN recommended almalinux:almalinux-hpc:8_6-hpc-gen2:latest and the new HPC-X tarball, Ubuntu-HPC 18.04+
Orchestrator	Azure CycleCloud, AKS; cluster configuration options

Software specifications	HBv5-series VMs
Support	

> ⓘ Note
>
> - These VMs support only Generation 2 VMs.
> - All Red Hat Enterprise Linux (RHEL) versions earlier than 8.10, including derivatives such as CentOS and AlmaLinux, are deprecated.
> - Windows Server isn't supported on HBv5.

Known Issues with IB RDMA and NUMA Node Affinity

Issue Overview

On certain virtual machines (VMs), the InfiniBand RDMA device names (such as mlx5_[0-3]) may not align correctly with their respective NUMA node affinities. Ideally, each RDMA device should be mapped as follows:

- mlx5_0 is on NUMA node: 0
- mlx5_1 is on NUMA node: 4
- mlx5_2 is on NUMA node: 8
- mlx5_3 is on NUMA node: 12

However, an incorrect mapping example could be:

- mlx5_0 is on NUMA node: 4
- mlx5_1 is on NUMA node: 8
- mlx5_2 is on NUMA node: 12
- mlx5_3 is on NUMA node: 0

This misalignment can lead to performance degradation, particularly when running multimode MPI workloads.

Verifying RDMA device to NUMA node mapping

To confirm whether your RDMA devices are correctly mapped to NUMA nodes, execute the following script:

Bash

```
for d in /sys/class/infiniband/*;
do
dev=$(basename "$d")
node=$(cat "$d/device/numa_node")
echo "$dev is on NUMA node: $node"
done
```

Compare the output with the ideal mapping listed above.

Solution: Persistent device naming with Udev rules

To remediate the misalignment issue, follow these steps:

1. Create a new file in /etc/udev/rules.d/, for example: 99-rdma-persistent-naming.rules
2. Add the following lines to the file:

   Bash

   ```
   ACTION=="add", SUBSYSTEMS=="pci", KERNELS=="0101:00:00.0",
   PROGRAM="rdma_rename %k NAME_FIXED mlx5_ib0"
   ACTION=="add", SUBSYSTEMS=="pci", KERNELS=="0102:00:00.0",
   PROGRAM="rdma_rename %k NAME_FIXED mlx5_ib1"
   ACTION=="add", SUBSYSTEMS=="pci", KERNELS=="0103:00:00.0",
   PROGRAM="rdma_rename %k NAME_FIXED mlx5_ib2"
   ACTION=="add", SUBSYSTEMS=="pci", KERNELS=="0104:00:00.0",
   PROGRAM="rdma_rename %k NAME_FIXED mlx5_ib3"
   ```

3. Reload udev rules and trigger device events:

   Bash

   ```
   # udevadm control --reload
   # udevadm trigger --type=devices --action=add
   ```

This solution ensures that RDMA device naming persists across VM reboots.

Next steps

- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/10/2026

HBv5 sizes series

HBv5-series VMs are optimized for the most memory bandwidth-intensive HPC applications, including:

- Computational fluid dynamics
- Automotive and aerospace simulations
- Weather modeling
- Energy research
- Molecular dynamics simulations
- Computer-aided engineering and other HPC workloads

HBv5 VMs feature 6.7 TB/s of memory bandwidth across 432 GB of high-bandwidth memory (HBM) and up to 368 4th Generation AMD EPYC™ processor cores with 4 GHz boost frequencies, 3.5 GHz base frequencies, and no simultaneous multithreading.

Each HBv5-series VM also includes 14.3 TiB of local NVMe SSD storage with up to 50 GB/s (reads) and 30 GB/s (writes) of block device performance.

All HBv5-series VMs feature 800 Gb/s per node (4 x 200 Gb/s CX-7 NIC) of InfiniBand connectivity from NVIDIA Networking to enable supercomputer-scale MPI workloads. These VMs are connected in a nonblocking fat tree for optimized and consistent RDMA performance.

The InfiniBand NICs support features like Adaptive Routing, Dynamically Connected Transport (DCT), hardware acceleration of MPI collectives, and congestion control. These features enhance application performance, scalability, and consistency, and their usage is recommended for optimal performance and cost efficiency.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	48 - 368 vCPUs	4th Generation AMD EPYC [x86-64]
L3 Cache	1.5 GB	
Memory	432 GB	6.7 TB/s

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Local Storage	1 Temp Disk 8 NVMe Disks	480 GiB 14.304 TiB (1.9 TB/Disk)
Remote Storage	32 Disks	
Network	8 vNICs 4 InfiniBand NICs	180 Gb/s 800 Gb/s (4 x 200 Gb/s)
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (GB)	Memory Bandwidth (TB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HB368rs_v5	368	432	1.5	6.7	3.5	4	4
Standard_HB368-336rs_v5	336	432	1.5	6.7	3.5	4	4
Standard_HB368-288rs_v5	288	432	1.5	6.7	3.5	4	4
Standard_HB368-240rs_v5	240	432	1.5	6.7	3.5	4	4
Standard_HB368-192rs_v5	192	432	1.5	6.7	3.5	4	4
Standard_HB368-144rs_v5	144	432	1.5	6.7	3.5	4	4
Standard_HB368-96rs_v5	96	432	1.5	6.7	3.5	4	4

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (GB)	Memory Bandwidth (TB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HB368-48rs_v5	48	432	1.5	6.7	3.5	4	4

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Backend Network	InfiniBand NDR

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

HBv5-series virtual machine performance

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Performance expectations using common HPC microbenchmarks are as follows:

Expand table

Workload	HBv5
STREAM Triad	≈ 6.6 TB/s
High-Performance Linpack (HPL)	Up to 16 TF (Rpeak, FP64) for 368-core VM size
RDMA latency & bandwidth	< 2 microseconds (1 byte), 800 Gb/s (one-way)
FIO on local NVMe SSDs (RAID0)	50 GB/s reads, 30 GB/s writes

Process pinning

Process pinning works well on HBv5-series VMs because we expose the underlying silicon as-is to the guest VM. We strongly recommend process pinning for optimal performance and consistency.

STREAM Memory bandwidth test

The STREAM memory test can be run using the scripts in this GitHub repository.

Bash

```
git clone https://github.com/Azure/woc-benchmarking
cd woc-benchmarking/apps/hpc/stream/
sh build_stream.sh
sh stream_run_script.sh $PWD “hbrs_v5”
```

Compute performance test

The HPL benchmark can be run using the script in this GitHub repository.

Bash

```
git clone https://github.com/Azure/woc-benchmarking
cd woc-benchmarking/apps/hpc/hpl
sh hpl_build_script.sh
sh hpl_run_scr_HBv5.sh $PWD
```

MPI latency

The MPI latency test from the OSU microbenchmark suite can be executed as shown. Sample scripts are on GitHub.

Bash

```
module load mpi/hpcx
mpirun -np 2 --host $src,$dst --map-by node -x LD_LIBRARY_PATH
$HPCX_OSU_DIR/osu_latency
```

> **Note**
>
> Define source(src) and destination(dst).

MPI bandwidth

The MPI bandwidth test from the OSU microbenchmark suite can be executed per below. Sample scripts are on GitHub.

Bash

```
module load mpi/hpcx
mpirun -np 2 --host $src,$dst --map-by node -x LD_LIBRARY_PATH $HPCX_OSU_DIR/osu_bw
```

Next steps

- Learn about scaling MPI applications.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure HPC Microsoft Community Hub.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/10/2026

'HC' sub-family storage optimized VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'HC' family of VM size series are one of Azure's high-performance computing (HPC) optimized VM instances. These VMs are designed for compute-intensive workloads that need substantial CPU power. Examples include genomic sequencing, engineering simulations, and financial modeling. HB-series VMs use high-performance Intel Xeon Scalable processors and fast memory. HC-series VMs deliver exceptional computational performance and memory bandwidth, for applications that demand intense processing power to handle complex calculations and massive data sets efficiently. They're well-suited for industries such as healthcare, finance, and engineering, where rapid data processing and simulation accuracy are critical for advanced research and development.

Workloads and use cases

Genomic Sequencing: HC-series VMs provide the computational power needed for genomic sequencing, enabling researchers to process and analyze large genetic datasets quickly.

Engineering Simulations: Ideal for running complex simulations in fields such as automotive, aerospace, and mechanical engineering. These simulations often include finite element analysis (FEA) and computational fluid dynamics (CFD).

Financial Modeling: These VMs can handle the high demands of financial applications, including risk analysis and quantitative simulations, which require massive computational resources to execute many calculations quickly.

Scientific Research: HC-series VMs support a wide range of scientific computing needs, particularly in physics and chemistry, where large-scale computations and data analysis are crucial.

Weather Forecasting and Climate Simulation: They are used in meteorology for high-resolution weather modeling and climate simulations, which require processing large datasets and performing complex simulations.

Series in family

HC-series

HC-series Virtual Machine (VM)s are built for heavy computing tasks. They handle jobs like analyzing complex structures, studying how molecules behave, and running advanced chemistry simulations. HC VMs feature 44 Intel Xeon Platinum 8168 processor cores, 8 GB of RAM per CPU core, and no hyperthreading. Intel Xeon Platinum processors work with Intel's software tools, including the Intel Math Kernel Library. They also support advanced vector processing features like AVX-512 for faster calculations.

HC-series VMs feature 100 Gb/sec Mellanox EDR InfiniBand. These VMs are connected in a nonblocking fat tree for optimized and consistent RDMA performance. These VMs also supports Adaptive Routing and Dynamic Connected Transport (DCT), along with standard Reliable Connection (RC) and Unreliable Datagram (UD) transports. These features enhance application performance, scalability, and consistency, and their usage is recommended.

View the full HC-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	32 - 44 vCPUs	Intel Xeon Platinum 8168 (Skylake) [x86-64]
L3 Cache	60.5 MB	
Memory	352 GB	190 GB/s
Local Storage	1 Disk	700 GiB
Remote Storage	4 Disks	
Network	8 vNICs 1 InfiniBand EDR NIC	40 Gb/s 100 Gb/s
Accelerators	None	

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 11/25/2025

HC-series virtual machine overview

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Maximizing HPC application performance on Intel Xeon Scalable Processors requires a thoughtful approach to process placement on this new architecture. Here, we outline our implementation of it on Azure HC-series VMs for HPC applications. We'll use the term “pNUMA” to refer to a physical NUMA domain, and “vNUMA” to refer to a virtualized NUMA domain. Similarly, we'll use the term “pCore” to refer to physical CPU cores, and “vCore” to refer to virtualized CPU cores.

Physically, an HC-series server is 2 * 24-core Intel Xeon Platinum 8168 CPUs for a total of 48 physical cores. Each CPU is a single pNUMA domain, and has unified access to six channels of DRAM. Intel Xeon Platinum CPUs feature a 4x larger L2 cache than in prior generations (256 KB/core -> 1 MB/core), while also reducing the L3 cache compared to prior Intel CPUs (2.5 MB/core -> 1.375 MB/core).

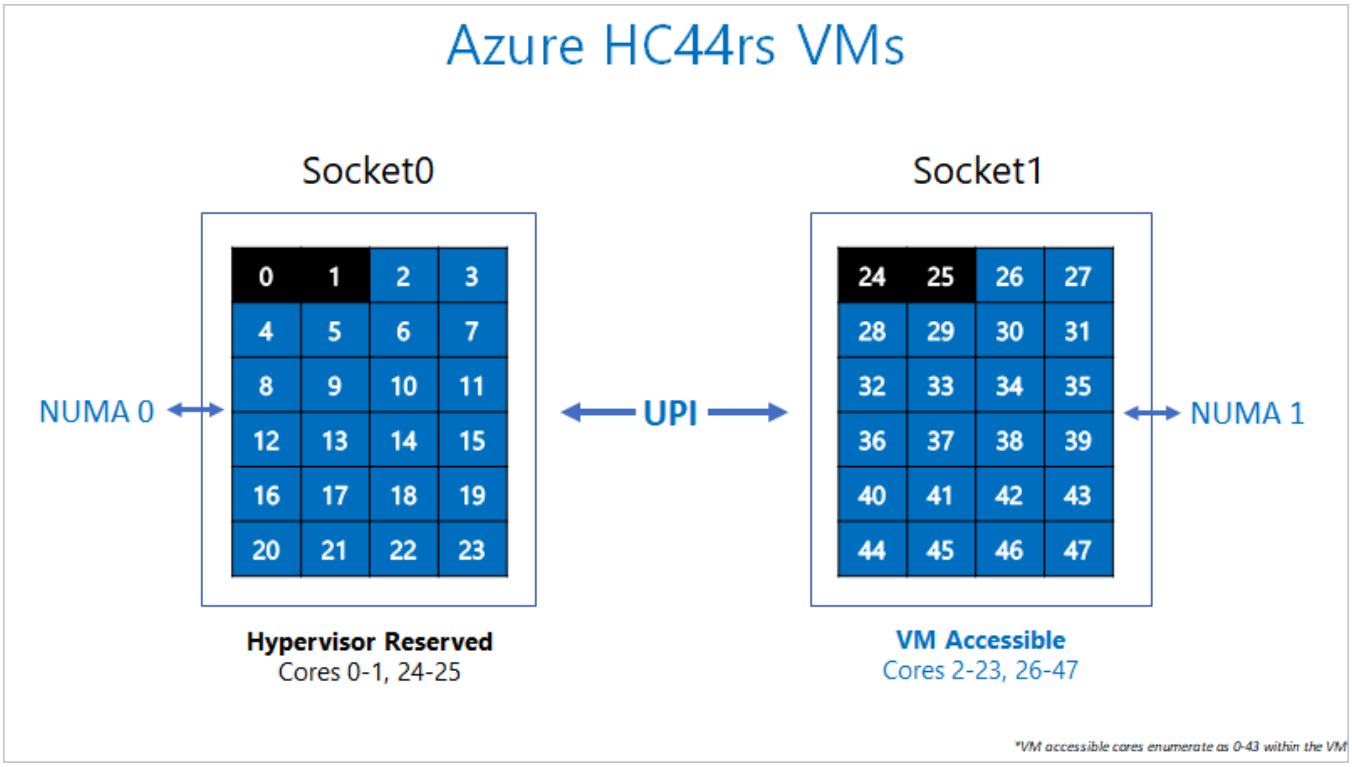
The above topology carries over to the HC-series hypervisor configuration as well. To provide room for the Azure hypervisor to operate without interfering with the VM, we reserve pCores 0-1 and 24-25 (that is, the first 2 pCores on each socket). We then assign pNUMA domains all remaining cores to the VM. Thus, the VM will see:

`(2 vNUMA domains) * (22 cores/vNUMA) = 44` cores per VM

The VM has no knowledge that pCores 0-1 and 24-25 weren't given to it. Thus, it exposes each vNUMA as if it natively had 22 cores.

Intel Xeon Platinum, Gold, and Silver CPUs also introduce an on-die 2D mesh network for communication within and external to the CPU socket. We strongly recommend process pinning for optimal performance and consistency. Process pinning will work on HC-series VMs because the underlying silicon is exposed as-is to the guest VM.

The following diagram shows the segregation of cores reserved for Azure Hypervisor and the HC-series VM.



Hardware specifications

Expand table

Hardware Specifications	HC-series VM
Cores	44 (HT disabled)
CPU	Intel Xeon Platinum 8168
CPU Frequency (non-AVX)	3.7 GHz (single core), 2.7-3.4 GHz (all cores)
Memory	8 GB/core (352 total)
Local Disk	700 GB SSD
Infiniband	100 Gb EDR Mellanox ConnectX-5
Network	50 Gb Ethernet (40 Gb usable) Azure second Gen SmartNIC

Software specifications

Expand table

Software Specifications	HC-series VM
Max MPI Job Size	13200 cores (300 VMs in a single virtual machine scale set with

Software Specifications	HC-series VM
	singlePlacementGroup=true)
MPI Support	HPC-X, Intel MPI, OpenMPI, MVAPICH2, MPICH, Platform MPI
Additional Frameworks	UCX, libfabric, PGAS
Azure Storage Support	Standard and Premium Disks (maximum 4 disks)
OS Support for SRIOV RDMA	RHEL 7.6+, Ubuntu 20.04+, SLES 15.4, WinServer 2016+
Orchestrator Support	CycleCloud, Batch, AKS; cluster configuration options

> **Note**
>
> Official kernel-level support from AMD starts with RHEL 8.6 and AlmaLinux 8.6, which is a derivative of RHEL.

Next steps

- Learn more about Intel Xeon SP architecture.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/10/2026

HC sizes series

> **Important**
>
> Azure Virtual Machines HC-series sizes (Standard_HC44rs, Standard_HC44-16rs, Standard_HC44-32rs) will be retired on May 31, 2027. After this date, any remaining HC-series VMs will be deallocated, stop running, and will no longer incur charges. HC-series will no longer have SLA or support following retirement. Sales of 1-year and 3-year Reserved Instances for HC-series ended on April 2, 2026. For more information, see Migrate your HC-series virtual machines by May 31, 2027.

HC-series Virtual Machine (VM)s are built for heavy computing tasks. They handle jobs like analyzing complex structures, studying how molecules behave, and running advanced chemistry simulations. HC VMs feature 44 Intel Xeon Platinum 8168 processor cores, 8 GB of RAM per CPU core, and no hyperthreading. Intel Xeon Platinum processors work with Intel's software tools, including the Intel Math Kernel Library. They also support advanced vector processing features like AVX-512 for faster calculations.

HC-series VMs feature 100 Gb/sec Mellanox EDR InfiniBand. These VMs are connected in a nonblocking fat tree for optimized and consistent RDMA performance. These VMs also supports Adaptive Routing and Dynamic Connected Transport (DCT), along with standard Reliable Connection (RC) and Unreliable Datagram (UD) transports. These features enhance application performance, scalability, and consistency, and their usage is recommended.

Migration and alternatives

To avoid service disruption, plan to migrate your HC-series workloads before May 31, 2027. For current-generation HPC alternatives, consider:

- **HBv5-series** – Offers higher compute performance and better price-performance than HBv4, suitable for most memory-bandwidth intensive HPC workloads.
- **HX-series** – Optimized for high-memory HPC workloads with approximately twice the memory capacity of HBv4, ideal for large-scale EDA and molecular dynamics.

For migration steps, see Change the size of a virtual machine.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	32 - 44 vCPUs	Intel Xeon Platinum 8168 (Skylake) [x86-64]
L3 Cache	60.5 MB	
Memory	352 GB	190 GB/s
Local Storage	1 Disk	700 GiB
Remote Storage	4 Disks	
Network	8 vNICs 1 InfiniBand EDR NIC	40 Gb/s 100 Gb/s
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (MB)	Memory Bandwidth (GB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HC44rs	44	352	60.5	191	2.7	3.7	3.4
Standard_HC44-16rs	16	352	60.5	191	2.7	3.7	3.4
Standard_HC44-32rs	32	352	60.5	191	2.7	3.7	3.4

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Backend Network	InfiniBand EDR

Availability and purchasing

Important

Sales of 1-year and 3-year Reserved Instances for HC-series are no longer available as of April 2, 2026. Pay-as-you-go remains available until the retirement date of May 31, 2027. After May 31, 2027, service-level agreements and support for HC-series end, and all remaining HC-series VMs will be automatically deallocated.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Next steps for HC-series customers

> **⊗ Caution**
>
> Plan to migrate your HC-series workloads before May 31, 2027. After this date, remaining HC-series VMs will be automatically deallocated. Evaluate these current-generation alternatives for your HPC workloads:
>
> - **HBv5-series** – Higher compute performance and better price-performance than HBv4
> - **HX-series** – High-memory HPC workloads with approximately twice the memory capacity of HBv4
>
> For migration steps, see **Change the size of a virtual machine** and **Migrate your HC-series virtual machines by May 31, 2027**.

Last updated on 04/17/2026

HC-series virtual machine sizes

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Performance testing was conducted across multiple HC-series sizes. The following table summarizes key findings from these tests.

Expand table

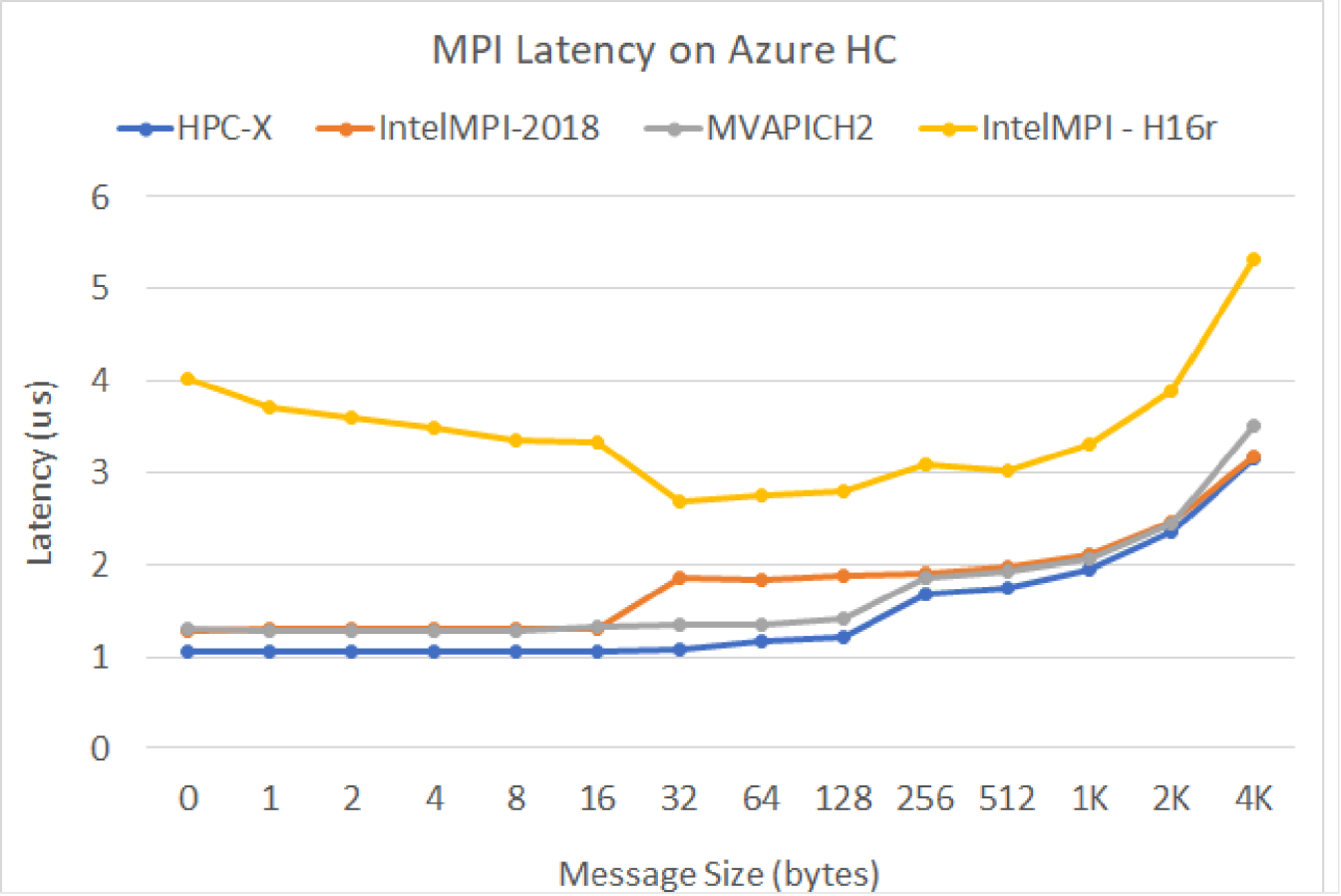
Workload	HC
STREAM Triad	190 GB/s (Intel MLC AVX-512)
High-Performance Linpack (HPL)	3520 GigaFLOPS (Rpeak), 2970 GigaFLOPS (Rmax)
RDMA latency & bandwidth	1.05 microseconds, 96.8 Gb/s
FIO on local NVMe SSD	1.3 GB/s reads, 900 MB/s writes
IOR on 4 Azure Premium SSD (P30 Managed Disks, RAID0)**	780 MB/s reads, 780 MB/writes

MPI latency

MPI latency test from the OSU Micro-Benchmarks (OMB) suite is run. Sample scripts are on GitHub

Bash

```
./bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=[INSERT CORE #]
./osu_latency
```

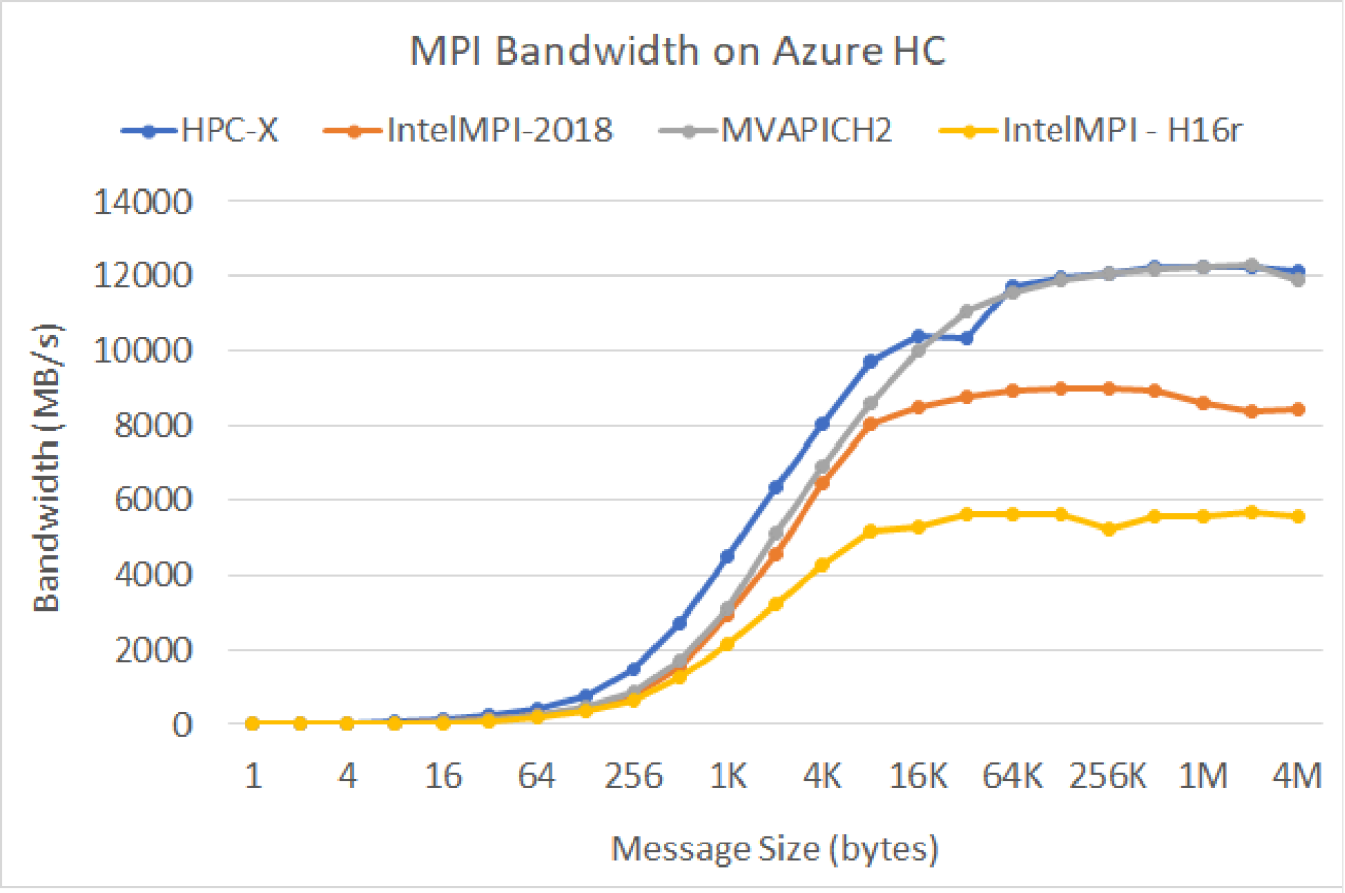


MPI bandwidth

MPI bandwidth test from the OSU microbenchmark suite is run. Sample scripts are on GitHub

Bash

```
./mvapich2-2.3.install/bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=
[INSERT CORE #] ./mvapich2-2.3/osu_benchmarks/mpi/pt2pt/osu_bw
```



Mellanox Perftest

The Mellanox Perftest package has many InfiniBand tests such as latency (ib_send_lat) and bandwidth (ib_send_bw). An example command is below.

Console

```
numactl --physcpubind=[INSERT CORE #]  ib_send_lat -a
```

Next steps

- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/11/2026

'HX' sub-family storage optimized VM size series

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The 'HX' family of VM size series are one of Azure's high-memory, high-performance computing (HPC) optimized VM instances. They're built for tasks that need a lot of memory and strong CPU power. Examples include in-memory databases, big data analysis, and advanced scientific simulations. HX-series VMs, equipped with expansive memory and powerful CPUs, delivers the resources to efficiently handle large datasets and perform rapid data processing. They work well for financial services, scientific research, and business planning. These areas need real-time handling of large amounts of data to stay efficient and drive new ideas.

Workloads and use cases

In-Memory Databases: HX-series VMs are excellent for hosting in-memory databases like SAP HANA, which require extensive memory to maintain large datasets in RAM for ultra-fast processing and access.

Big Data Analytics: They can handle big data analytics applications that need to process vast amounts of data in memory to speed up analysis, which is critical for real-time decision-making.

Genomic Research: Genomics research often involves large-scale data analysis, where high memory capacity can significantly enhance performance by allowing more of the dataset to be held in memory, speeding up the analysis.

Financial Simulations: Banks use HX-series VMs for fast trading and risk checks. These systems process huge amounts of data quickly to predict stock trends and figure out credit risks right away.

ERP Systems: HX-series VMs offer plenty of memory and strong processing power. Large business systems rely on these features to manage large amounts of data and keep many users working at once.

Series in family

HB-series

HX-series VMs are optimized for workloads that require significant memory capacity with twice the memory capacity as HBv4. For example, workloads such as silicon design can use HX-series VMs to enable EDA customers targeting the most advanced manufacturing processes to run their most memory-intensive workloads.

HX VMs feature up to 176 AMD EPYC™ 9V33X ("Genoa-X") CPU cores with AMD's 3D V-Cache, clock frequencies up to 3.7 GHz, and no simultaneous multithreading. HX-series VMs also provide 1408 GB of RAM, 2.3 GB L3 cache. The 2.3 GB L3 cache per VM can deliver up to 5.7 TB/s of bandwidth to amplify up to 780 GB/s of bandwidth from DRAM, for a blended average of 1.2 TB/s of effective memory bandwidth across a broad range of customer workloads. The VMs also provide up to 12 GB/s (reads) and 7 GB/s (writes) of block device SSD performance.

All HX-series VMs feature 400 Gb/s NDR InfiniBand from NVIDIA Networking to enable supercomputer-scale MPI workloads. These VMs are connected in a non-blocking fat tree for optimized and consistent RDMA performance. NDR continues to support features like Adaptive Routing and the Dynamically Connected Transport (DCT). This newest generation of InfiniBand also brings greater support for offload of MPI collectives, optimized real-world latencies due to congestion control intelligence, and enhanced adaptive routing capabilities. These features enhance application performance, scalability, and consistency, and their usage is recommended.

View the full HX-series page.

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	24 - 176 vCPUs	AMD EPYC 9V33X (Genoa-X) [x86-64]
L3 Cache	2304 MB	
Memory	1408 GB	780 GB/s
Local Storage	1 Temp Disk 2 NVMe Disks	480 GiB 1800 GiB
Remote Storage	32 Disks	
Network	8 vNICs 1 InfiniBand NDR NIC	80 Gb/s 400 Gb/s
Accelerators	None	

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 11/25/2025

HX-series virtual machine overview

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

An HX-series server features 2 * 96-core EPYC 9V33X CPUs for a total of 192 physical "Zen4" cores with AMD 3D-V Cache. Simultaneous Multithreading (SMT) is disabled on HX. These 192 cores are divided into 24 sections (12 per socket), each section containing 8 processor cores with uniform access to a 96 MB L3 cache. Azure HX servers also run the following AMD BIOS settings:

Bash

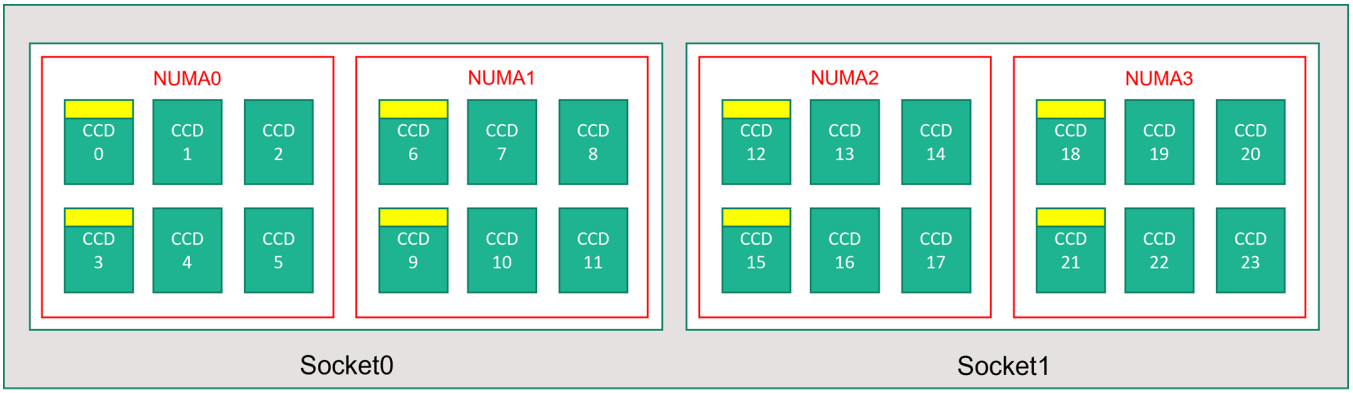
```
Nodes per Socket (NPS) = 2
L3 as NUMA = Disabled
NUMA domains within VM OS = 4
C-states = Enabled
```

As a result, the server boots with 4 NUMA domains (2 per socket) each 48-cores in size. Each NUMA has direct access to 6 channels of physical DRAM.

To provide room for the Azure hypervisor to operate without interfering with the VM, we reserve 16 physical cores per server.

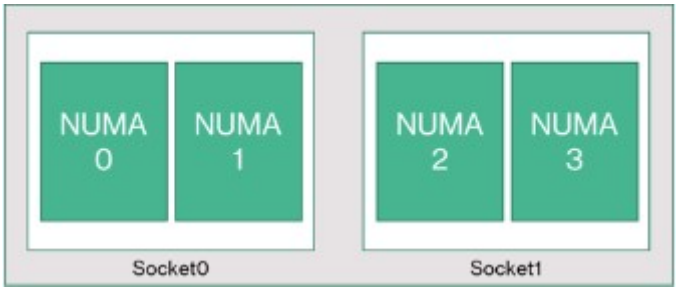
VM topology

The following diagram shows the topology of the server. We reserve these 16 hypervisor host cores (yellow) symmetrically across both CPU sockets, taking the first 2 cores from specific Core Complex Dies (CCDs) in each NUMA domain, with the remaining cores for the HX-series VM (green).



The CCD boundary is different from a NUMA boundary. On HX, a group of six (6) consecutive CCDs is configured as a NUMA domain, both at the host server level and within a guest VM.

Thus, all HX VM sizes expose 4 uniform NUMA domains that appear to an OS and application as shown each with different number of cores depending on the specific HX VM size.



Each HX VM size is similar in physical layout, features, and performance of a different CPU from the AMD EPYC 9004-series, as follows:

Expand table

HX-series VM size	NUMA domains	Cores per NUMA domain	Similarity with AMD EPYC
Standard_HX176rs	4	44	Dual-socket EPYC 9V33X
Standard_HX176-144rs	4	36	Dual-socket EPYC 9V33X
Standard_HX176-96rs	4	24	Dual-socket EPYC 9V33X
Standard_HX176-48rs	4	12	Dual-socket EPYC 9V33X
Standard_HX176-24rs	4	6	Dual-socket EPYC 9V33X

> **Note**
>
> - The constrained cores VM sizes only reduce the number of physical cores exposed to the VM. All global shared assets (RAM, memory bandwidth, L3 cache, GMI and xGMI connectivity, InfiniBand, Azure Ethernet network, local SSD) stay constant. It allows a customer to pick a VM size best tailored to a given set of workload or software licensing needs.
> - We are currently experiencing a core mapping issue on the HX-series parent size Standard_HX176rs.

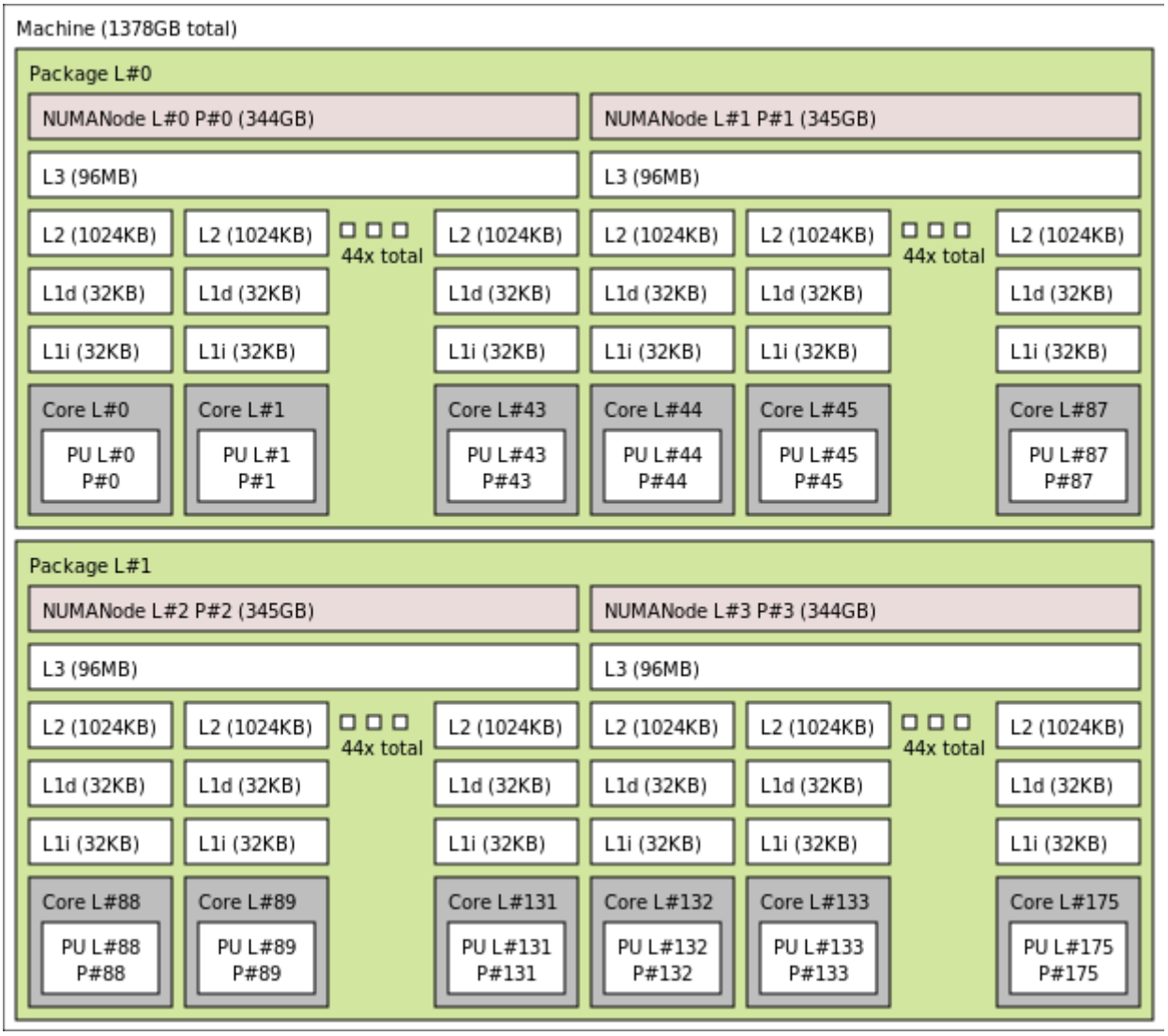
The virtual NUMA mapping of each HX VM size is mapped to the underlying physical NUMA topology. There is no potential misleading abstraction of the hardware topology.

The exact topology for the various HX VM size appears as follows using the output of lstopo:

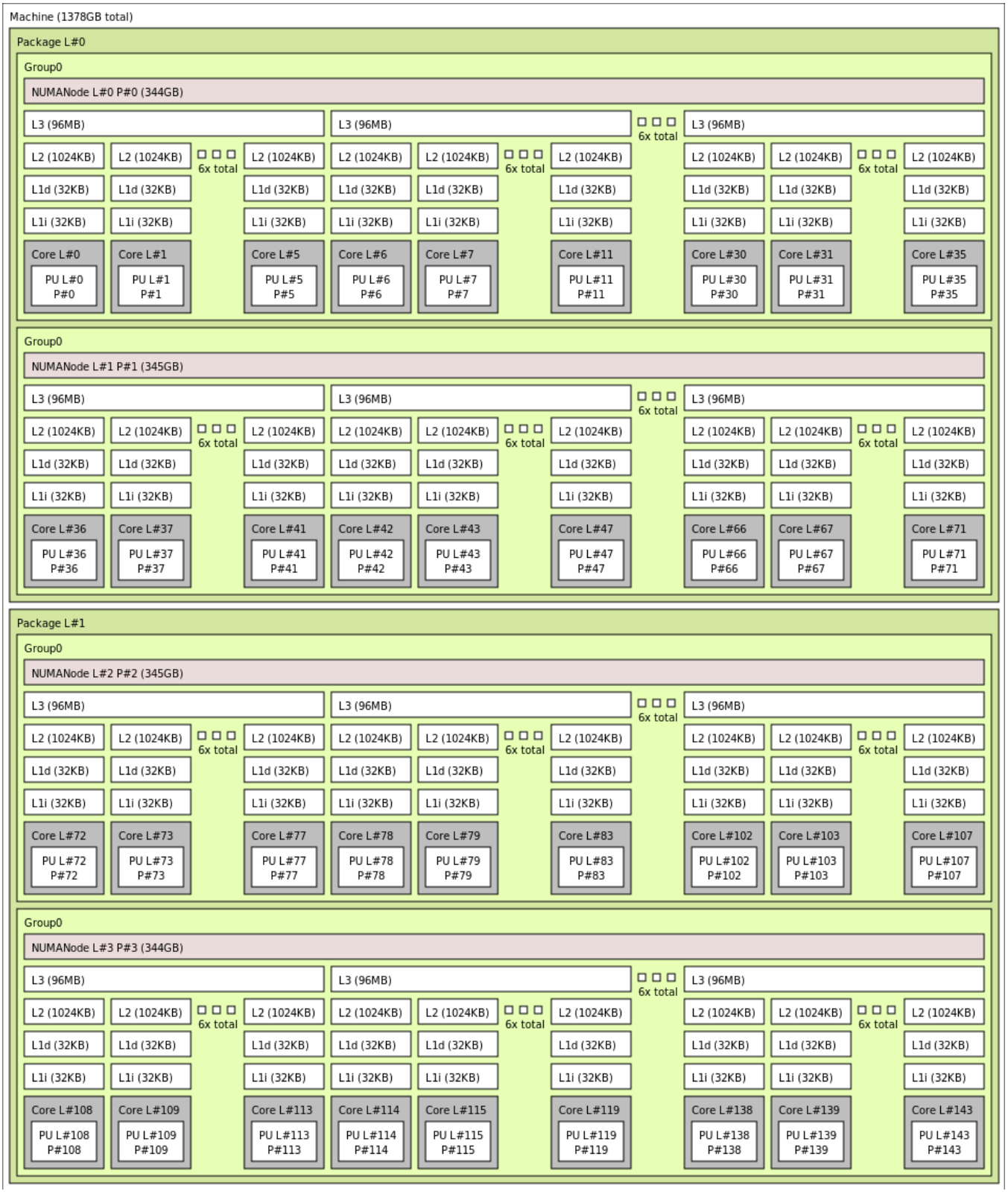
Bash

```
lstopo-no-graphics --no-io --no-legend --of txt
```

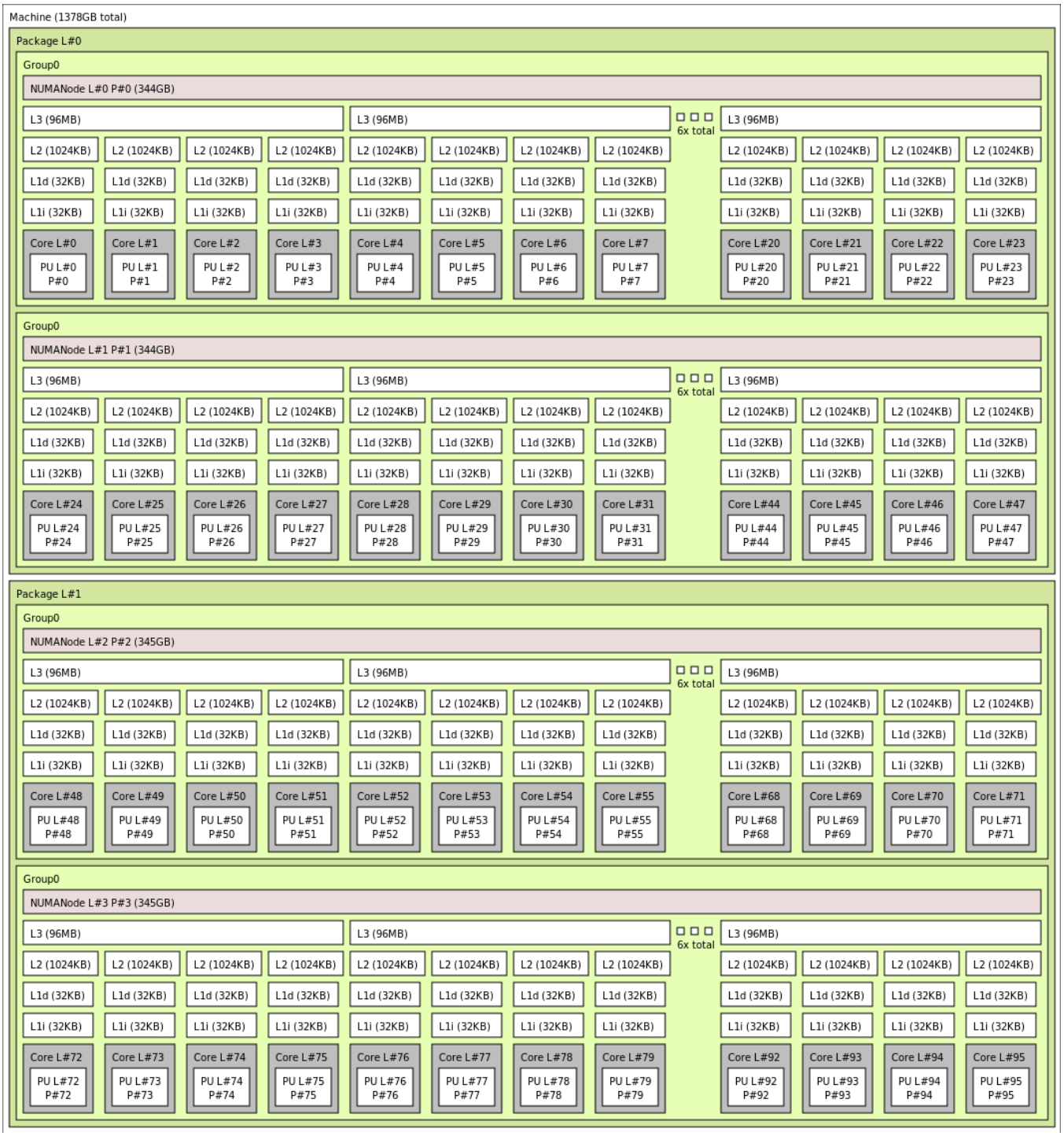
▼ Click to view lstopo output for Standard_HX176rs



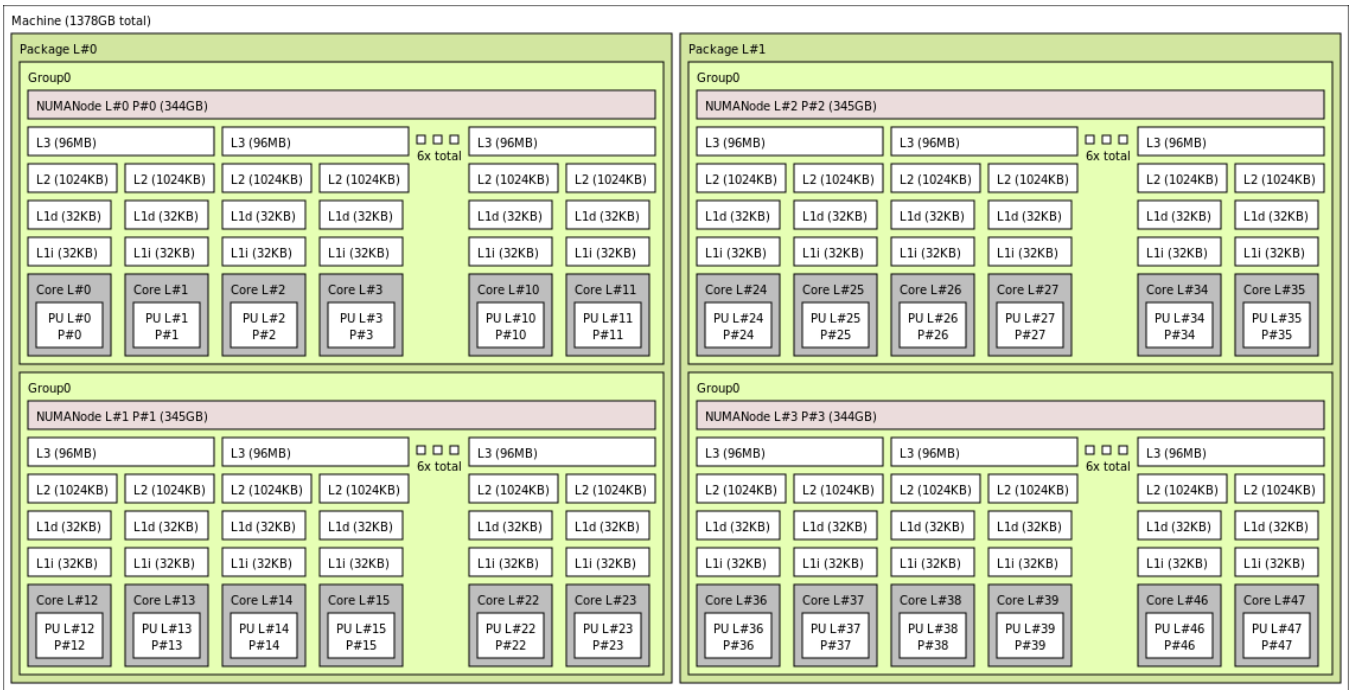
▼ Click to view lstopo output for Standard_HX176-144rs



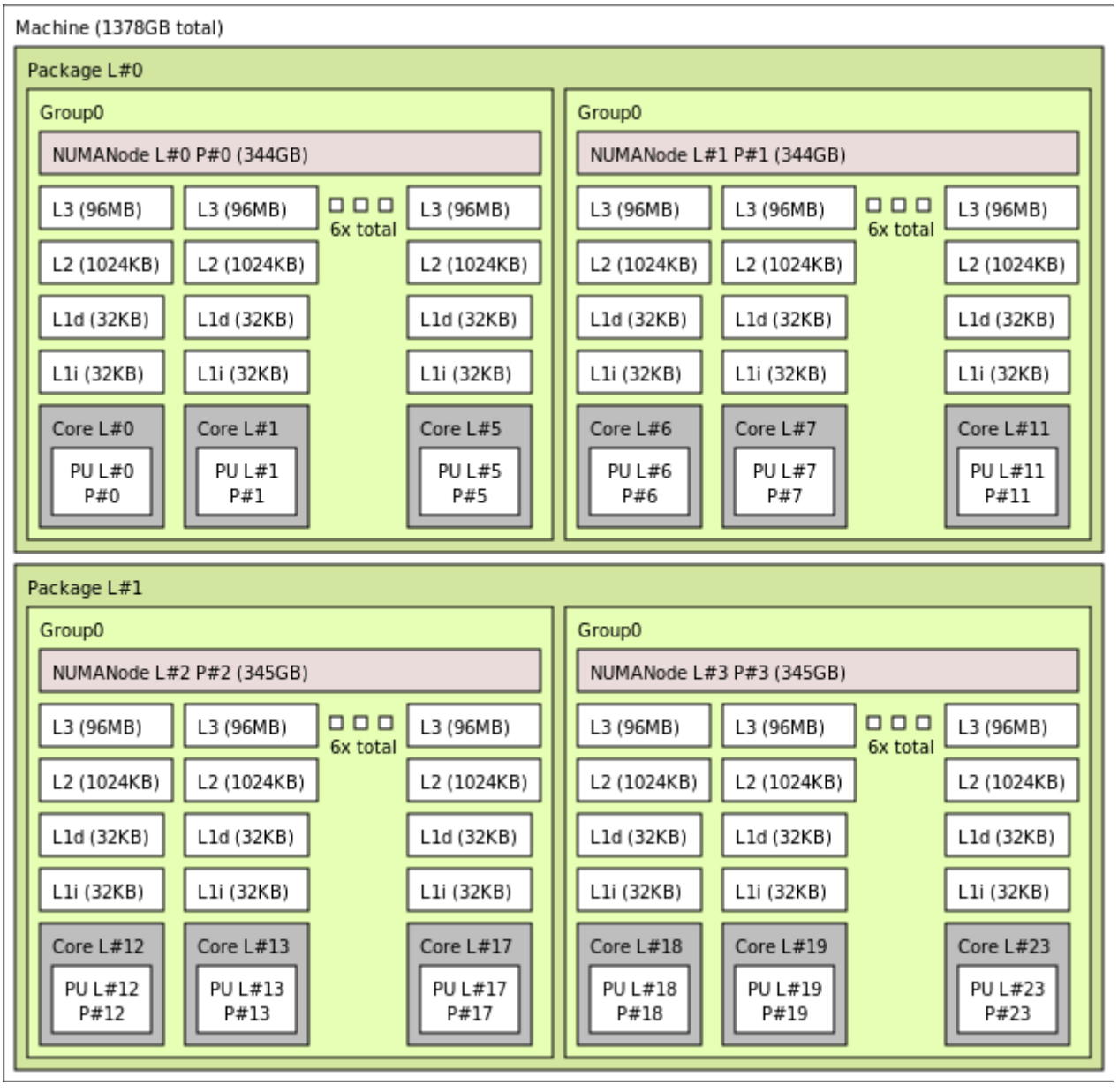
▼ Click to view lstopo output for Standard_HX176-96rs



▼ Click to view lstopo output for Standard_HX176-48rs



▼ Click to view lstopo output for Standard_HX176-24rs



InfiniBand networking

HX VMs also feature NVIDIA Mellanox NDR InfiniBand network adapters (ConnectX-7) operating at up to 400 Gigabits/sec. The NIC is passed through to the VM via SRIOV, enabling network traffic to bypass the hypervisor. As a result, customers load standard Mellanox OFED drivers on HX VMs as they would a bare metal environment.

HX VMs support Adaptive Routing, Dynamic Connected Transport (DCT, in addition to standard RC and UD transports), and hardware-based offload of MPI collectives to the onboard processor of the ConnectX-7 adapter. These features enhance application performance, scalability, and consistency, and usage of them is recommended.

Temporary storage

HX VMs feature 3 physically local SSD devices. One device is preformatted to serve as a page file and it appeared within your VM as a generic "SSD" device.

Two other, larger SSDs are provided as unformatted block NVMe devices via NVMeDirect. As the block NVMe device bypasses the hypervisor, it has higher bandwidth, higher IOPS, and lower latency per IOP.

When paired in a striped array, the NVMe SSD provides up to 12 GB/s reads and 7 GB/s writes, and up to 186,000 IOPS (reads) and 201,000 IOPS (writes) for deep queue depths.

Hardware specifications

Expand table

Hardware specifications	HX-series VMs
Cores	176, 144, 96, 48, or 24 (SMT disabled)
CPU	AMD EPYC 9V33X
CPU Frequency (non-AVX)	2.55 GHz base, 3.7 GHz peak boost
Memory	1.4 TB (RAM per core depends on VM size)
Local Disk	2 * 1.8 TB NVMe (block), 480 GB SSD (page file)
InfiniBand	400 Gb/s Mellanox ConnectX-7 NDR InfiniBand
Network	80 Gb/s Ethernet (40 Gb/s usable) Azure second Gen SmartNIC

Software specifications

Expand table

Software specifications	HX-series VMs
Max MPI Job Size	52,800 cores (300 VMs in a single virtual machine scale set with singlePlacementGroup=true)
MPI Support	HPC-X (2.13 or higher), Intel MPI (2021.7.0 or higher), OpenMPI (4.1.3 or higher), MVAPICH2 (2.3.7 or higher), MPICH (4.1 or higher)
Other Frameworks	UCX, libfabric, PGAS, or other InfiniBand based runtimes
Azure Storage Support	Standard and Premium Disks (maximum 32 disks), Azure NetApp Files, Azure Files, Azure HPC Cache, Azure Managed Lustre File System
Supported and Validated OS	AlmaLinux 8.6, 8.7, Ubuntu 20.04+
Recommended OS for Performance	AlmaLinux HPC 8.7, Ubuntu-HPC 20.04+
Orchestrator Support	Azure CycleCloud, Azure Batch, AKS; cluster configuration options

Note

- These VMs support only Generation 2.
- Official kernel-level support from AMD starts with RHEL 8.6 and AlmaLinux 8.6, which is a derivative of RHEL.
- Windows Server 2012 R2 is not supported on HX and other VMs with more than 64 (virtual or physical) cores. For more information, see Supported Windows guest operating systems for Hyper-V on Windows Server. Windows Server 2022 is required for 144 and 176 core sizes, Windows Server 2016 also works for 24, 48, and 96 core sizes, Windows Server works for only 24 and 48 core sizes.

Important

Recommended image URN: almalinux:almalinux-hpc:8_7-hpc-gen2:8.7.2023060101, To deploy this image over Azure CLI, ensure the following parameters are included **--plan 8_7-hpc-gen2 --product almalinux-hpc --publisher almalinux**. For scaling tests, use the recommended URN along with the new HPC-X tarball.

> **Note**
>
> - NDR support is added in UCX 1.13 or later. Older UCX versions report the runtime error. UCX Error: Invalid active speed `[1677010492.951559] [updsb-vm-0:2754 :0] ib_iface.c:1549 UCX ERROR Invalid active_speed on mlx5_ib0:1: 128`.
> - Ibstat shows low speed (SDR): Older Mellanox OFED (MOFED) versions do not support NDR and it may report slower IB speeds. Use MOFED versions MOFED 5.6-1.0.3.3 or greater.

Next steps

- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 02/10/2026

HX sizes series

Summarize this article for me

HX-series VMs are optimized for workloads that require significant memory capacity with twice the memory capacity as HBv4. For example, workloads such as silicon design can use HX-series VMs to enable EDA customers targeting the most advanced manufacturing processes to run their most memory-intensive workloads.

HX VMs feature up to 176 AMD EPYC™ 9V33X ("Genoa-X") CPU cores with AMD's 3D V-Cache, clock frequencies up to 3.7 GHz, and no simultaneous multithreading. HX-series VMs also provide 1408 GB of RAM, 2.3 GB L3 cache. The 2.3 GB L3 cache per VM can deliver up to 5.7 TB/s of bandwidth to amplify up to 780 GB/s of bandwidth from DRAM, for a blended average of 1.2 TB/s of effective memory bandwidth across a broad range of customer workloads. The VMs also provide up to 12 GB/s (reads) and 7 GB/s (writes) of block device SSD performance.

All HX-series VMs feature 400 Gb/s NDR InfiniBand from NVIDIA Networking to enable supercomputer-scale MPI workloads. These VMs are connected in a non-blocking fat tree for optimized and consistent RDMA performance. NDR continues to support features like Adaptive Routing and the Dynamically Connected Transport (DCT). This newest generation of InfiniBand also brings greater support for offload of MPI collectives, optimized real-world latencies due to congestion control intelligence, and enhanced adaptive routing capabilities. These features enhance application performance, scalability, and consistency, and their usage is recommended.

Host specifications

Expand table

Part	**Quantity** **Count Units**	**Specs** **SKU ID, Performance Units, etc.**
Processor	24 - 176 vCPUs	AMD EPYC 9V33X (Genoa-X) [x86-64]
L3 Cache	2304 MB	
Memory	1408 GB	780 GB/s
Local Storage	1 Temp Disk 2 NVMe Disks	480 GiB 1800 GiB
Remote Storage	32 Disks	
Network	8 vNICs 1 InfiniBand NDR NIC	80 Gb/s 400 Gb/s

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)	L3 Cache (MB)	Memory Bandwidth (GB/s)	Base CPU Frequency (GHz)	Single-core Frequency Peak (GHz)	All-core Frequency Peak (GHz)
Standard_HX176rs	176	1408	2304	780	2.55	3.7	3.7
Standard_HX176-144rs	144	1408	2304	780	2.55	3.7	3.7
Standard_HX176-96rs	96	1408	2304	780	2.55	3.7	3.7
Standard_HX176-48rs	48	1408	2304	780	2.55	3.7	3.7
Standard_HX176-24rs	24	1408	2304	780	2.55	3.7	3.7

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Not Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported
Backend Network	InfiniBand NDR

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

HX-series virtual machine performance

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Performance expectations using common HPC microbenchmarks are as follows:

Expand table

Workload	HX
STREAM Triad	750-780GB/s of DDR5, up to 5.7 TB/s of 3D-V Cache bandwidth
High-Performance Linpack (HPL)	Up to 7.6 TF (Rpeak, FP64) for 144-core VM size
RDMA latency & bandwidth	< 2 microseconds (1 byte), 400 Gb/s (one-way)
FIO on local NVMe SSDs (RAID0)	12 GB/s reads, 7 GB/s writes; 186k IOPS reads, 201k IOPS writes

Memory bandwidth test

The STREAM memory test can be run using the scripts in this GitHub repository.

Bash

```
git clone https://github.com/Azure/woc-benchmarking
cd woc-benchmarking/apps/hpc/stream/
sh build_stream.sh
sh stream_run_script.sh $PWD “hbrs_v4”
```

Compute performance test

The HPL benchmark can be run using the script in this GitHub repository.

Bash

```
git clone https://github.com/Azure/woc-benchmarking
cd woc-benchmarking/apps/hpc/hpl
sh hpl_build_script.sh
sh hpl_run_scr_hbv4.sh $PWD
```

MPI latency

The MPI latency test from the OSU microbenchmark suite can be executed as shown. Sample scripts are on GitHub.

```Bash
module load mpi/hpcx
mpirun -np 2 --host $src,$dst --map-by node -x LD_LIBRARY_PATH
$HPCX_OSU_DIR/osu_latency
```

MPI bandwidth

The MPI bandwidth test from the OSU microbenchmark suite can be executed as shown. Sample scripts are on GitHub.

```Bash
module load mpi/hpcx
mpirun -np 2 --host $src,$dst --map-by node -x LD_LIBRARY_PATH
$HPCX_OSU_DIR/osu_bw
```

[!NOTE] Define source(src) and destination(dst).

Mellanox Perftest

The Mellanox Perftest package has many InfiniBand tests such as latency (ib_send_lat) and bandwidth (ib_send_bw). An example command is shown.

```Console
numactl --physcpubind=[INSERT CORE #]  ib_send_lat -a
```

[!NOTE] NUMA node affinity for InfiniBand NIC is NUMA0.

Next steps

- Learn about scaling MPI applications.
- Review the performance and scalability results of HPC applications on the HX VMs at the TechCommunity article.

- Read about the latest announcements, HPC workload examples, and performance results at the Azure HPC Microsoft Community Hub.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Set up Azure HPC or AI VMs

08/22/2024

This how-to guide explains how to create a basic Azure virtual machine (VM) for HPC and AI with NVIDIA or AMD GPUs. These VM sizes are intended for workloads that require high-performance computing (HPC sizes), or GPU-accelerated computing (AI sizes).

Choose your VM size

Azure VMs have many different options, called VM sizes. There are different series of VM sizes for HPC and VM sizes GPU-optimized computing. Select the appropriate VM size for the workload you want to use. For help with selecting sizes, see the VM selector tool.

Not all Azure products are available in all Azure regions. For more information, see the current list of products available by region.

Create your VM

Before you can deploy a workload, you need to create your VM through the Azure portal.

Depending on your VM's operating system, review either the Linux VM quickstart or Windows VM quickstart. Then, create your VM with the following settings:

1. For **Subscription**, select the Azure subscription that you want to use for this VM.
2. For **Region**, select a region with capacity available for your VM size.
3. For **Image**, select the image of the VM you chose in the previous section.

   > **Note**
   >
   > For the purpose of example, this guide uses the image **NVIDIA GPU-Optimized Image for AI & HPC – v21.04.1 – Gen 1**. If you're using another image, you might need to install other software, like the NVIDIA driver and Docker, before proceeding.

4. For **Size**, select the HPC or GPU instance type. For more information, see how to choose your VM size.
5. For **SSH public key source**, select **Generate a new key pair**.
6. Wait for key validation to complete.

7. When prompted, select **Download private key and create resource**.

   > ⓘ Note
   >
   > Downloading the key pair is necessary to SSH into your VM for later configuration.

8. For **Key pair name**, enter a name for your key pair.

9. Under the **Networking** tab, make sure **Accelerated Networking** is disabled.

10. Optionally, add a data disk to your VM. For more information, see how to add a data disk to a Linux VM or to a Windows VM.

    > ⓘ Note
    >
    > Adding a data disk helps you store models, data sets, and other necessary components for benchmarking.

11. Select **Review + create** to create your VM.

Connect to your VM

Connect to your new VM using SSH, which allows you to perform further configuration. Some connection methods include:

- Connect over SSH on Linux or macOS
- Connect over SSH on Windows
- Connect over SSH using Azure Bastion

Set up VM

Set up your new VM for HPC or AI workloads. Install the newest NVIDIA or AMD GPU driver, which maps to your VM size.

- Install NVIDIA GPU drivers on N-series VMs running Linux
- Install NVIDIA GPU drivers on N-series VMs running Windows
- Install AMD GPU drivers on N-series VMs running Windows

Next steps

- High performance computing VM sizes

- GPU optimized virtual machine sizes

Set up InfiniBand

Article • 11/21/2024

> This article references CentOS, a Linux distribution that is nearing End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

> **Tip**
>
> Try the **Virtual machines selector tool** to find other sizes that best fit your workload.

This article shares some information on RDMA-capable instances to be used over an InfiniBand (IB) network. The InfiniBand network architecture features a full fat-tree design(that is a network topology that provides high bandwidth and low latency), ensuring non-blocking, bisectional symmetry. This configuration provides equal bandwidth between any two virtual machines (VMs) within the same virtual machine scale set (VMSS).

RDMA-capable instances

Most of the HPC VM sizes feature a network interface for remote direct memory access (RDMA) connectivity. Selected N-series sizes designated with 'r' are also RDMA-capable. This interface is in addition to the standard Azure Ethernet network interface available in the other VM sizes.

This secondary interface allows the RDMA-capable instances to communicate over an InfiniBand network, operating at HDR rates for HBv4, HBv3, HBv2, EDR rates for HB, HC, HX, NDv2, and FDR rates for H16r, H16mr, and other RDMA-capable N-series virtual machines. These RDMA capabilities can boost the scalability and performance of Message Passing Interface (MPI) based applications.

> **Note**
>
> **SR-IOV support**: In Azure HPC, currently there are two classes of VMs depending on whether they are SR-IOV enabled for InfiniBand. Currently, almost all the newer

generation, RDMA-capable or InfiniBand enabled VMs on Azure are SR-IOV enabled except for H16r, H16mr, and NC24r. RDMA is only enabled over the InfiniBand network and is supported for all RDMA-capable VMs. IP over IB is only supported on the SR-IOV enabled VMs. RDMA is not enabled over the Ethernet network.

- **Operating System** - Linux distributions such as CentOS, RHEL, AlmaLinux, Ubuntu, SUSE are commonly used. Windows Server 2016 and newer versions are supported on all the HPC series VMs. Note that Windows Server 2012 R2 is not supported on HBv2 onwards as VM sizes with more than 64 (virtual or physical) cores. See VM Images for a list of supported Linux VM images on the Azure Marketplace and how they can be configured appropriately. The respective VM size pages also list out the software stack support.
- **InfiniBand and Drivers** - On InfiniBand enabled VMs, the appropriate drivers are required to enable RDMA. See enabling InfiniBand to learn about VM extensions or manual installation of InfiniBand drivers.
- **MPI** - The SR-IOV enabled VM sizes on Azure allow almost any flavor of MPI to be used with Mellanox OFED. See Setup MPI for HPC for more details on setting up MPI on HPC VMs on Azure.

  > ⓘ Note
  >
  > **RDMA network address space**: The RDMA network in Azure reserves the address space 172.16.0.0/16. To run MPI applications on instances deployed in an Azure virtual network, make sure that the virtual network address space does not overlap the RDMA network.

Cluster configuration options

Azure provides several options to create clusters of HPC VMs that can communicate using the RDMA network, including:

- **Virtual machines** - Deploy the RDMA-capable HPC VMs in the same scale set or availability set (when you use the Azure Resource Manager deployment model). If you use the classic deployment model, deploy the VMs in the same cloud service.
- **Virtual machine scale sets** - In a virtual machine scale set, ensure that you limit the deployment to a single placement group for InfiniBand communication within the

scale set. For example, in a Resource Manager template, set the `singlePlacementGroup` property to `true`.

Note that the maximum scale set size that can be spun up with `singlePlacementGroup=true` is capped at 100 VMs by default. If your HPC job scale needs are higher than 100 VMs in a single tenant, you may request an increase, open an online customer support request at no charge. The limit on the number of VMs in a single scale set can be increased to 300. Note that when deploying VMs using Availability Sets the maximum limit is at 200 VMs per Availability Set.

Also, VMSS serves as the isolation boundary between workloads within the same cluster, ensuring that instances in different VMSSs remain isolated from each other to guarantee security.

> ⓘ **Note**
>
> **MPI among virtual machines**: If RDMA (e.g. using MPI communication) is required between virtual machines, ensure that the VMs are in the same virtual machine scale set or availability set.

- **Azure CycleCloud** - Create an HPC cluster using Azure CycleCloud to run MPI jobs.
- **Azure Batch** - Create an Azure Batch pool to run MPI workloads. To use compute-intensive instances when running MPI applications with Azure Batch, see Use multi-instance tasks to run Message Passing Interface (MPI) applications in Azure Batch.
- **Microsoft HPC Pack** - HPC Pack includes a runtime environment for MS-MPI that uses the Azure RDMA network when deployed on RDMA-capable Linux VMs. For example deployments, see Set up a Linux RDMA cluster with HPC Pack to run MPI applications.

Deployment considerations

- **Azure subscription** – To deploy more than a few compute-intensive instances, consider a pay-as-you-go subscription or other purchase options. If you're using an Azure free account, you can use only a limited number of Azure compute cores.
- **Pricing and availability** - Check VM pricing and availability by Azure regions.

- **Cores quota** – You might need to increase the cores quota in your Azure subscription from the default value. Your subscription might also limit the number of cores you can deploy in certain VM size families, including the H-series. To request a quota increase, open an online customer support request at no charge. (Default limits may vary depending on your subscription category.)

  > **Note**
  >
  > Contact Azure Support if you have large-scale capacity needs. Azure quotas are credit limits, not capacity guarantees. Regardless of your quota, you are only charged for cores that you use.

- **Virtual network** – An Azure virtual network is not required to use the compute-intensive instances. However, for many deployments you need at least a cloud-based Azure virtual network, or a site-to-site connection if you need to access on-premises resources. When needed, create a new virtual network to deploy the instances. Adding compute-intensive VMs to a virtual network in an affinity group is not supported.

- **Resizing** – Because of their specialized hardware, you can only resize compute-intensive instances within the same size family (H-series or N-series). For example, you can only resize an H-series VM from one H-series size to another. Additional considerations around InfiniBand driver support and NVMe disks may need to be considered for certain VMs.

Next steps

- Learn more about configuring your VMs, enabling InfiniBand, setting up MPI and optimizing HPC applications for Azure at HPC Workloads.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Feedback

Was this page helpful? Yes No

Configure and optimize the InfiniBand enabled H-series and N-series VMs

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article shares some guidance on configuring and optimizing the InfiniBand-enabled HB-series and N-series VMs for HPC.

VM images

On InfiniBand (IB) enabled VMs, the appropriate IB drivers are required to enable RDMA.

- The Ubuntu-HPC VM images in the Marketplace come preconfigured with the appropriate NVIDIA IB drivers and GPU drivers.
- The AlmaLinux-HPC VM images in the Marketplace come preconfigured with the appropriate NVIDIA IB drivers and GPU drivers.

These VM images are based on the base Ubuntu and AlmaLinux marketplace VM images. Scripts used in the creation of these VM images from their base marketplace images are on the azhpc-images repo.

On GPU enabled N-series VMs, the appropriate GPU drivers are additionally required. This can be available by the following methods:

- Use the Ubuntu-HPC VM images or AlmaLinux-HPC VM images that come preconfigured with the NVIDIA GPU drivers and GPU compute software stack (CUDA, NCCL).
- Add the GPU drivers through the VM extensions.
- Install the GPU drivers manually.
- Some other VM images on the Marketplace also come preinstalled with the NVIDIA GPU drivers, including some VM images from NVIDIA.

Depending on the workloads' Linux distro and version needs, Ubuntu-HPC VM images and AlmaLinux-HPC VM images on the Marketplace are the easiest way to get started with HPC and AI workloads on Azure. It's also recommended to create custom VM images with workload specific customization and configuration for reuse.

VM sizes supported by the HPC VM images

InfiniBand OFED support

The latest Azure HPC marketplace images come with Mellanox OFED 5.1 and above, which do not support ConnectX3-Pro InfiniBand cards. ConnectX-3 Pro InfiniBand cards require MOFED 4.9 LTS version. These VM images only support ConnextX-5 and newer InfiniBand cards. The following VM size support matrix for the InfiniBand OFED in these HPC VM images:

- HB-series: HB, HC, HBv2, HBv3, HBv4
- N-series: NDv2, NDv4

GPU driver support

Currently only the Ubuntu-HPC VM images and AlmaLinux-HPC VM images come preconfigured with the NVIDIA GPU drivers and GPU compute software stack (CUDA, NCCL).

The VM size support matrix for the GPU drivers in supported HPC VM images is as follows:

- N-series: NDv2, NDv4 VM sizes are supported with the NVIDIA GPU drivers and GPU compute software stack (CUDA, NCCL).
- The other 'NC' and 'ND' VM sizes in the N-series are supported with the NVIDIA GPU drivers.

All of the VM sizes in the N-series support Gen 2 VMs, though some older ones also support Gen 1 VMs. Gen 2 support is also indicated with a "01" at the end of the image URN or version.

SR-IOV enabled VMs

Ubuntu-HPC VM images

For SR-IOV enabled RDMA capable VMs, Ubuntu-HPC VM images versions 18.04, 20.04, and 22.04 are suitable. These VM images come preconfigured with the Mellanox OFED drivers for RDMA, NVIDIA GPU drivers, GPU compute software stack (CUDA, NCCL), and commonly used MPI libraries and scientific computing packages. Refer to the VM size support matrix.

- The available or latest versions of the VM images can be listed with the following information using CLI or Marketplace.

  Output

  ```
  "publisher": "Microsoft-DSVM",
  "offer": "Ubuntu-HPC",
  ```

- Scripts used in the creation of the Ubuntu-HPC VM images from a base Ubuntu Marketplace image are on the azhpc-images repo.

AlmaLinux-HPC VM images

For SR-IOV enabled RDMA capable VMs, AlmaLinux-HPC VM images versions 8.5, 8.6, and 8.7 are suitable. These VM images come preconfigured with the Mellanox OFED drivers for RDMA, NVIDIA GPU drivers, GPU compute software stack (CUDA, NCCL), and commonly used MPI libraries and scientific computing packages. Refer to the VM size support matrix.

- The available or latest versions of the VM images can be listed with the following information using CLI or Marketplace.

  Output

  ```
  "publisher": "AlmaLinux",
  "offer": "AlmaLinux-HPC",
  ```

- Scripts used in the creation of the AlmaLinux-HPC VM images from a base AlmaLinux Marketplace image are on the azhpc-images repo.

Additionally, more details on what's included in the Ubuntu-HPC VM images and AlmaLinux-HPC VM images, and how to deploy them are in Azure HPC VM images.

RHEL VM images

The base RHEL-based non-HPC VM images on the Marketplace can be configured for use on the SR-IOV enabled RDMA capable VMs. Learn more about enabling InfiniBand and setting up MPI on the VMs.

Ubuntu VM images

The base Ubuntu Server 20.04 LTS and 22.04 LTS VM images in the Marketplace are supported for both SR-IOV and non-SR-IOV RDMA capable VMs. Learn more about enabling InfiniBand and setting up MPI on the VMs.

- Instructions for enabling InfiniBand on the Ubuntu VM images are in a TechCommunity article.

> **Note**
>
> Mellanox OFED 5.1 and above don't support ConnectX3-Pro InfiniBand cards on SR-IOV enabled N-series VM sizes with FDR InfiniBand (e.g. NCv3). Please use LTS Mellanox OFED version 4.9-0.1.7.0 or older on the N-series VM's with ConnectX3-Pro cards. For more information, see **Linux InfiniBand Drivers**.

SUSE Linux Enterprise Server VM images

SLES 12 SP3 for HPC, SLES 12 SP3 for HPC (Premium), SLES 12 SP1 for HPC, SLES 12 SP1 for HPC (Premium), SLES 12 SP4 and SLES 15 VM images in the Marketplace are supported. These VM images come preloaded with the Network Direct drivers for RDMA (on the non-SR-IOV VM sizes) and Intel MPI version 5.1. Learn more about setting up MPI on the VMs.

Optimize VMs

The following are some optional optimization settings for improved performance on the VM.

Update LIS

If necessary for functionality or performance, Linux Integration Services (LIS) drivers can be installed or updated on supported OS distros, especially is deploying using a custom image or an older OS version such as RHEL 6.x or earlier version of 7.x.

Bash

```
wget https://aka.ms/lis
tar xzf lis
pushd LISISO
sudo ./upgrade.sh
```

Reclaim memory

Improve performance by automatically reclaiming memory to avoid remote memory access.

Bash

```
sudo echo 1 >/proc/sys/vm/zone_reclaim_mode
```

Keep reclaim memory mode persistent after VM reboots:

Bash

```
sudo echo "vm.zone_reclaim_mode = 1" >> /etc/sysctl.conf sysctl -p
```

Disable firewall and SELinux

Bash

```
sudo systemctl stop iptables.service
sudo systemctl disable iptables.service
sudo systemctl mask firewalld
sudo systemctl stop firewalld.service
sudo systemctl disable firewalld.service
sudo iptables -nL
sudo sed -i -e's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

Disable cpupower

Bash

```
sudo service cpupower status
```

If enabled, disable it:

Bash

```
sudo service cpupower stop
sudo systemctl disable cpupower
```

Configure WALinuxAgent

Bash

```
sudo sed -i -e 's/# OS.EnableRDMA=y/OS.EnableRDMA=y/g' /etc/waagent.conf
```

Optionally, the WALinuxAgent may be disabled before running a job then enabled post-job for maximum VM resource availability to the HPC workload.

Next steps

- Learn more about enabling InfiniBand on the InfiniBand-enabled HB-series and N-series VMs.
- Learn more about installing and running various supported MPI libraries on the VMs.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 08/22/2024

Azure HPC/AI VM images

09/04/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article shares some information on HPC/AI VM images to be used to launch InfiniBand enabled H-series and GPU enabled N-series VMs.

The Azure HPC team is offering optimized and pre-configured Linux VM images for HPC and AI workloads. These VM images are:

- Based on upstream Ubuntu and AlmaLinux marketplace VM images.
- Pre-configured with NVIDIA Mellanox OFED driver for InfiniBand, NVIDIA GPU drivers, popular MPI libraries, vendor tuned HPC libraries, and recommended performance optimizations.
- Including optimizations and recommended configurations to deliver optimal performance, consistency, and reliability.

Availability on Azure

You may use the HPC/AI images when creating a VM from either Azure Marketplace or Azure CLI. For other deployment methods, refer to the section of Deploying HPC VM Images.

Azure Marketplace

Search for `Ubuntu HPC` by the publisher `Microsoft-DSVM`, or `AlmaLinux HPC` by the publisher `AlmaLinux`.

Azure CLI

Run the following commands to find image URNs of the HPC images:

Ubuntu-HPC

```
az vm image list --publisher microsoft-dsvm --offer ubuntu-hpc --output table --
all
```

All images support Gen 2 VMs.

AlmaLinux-HPC

```
az vm image list --publisher almalinux --offer almalinux-hpc --output table --all
```

All images support both Gen 1 and Gen 2 VMs.

Supported VM sizes

The HPC VM images support the following VM sizes:

- Standard_HB60rs
- Standard_HB120rs_v2
- Standard_HB120rs_v3
- Standard_HB120rs_v4
- Standard_HC44rs
- Standard_ND40rs_v2
- Standard_ND96asr_v4
- Standard_ND96amsr_A100_v4
- Standard_ND96isr_H100_v5
- Standard_ND96isr_H200_v5
- Standard_ND96isr_MI300X_v5

Refer to Azure VM sizes for the latest H- and N-series VM size support matrix.

Installed software packages

- Mellanox OFED 24.01-0.3.3.1
- Pre-configured IPoIB (IP-over-InfiniBand)
- Popular InfiniBand based MPI Libraries
  - HPC-X v2.18 with/without PMIx-4
  - Intel MPI 2021.12.0
  - MVAPICH2 2.3.7-1
  - OpenMPI 5.0.2 with PMIx-4
- Communication Runtimes
  - Libfabric
  - OpenUCX
  - NCCL 2.21.5-1
  - NCCL RDMA Sharp Plugin
- Optimized libraries

    - AMD Optimizing C/C++ and Fortran Compilers 4.0.0-1
    - Intel MKL 2024.0.0.49673
- GPU Drivers
    - NVIDIA GPU Driver 535.161.08
    - NVIDIA Peer Memory (GPU Direct RDMA)
    - NVIDIA Fabric Manager
    - CUDA 12.4
- GDRCopy 2.3
- Data Center GPU Manager 3.3.3
- Azure HPC Diagnostics Tool
- SKU based customizations
    - Topology files
    - NCCL configuration
- Moby 24.0.7-ubuntu22.04u1
- NVIDIA Docker container 24.0.7-1
- Azure Managed Lustre 2.15.4-42-gd6d405d
- Moneo v0.3.5
- Azure HPC Health Checks v0.4.2

An installed version index within the VM image is located at this location: `/opt/azurehpc/component_versions.txt`.

MPI libraries and software packages are available as environment modules. To load an MPI library/package, run:

```
module load mpi/<mpi-library-name>
module load <package-name>
```

Configuration and optimization

Refer to the azhpc-images repo at GitHub for the latest details on what packages and configuration is included in each VM image. The included configurations are based on optimization recommendations from vendors and partners, as well as learnings from common HPC workloads and usage practices in traditional HPC systems.

- Azure Linux Agent (WAAgent)
    - Limit waagent's (VM agent running on every Azure Linux VM) usage of CPU/memory resources.
    - Optionally, consider disabling waagent at the beginning of your job script, and enabling it back at the end, for CPU sensitive workloads as follows:

```
sudo systemctl stop waagent
<HPC job>
sudo systemctl restart waagent
```

- Higher Memory Limits
  - Set max-locked-memory limit to unlimited
  - Set number of open files limit to 65535
- Zone Reclaim mode
  - Set zone_reclaim_mode to 1
- Disable firewall daemon to help MPI job launchers

Deploying HPC VM images

As shown, the HPC VM images are available from Azure Marketplace and Azure CLI. They can be deployed through a variety of deployment vehicles on Azure (Azure CycleCloud, Azure Batch, ARM templates, etc.). AzureHPC scripts provide an easy way to quickly deploy an HPC cluster using these images.

High-performance computing on InfiniBand enabled HB-series and N-series VMs

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure's InfiniBand enabled H-series and N-series VMs are designed to deliver leadership-class performance, Message Passing Interface (MPI) scalability, and cost efficiency for a variety of real-world HPC and AI workloads. These high-performance computing (HPC) optimized VMs are used to solve some of the most computationally intensive problems in science and engineering such as: fluid dynamics, earth modeling, weather simulations, etc.

These articles describe how to get started on the InfiniBand-enabled H-series and N-series VMs on Azure as well as optimal configuration of the HPC and AI workloads on the VMs for scalability.

Features and capabilities

The InfiniBand enabled HB-series and N-series VMs are designed to provide the best HPC performance, MPI scalability, and cost efficiency for HPC workloads. See HB-series and N-series VMs to learn more about the features and capabilities of the VMs.

RDMA and InfiniBand

RDMA capable HB-series and N-series VMs communicate over the low latency and high bandwidth InfiniBand network. The RDMA capability over such an interconnect is critical to boost the scalability and performance of distributed-node HPC and AI workloads. The InfiniBand enabled H-series and N-series VMs are connected in a non-blocking fat tree with a low-diameter design for optimized and consistent RDMA performance. See Enable InfiniBand to learn more about setting up InfiniBand on the InfiniBand enabled VMs.

Message passing interface

The SR-IOV enabled HB-series and N-series support almost all MPI libraries and versions. Some of the most commonly used MPI libraries are: Intel MPI, OpenMPI, HPC-X, MVAPICH2, MPICH, Platform MPI. All remote direct memory access (RDMA) verbs are supported. See Set up MPI to learn more about installing various supported MPI libraries and their optimal configuration.

Get started

The first step is to select the HB-series and N-series VM type optimal for the workload based on the VM specifications and RDMA capability. Second, configure the VM by enabling InfiniBand. There are various methods to doing this including using optimized VM images with drivers baked-in; see Optimization for Linux and Enable InfiniBand for details. Third, for distributed node workloads, choosing and configuring MPI appropriately is critical. See Set up MPI for details. Fourth, for performance and scalability, optimally configure the workloads by following guidance specific to the VM family, such as for HBv3-series overview and HC-series overview.

Next steps

- Learn about configuring and optimizing the InfiniBand enabled HB-series and N-series VMs.
- Review the HBv3-series overview and HC-series overview to learn about optimally configuring workloads for performance and scalability.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- Test your knowledge with a learning module on optimizing HPC applications on Azure.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 08/22/2024

Known issues with HB-series and N-series VMs

09/22/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article attempts to list recent common issues and their solutions when using the HB-series and N-series HPC and GPU VMs.

Cache topology on Standard_HB120rs_v3

`lstopo` displays incorrect cache topology on the Standard_HB120rs_v3 VM size. It may display that there's only 32 MB L3 per nonuniform memory access (NUMA) node. However, in practice, there's indeed 120 MB L3 per NUMA as expected since the same 480 MB of L3 to the entire VM is available as with the other constrained-core HBv3 VM sizes. This incorrect display is a cosmetic error and shouldn't affect workloads.

qp0 Access Restriction

To prevent low-level hardware access that can result in security vulnerabilities, Queue Pair 0 isn't accessible to guest VMs. This restriction should only affect actions typically associated with administration of the ConnectX InfiniBand network interface card (NIC) and running some InfiniBand diagnostics like ibdiagnet, but not end-user applications.

MOFED installation on Ubuntu

On Ubuntu-18.04 based marketplace VM images with kernels version `5.4.0-1039-azure #42` and newer, some older Mellanox OFED are incompatible causing an increase in VM boot time up to 30 minutes in some cases. This issue is reported in both Mellanox OFED versions 5.2-1.0.4.0 and 5.2-2.2.0.0. The issue is resolved with Mellanox OFED 5.3-1.0.0.1. If it's necessary to use the incompatible OFED, a solution is to use the **Canonical:UbuntuServer:18_04-lts-gen2:18.04.202101290** marketplace VM image, or older and not to update the kernel.

Accelerated Networking on HB, HC, HBv2, HBv3, HBv4, HX, NDv2 and NDv4

Azure Accelerated Networking is now available on the RDMA and InfiniBand capable. It's also available on SR-IOV enabled VM sizes HB, HC, HBv2, HBv3, HBv4, HX, NDv2, and NDv4. This

capability now allows enhanced throughout (up to 30 Gbps) and latencies over the Azure Ethernet network. Though this enhanced throughput is separate from the RDMA capabilities over the InfiniBand network, some platform changes for this capability could affect behavior of certain MPI implementations when running jobs over InfiniBand. Specifically the InfiniBand interface on some VMs may have a slightly different name (mlx5_1 as opposed to earlier mlx5_0). This issue may require tweaking of the MPI command lines, especially when using the UCX interface (commonly with OpenMPI and HPC-X).

For more information on this issue, see the TechCommunity article with instructions on how to address any observed issues.

InfiniBand driver installation on non-SR-IOV VMs

Currently H16r, H16mr, and NC24r aren't SR-IOV enabled. For more information on the InfiniBand stack bifurcation, see Azure VM sizes - HPC. InfiniBand can be configured on the SR-IOV enabled VM sizes with the OFED drivers while the non-SR-IOV VM sizes require ND drivers. This IB support is available appropriately for RHEL and Ubuntu.

Duplicate MAC with cloud-init with Ubuntu on H-series and N-series VMs

There's a known issue with cloud-init on Ubuntu VM images as it tries to bring up the IB interface. This issue can happen either on VM reboot or when trying to create a VM image after generalization. The VM boot logs may show an error like so:

Output

```
“Starting Network Service...RuntimeError: duplicate mac found! both 'eth1' and
'ib0' have mac”.
```

This 'duplicate MAC with cloud-init on Ubuntu" is a known issue. Newer kernels plan to resolve the issue. If this issue is encountered, the workaround is:

1. Deploy the (Ubuntu 18.04) marketplace VM image
2. Install the necessary software packages to enable IB (instruction here)
3. Edit waagent.conf and set EnableRDMA=y
4. Disable networking in cloud-init:

   Bash

   ```
   echo network: {config: disabled} | sudo tee /etc/cloud/cloud.cfg.d/99-
   ```

```
disable-network-config.cfg
```

5. To remove the MAC, edit netplan's networking configuration file generated by cloud-init:

```
Bash

sudo bash -c "cat > /etc/netplan/50-cloud-init.yaml" <<'EOF'
network:
  ethernets:
    eth0:
      dhcp4: true
  version: 2
EOF
```

DRAM on HB-series VMs

HB-series VMs can only expose 228 GB of RAM to guest VMs at this time. Similarly, 458 GB on HBv2 and 448 GB on HBv3 VMs. This limitation is due to a known limitation of Azure hypervisor to prevent pages from being assigned to the local DRAM of AMD CCXs (NUMA domains) reserved for the guest VM.

GSS Proxy

GSS Proxy has a known bug in RHEL 7.5 that can manifest as a significant performance and responsiveness penalty when used with NFS. This bug can be mitigated with:

```
Bash

sudo sed -i 's/GSS_USE_PROXY="yes"/GSS_USE_PROXY="no"/g' /etc/sysconfig/nfs
```

Cache Cleaning

On HPC systems, it's often useful to clean up the memory after a job finishes before the next user is assigned the same node. After running applications in Linux, you may find that your available memory reduces while your buffer memory increases, despite not running any applications.

```
[root@mun-smc05 ~]# free -g
              total        used        free      shared  buff/cache   available
Mem:            251           0         233           0          17         248
Swap:             3           0           3
```

Using `numactl -H` shows which NUMAnodes the memory is buffered with (possibly all). In Linux, users can clean the caches in three ways to return buffered or cached memory to ‘free’.

You need to be root or have sudo permissions.

Bash

```
sudo echo 1 > /proc/sys/vm/drop_caches [frees page-cache]
sudo echo 2 > /proc/sys/vm/drop_caches [frees slab objects e.g. dentries, inodes]
sudo echo 3 > /proc/sys/vm/drop_caches [cleans page-cache and slab objects]
```



Thermal alerts and reduced performance due to thermal degradation of NVIDIA GPUs in ND H100/H200 v5 VMs

Microsoft has identified an issue that may cause some GPUs that have been in service for extended periods to show thermal alerts or reduced performance. In some cases, this can be due to software reporting errors in older NVIDIA drivers, while in others it may indicate a true hardware issue, with thermal degradation over time resulting in throttling and reduced performance.

To ensure accurate monitoring, we recommend upgrading to NVIDIA driver version 570.124.06 or higher. If thermal alerts or throttling continue after the upgrade, this may point to a hardware problem. Microsoft is proactively replacing hardware that is currently exhibiting persistent thermal throttling, as well as units identified as being at elevated risk. We remain committed to maintaining optimal system performance and reliability for our customers.

Kernel warnings

You may ignore the following kernel warning messages when booting an HB-series VM under Linux. This is due to a known limitation of the Azure hypervisor that will be addressed over time.

Output

```
[  0.004000] WARNING: CPU: 4 PID: 0 at arch/x86/kernel/smpboot.c:376
topology_sane.isra.3+0x80/0x90
[  0.004000] sched: CPU #4's llc-sibling CPU #0 is not on the same node! [node: 1
!= 0]. Ignoring dependency.
[  0.004000] Modules linked in:
[  0.004000] CPU: 4 PID: 0 Comm: swapper/4 Not tainted 3.10.0-957.el7.x86_64 #1
[  0.004000] Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine,
BIOS 090007 05/18/2018
```

```
[  0.004000] Call Trace:
[  0.004000] [<ffffffffb8361dc1>] dump_stack+0x19/0x1b
[  0.004000] [<ffffffffb7c97648>] __warn+0xd8/0x100
[  0.004000] [<ffffffffb7c976cf>] warn_slowpath_fmt+0x5f/0x80
[  0.004000] [<ffffffffb7c02b34>] ? calibrate_delay+0x3e4/0x8b0
[  0.004000] [<ffffffffb7c574c0>] topology_sane.isra.3+0x80/0x90
[  0.004000] [<ffffffffb7c57782>] set_cpu_sibling_map+0x172/0x5b0
[  0.004000] [<ffffffffb7c57ce1>] start_secondary+0x121/0x270
[  0.004000] [<ffffffffb7c000d5>] start_cpu+0x5/0x14
[  0.004000] ---[ end trace 73fc0e0825d4ca1f ]---
```

Next steps

- Review the HB-series overview and HC-series overview to learn about optimally configuring workloads for performance and scalability.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher-level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Set up Message Passing Interface for HPC

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Message Passing Interface (MPI) is an open library and defacto standard for distributed memory parallelization. It's commonly used across many HPC workloads. HPC workloads on the RDMA capable HB-series and N-series VMs can use MPI to communicate over the low latency and high bandwidth InfiniBand network.

- The SR-IOV enabled VM sizes on Azure allow almost any flavor of MPI to be used with Mellanox OFED.
- On non-SR-IOV enabled VMs, supported MPI implementations use the Microsoft Network Direct (ND) interface to communicate between VMs. Hence, only Microsoft MPI (MS-MPI) 2012 R2 or later and Intel MPI 5.x versions are supported. Later versions (2017, 2018) of the Intel MPI runtime library may or may not be compatible with the Azure RDMA drivers.

For SR-IOV enabled RDMA capable VMs, Ubuntu-HPC VM images and AlmaLinux-HPC VM images are suitable. These VM images come optimized and preloaded with the OFED drivers for RDMA and various commonly used MPI libraries and scientific computing packages and are the easiest way to get started.

Though the examples here are for RHEL, but the steps are general and can be used for any compatible Linux operating system such as Ubuntu (18.04, 20.04, 22.04) and SLES (12 SP4 and 15 SP4). More examples for setting up other MPI implementations on others distros is on the azhpc-images repo.

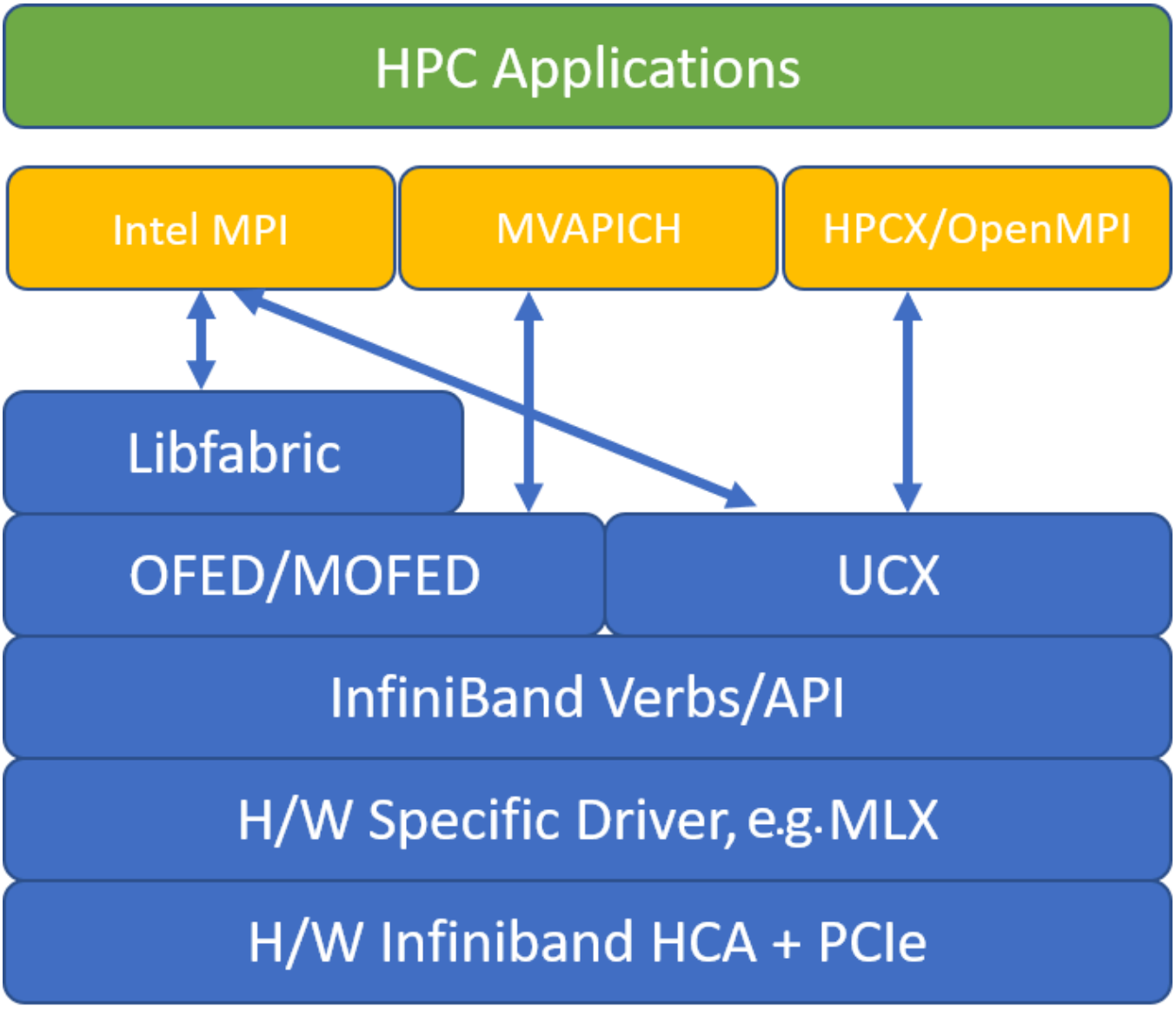
> **Note**
>
> Running MPI jobs on SR-IOV enabled VMs with certain MPI libraries (such as Platform MPI) may require setting up of partition keys (p-keys) across a tenant for isolation and security. Follow the steps in the **Discover partition keys** section for details on determining the p-key values and setting them correctly for an MPI job with that MPI library.

> **Note**
>
> The code snippets below are examples. We recommend using the latest stable versions of the packages, or referring to the **azhpc-images repo**.

Choosing MPI library

If an HPC application recommends a particular MPI library, try that version first. If you have flexibility regarding which MPI you can choose, and you want the best performance, try HPC-X. Overall, the HPC-X MPI performs the best by using the UCX framework for the InfiniBand interface, and takes advantage of all the Mellanox InfiniBand hardware and software capabilities. Additionally, HPC-X and OpenMPI are ABI compatible, so you can dynamically run an HPC application with HPC-X that was built with OpenMPI. Similarly, Intel MPI, MVAPICH, and MPICH are ABI compatible.

The following figure illustrates the architecture for the popular MPI libraries.



HPC-X

The HPC-X software toolkit contains UCX and HCOLL and can be built against UCX.

Bash

```
HPCX_VERSION="v2.6.0"
HPCX_DOWNLOAD_URL=https://azhpcstor.blob.core.windows.net/azhpc-images-store/hpcx-
```

```
v2.6.0-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-redhat7.7-x86_64.tbz
wget --retry-connrefused --tries=3 --waitretry=5 $HPCX_DOWNLOAD_URL
tar -xvf hpcx-${HPCX_VERSION}-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-redhat7.7-x86_64.tbz
mv hpcx-${HPCX_VERSION}-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-redhat7.7-x86_64
${INSTALL_PREFIX}
HPCX_PATH=${INSTALL_PREFIX}/hpcx-${HPCX_VERSION}-gcc-MLNX_OFED_LINUX-5.0-1.0.0.0-
redhat7.7-x86_64
```

The following command illustrates some recommended mpirun arguments for HPC-X and OpenMPI.

Bash

```
mpirun -n $NPROCS --hostfile $HOSTFILE --map-by
ppr:$NUMBER_PROCESSES_PER_NUMA:numa:pe=$NUMBER_THREADS_PER_PROCESS -report-bindings
$MPI_EXECUTABLE
```

where:

Expand table

Parameter	Description
`NPROCS`	Specifies the number of MPI processes. For example: `-n 16`.
`$HOSTFILE`	Specifies a file containing the hostname or IP address, to indicate the location of where the MPI processes run. For example: `--hostfile hosts`.
`$NUMBER_PROCESSES_PER_NUMA`	Specifies the number of MPI processes that run in each NUMA domain. For example, to specify four MPI processes per NUMA, you use `--map-by ppr:4:numa:pe=1`.
`$NUMBER_THREADS_PER_PROCESS`	Specifies the number of threads per MPI process. For example, to specify one MPI process and four threads per NUMA, you use `--map-by ppr:1:numa:pe=4`.
`-report-bindings`	Prints MPI processes mapping to cores, which is useful to verify that your MPI process pinning is correct.
`$MPI_EXECUTABLE`	Specifies the MPI executable built linking in MPI libraries. MPI compiler wrappers do this automatically. For example: `mpicc` or `mpif90`.

An example of running the OSU latency microbenchmark is as follows:

Bash

```
${HPCX_PATH}mpirun -np 2 --map-by ppr:2:node -x UCX_TLS=rc
${HPCX_PATH}/ompi/tests/osu-micro-benchmarks-5.3.2/osu_latency
```

Optimizing MPI collectives

MPI Collective communication primitives offer a flexible, portable way to implement group communication operations. They're widely used across various scientific parallel applications and have a significant impact on the overall application performance. Refer to the TechCommunity article for details on configuration parameters to optimize collective communication performance using HPC-X and HCOLL library for collective communication.

As an example, if you suspect your tightly coupled MPI application is doing an excessive amount of collective communication, you can try enabling hierarchical collectives (HCOLL). To enable those features, use the following parameters.

Bash

```
-mca coll_hcoll_enable 1 -x HCOLL_MAIN_IB=<MLX device>:<Port>
```

> **Note**
>
> With HPC-X 2.7.4+, it may be necessary to explicitly pass LD_LIBRARY_PATH if the UCX version on MOFED vs. that in HPC-X is different.

OpenMPI

Install UCX as described above. HCOLL is part of the HPC-X software toolkit and doesn't requires special installation.

OpenMPI can be installed from the packages available in the repo.

Bash

```
sudo yum install -y openmpi
```

We recommend building a latest, stable release of OpenMPI with UCX.

Bash

```
OMPI_VERSION="4.0.3"
OMPI_DOWNLOAD_URL=https://download.open-mpi.org/release/open-
mpi/v4.0/openmpi-${OMPI_VERSION}.tar.gz
wget --retry-connrefused --tries=3 --waitretry=5 $OMPI_DOWNLOAD_URL
tar -xvf openmpi-${OMPI_VERSION}.tar.gz
cd openmpi-${OMPI_VERSION}
./configure --prefix=${INSTALL_PREFIX}/openmpi-${OMPI_VERSION} --with-
```

```
ucx=${UCX_PATH} --with-hcoll=${HCOLL_PATH} --enable-mpirun-prefix-by-default --with-platform=contrib/platform/mellanox/optimized && make -j$(nproc) && make install
```

For optimal performance, run OpenMPI with `ucx` and `hcoll`. Also see the example with HPC-X.

Bash

```
${INSTALL_PREFIX}/bin/mpirun -np 2 --map-by node --hostfile ~/hostfile -mca pml ucx --mca btl ^vader,tcp,openib -x UCX_NET_DEVICES=mlx5_0:1  -x UCX_IB_PKEY=0x0003 ./osu_latency
```

Check your partition key as mentioned above.

Intel MPI

Download your choice of version of Intel MPI. The Intel MPI 2019 release switched from the Open Fabrics Alliance (OFA) framework to the Open Fabrics Interfaces (OFI) framework, and currently supports libfabric. There are two providers for InfiniBand support: mlx and verbs. Change the I_MPI_FABRICS environment variable depending on the version.

- Intel MPI 2019 and 2021: use `I_MPI_FABRICS=shm:ofi`, `I_MPI_OFI_PROVIDER=mlx`. The `mlx` provider uses UCX. Usage of verbs has been found to be unstable and less performant. See the TechCommunity article for more details.
- Intel MPI 2018: use `I_MPI_FABRICS=shm:ofa`
- Intel MPI 2016: use `I_MPI_DAPL_PROVIDER=ofa-v2-ib0`

Here are some suggested mpirun arguments for Intel MPI 2019 update 5+.

Bash

```
export FI_PROVIDER=mlx
export I_MPI_DEBUG=5
export I_MPI_PIN_DOMAIN=numa

mpirun -n $NPROCS -f $HOSTFILE $MPI_EXECUTABLE
```

where:

Expand table

Parameter	Description
`FI_PROVIDER`	Specifies which libfabric provider to use, which will affect the API, protocol, and network used. verbs is another option, but generally mlx gives you better performance.

Parameter	Description
`I_MPI_DEBUG`	Specifies the level of extra debug output, which can provide details about where processes are pinned, and which protocol and network are used.
`I_MPI_PIN_DOMAIN`	Specifies how you want to pin your processes. For example, you can pin to cores, sockets, or NUMA domains. In this example, you set this environmental variable to numa, which means processes will be pinned to NUMA node domains.

Optimizing MPI collectives

There are some other options that you can try, especially if collective operations are consuming a significant amount of time. Intel MPI 2019 update 5+ supports the provide mlx and uses the UCX framework to communicate with InfiniBand. It also supports HCOLL.

Bash

```
export FI_PROVIDER=mlx
export I_MPI_COLL_EXTERNAL=1
```

Non SR-IOV VMs

For non SR-IOV VMs, an example of downloading the 5.x runtime free evaluation version is as follows:

Bash

```
wget http://registrationcenter-
download.intel.com/akdlm/irc_nas/tec/9278/l_mpi_p_5.1.3.223.tgz
```

For installation steps, see the Intel MPI Library Installation Guide. Optionally, you may want to enable ptrace for non-root non-debugger processes (needed for the most recent versions of Intel MPI).

Bash

```
echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope
```

SUSE Linux

For SUSE Linux Enterprise Server VM image versions - SLES 12 SP3 for HPC, SLES 12 SP3 for HPC (Premium), SLES 12 SP1 for HPC, SLES 12 SP1 for HPC (Premium), SLES 12 SP4 and SLES

15, the RDMA drivers are installed and Intel MPI packages are distributed on the VM. Install Intel MPI by running the following command:

```bash
sudo rpm -v -i --nodeps /opt/intelMPI/intel_mpi_packages/*.rpm
```

MVAPICH

The following is an example of building MVAPICH2. Note newer versions may be available than what is used below.

```bash
wget http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.3.tar.gz
tar -xv mvapich2-2.3.tar.gz
cd mvapich2-2.3
./configure --prefix=${INSTALL_PREFIX}
make -j 8 && make install
```

An example of running the OSU latency microbenchmark is as follows:

```bash
${INSTALL_PREFIX}/bin/mpirun_rsh -np 2 -hostfile ~/hostfile MV2_CPU_MAPPING=48
./osu_latency
```

The following list contains several recommended `mpirun` arguments.

```bash
export MV2_CPU_BINDING_POLICY=scatter
export MV2_CPU_BINDING_LEVEL=numanode
export MV2_SHOW_CPU_BINDING=1
export MV2_SHOW_HCA_BINDING=1

mpirun -n $NPROCS -f $HOSTFILE $MPI_EXECUTABLE
```

where:

Expand table

Parameter	Description
`MV2_CPU_BINDING_POLICY`	Specifies which binding policy to use, which will affect how processes are pinned to core IDs. In this case, you specify `scatter`, so processes are evenly

Parameter	Description
	scattered among the NUMA domains.
`MV2_CPU_BINDING_LEVEL`	Specifies where to pin processes. In this case, you set it to numanode, which means processes are pinned to units of NUMA domains.
`MV2_SHOW_CPU_BINDING`	Specifies if you want to get debug information about where the processes are pinned.
`MV2_SHOW_HCA_BINDING`	Specifies if you want to get debug information about which host channel adapter each process is using.

Platform MPI

Install required packages for Platform MPI Community Edition.

Bash

```
sudo yum install libstdc++.i686
sudo yum install glibc.i686
Download platform MPI at
https://www.ibm.com/developerworks/downloads/im/mpi/index.html
sudo ./platform_mpi-09.01.04.03r-ce.bin
```

Follow the installation process.

MPICH

Install UCX as described above. Build MPICH.

Bash

```
wget https://www.mpich.org/static/downloads/3.3/mpich-3.3.tar.gz
tar -xvf mpich-3.3.tar.gz
cd mpich-3.3
./configure --with-ucx=${UCX_PATH} --prefix=${INSTALL_PREFIX} --with-device=ch4:ucx
make -j 8 && make install
```

Running MPICH.

Bash

```
${INSTALL_PREFIX}/bin/mpiexec -n 2 -hostfile ~/hostfile -env UCX_IB_PKEY=0x0003 -
bind-to hwthread ./osu_latency
```

Check your partition key as mentioned above.

OSU MPI Benchmarks

Download OSU MPI Benchmarks and untar.

```Bash
wget http://mvapich.cse.ohio-state.edu/download/mvapich/osu-micro-benchmarks-5.5.tar.gz
tar -xvf osu-micro-benchmarks-5.5.tar.gz
cd osu-micro-benchmarks-5.5
```

Build Benchmarks using a particular MPI library:

```Bash
CC=<mpi-install-path/bin/mpicc>CXX=<mpi-install-path/bin/mpicxx> ./configure
make
```

MPI Benchmarks are under `mpi/` folder.

Discover partition keys

Discover partition keys (p-keys) for communicating with other VMs within the same tenant (Availability Set or Virtual Machine Scale Set).

```Bash
/sys/class/infiniband/mlx5_0/ports/1/pkeys/0
/sys/class/infiniband/mlx5_0/ports/1/pkeys/1
```

The larger of the two is the tenant key that should be used with MPI. Example: If the following are the p-keys, 0x800b should be used with MPI.

```Bash
cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/0
0x800b
cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/1
0x7fff
```

Note interfaces are named as `mlx5_ib*` inside HPC VM images.

Also note that as long as the tenant (Availability Set or Virtual Machine Scale Set) exists, the PKEYs remain the same. This is true even when nodes are added/deleted. New tenants get different PKEYs.

Set up user limits for MPI

Set up user limits for MPI.

Bash

```
cat << EOF | sudo tee -a /etc/security/limits.conf
*               hard    memlock         unlimited
*               soft    memlock         unlimited
*               hard    nofile          65535
*               soft    nofile          65535
EOF
```

Set up SSH keys for MPI

Set up SSH keys for MPI types that require it.

Bash

```
ssh-keygen -f /home/$USER/.ssh/id_rsa -t rsa -N ''
cat << EOF > /home/$USER/.ssh/config
Host *
    StrictHostKeyChecking no
EOF
cat /home/$USER/.ssh/id_rsa.pub >> /home/$USER/.ssh/authorized_keys
chmod 600 /home/$USER/.ssh/authorized_keys
chmod 644 /home/$USER/.ssh/config
```

The above syntax assumes a shared home directory, else `.ssh` directory must be copied to each node.

Next steps

- Learn about the InfiniBand enabled HB-series and N-series VMs.
- Review the HBv3-series overview and HC-series overview.
- Read Optimal MPI process placement for HB-series VMs.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 09/17/2024

Scaling HPC applications

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Optimal scale-up and scale-out performance of HPC applications on Azure requires performance tuning and optimization experiments for the specific workload. This section and the VM series-specific pages offer general guidance for scaling your applications.

Application setup

The azurehpc repo contains many examples of:

- Setting up and running applications optimally.
- Configuration of file systems, and clusters.
- Tutorials on how to get started easily with some common application workflows.

Optimally scaling MPI

The following suggestions apply for optimal application scaling efficiency, performance, and consistency:

- For smaller scale jobs (< 256K connections) use: `UCX_TLS=rc,sm`
- For larger scale jobs (> 256K connections) use: `UCX_TLS=dc,sm`
- To calculate the number of connections for your MPI job, use: `Max Connections = (processes per node) x (number of nodes per job) x (number of nodes per job)`

Adaptive Routing

Adaptive Routing (AR) allows Azure Virtual Machines (VMs) running EDR and HDR InfiniBand to automatically detect and avoid network congestion by dynamically selecting optimal network paths. As a result, AR offers improved latency and bandwidth on the InfiniBand network, which in turn drives higher performance and scaling efficiency. For more information, see TechCommunity article.

Process pinning

- Pin processes to cores using a sequential pinning approach (as opposed to an autobalance approach).
- Binding by Numa/Core/HwThread is better than default binding.

- For hybrid parallel applications (OpenMP+MPI), use four threads and one MPI rank per [CCX](HB-series virtual machines overview including info on CCXs) on HB and HBv2 VM sizes.
- For pure MPI applications, experiment with between one to four MPI ranks per CCX for optimal performance on HB and HBv2 VM sizes.
- Some applications with extreme sensitivity to memory bandwidth may benefit from using a reduced number of cores per CCX. For these applications, using three or two cores per CCX may reduce memory bandwidth contention and yield higher real-world performance or more consistent scalability. In particular, MPI 'Allreduce' may benefit from this approach.
- For larger scale runs, it's recommended to use UD or hybrid RC+UD transports. Many MPI libraries/runtime libraries use these transports internally (such as UCX or MVAPICH2). Check your transport configurations for large-scale runs.

Compiling applications

▼ Click to expand

Though not necessary, compiling applications with appropriate optimization flags provides the best scale-up performance on HB and HC-series VMs.

AMD Optimizing C/C++ Compiler

The AMD Optimizing C/C++ Compiler (AOCC) compiler system offers a high level of advanced optimizations, multi-threading, and processor support that includes global optimization, vectorization, inter-procedural analyses, loop transformations, and code generation. AOCC compiler binaries are suitable for Linux systems having GNU C Library (glibc) version 2.17 and above. The compiler suite consists of a C/C++ compiler (clang), a Fortran compiler (FLANG), and a Fortran front end to Clang (Dragon Egg).

Clang

Clang is a C, C++, and Objective-C compiler handling preprocessing, parsing, optimization, code generation, assembly, and linking. Clang supports the `-march=znver1` flag to enable best code generation and tuning for AMD's Zen based x86 architecture.

FLANG

The FLANG compiler is a recent addition to the AOCC suite (added April 2018) and is currently in prerelease for developers to download and test. Based on Fortran 2008, AMD extends the GitHub version of FLANG. The FLANG compiler supports all Clang compiler options and other number of FLANG-specific compiler options.

DragonEgg

DragonEgg is a gcc plugin that replaces GCC's optimizers and code generators from the LLVM project. DragonEgg that comes with AOCC works with gcc-4.8.x, has been tested for x86-32/x86-64 targets, and has been successfully used on various Linux platforms.

GFortran is the actual frontend for Fortran programs responsible for preprocessing, parsing, and semantic analysis generating the GCC GIMPLE intermediate representation (IR). DragonEgg is a GNU plugin, plugging into GFortran compilation flow. It implements the GNU plugin API. With the plugin architecture, DragonEgg becomes the compiler driver, driving the different phases of compilation. After following the download and installation instructions, Dragon Egg can be invoked using:

Bash

```
gfortran [gFortran flags]
   -fplugin=/path/AOCC-1.2-Compiler/AOCC-1.2-
   FortranPlugin/dragonegg.so [plugin optimization flags]
   -c xyz.f90 $ clang -O3 -lgfortran -o xyz xyz.o $./xyz
```

PGI Compiler

PGI Community Edition 17 is confirmed to work with AMD EPYC. A PGI-compiled version of STREAM does deliver full memory bandwidth of the platform. The newer Community Edition 18.10 (Nov 2018) should likewise work well. Use this CLI command to compile with the Intel Compiler:

Bash

```
pgcc $(OPTIMIZATIONS_PGI) $(STACK) -DSTREAM_ARRAY_SIZE=800000000 stream.c -o
stream.pgi
```

Intel Compiler

Intel Compiler 18 is confirmed to work with AMD EPYC. Use this CLI command to compile with the Intel Compiler.

Bash

```
icc -o stream.intel stream.c -DSTATIC -DSTREAM_ARRAY_SIZE=800000000 -mcmodel=large -
shared-intel -Ofast –qopenmp
```

GCC Compiler

For HPC workloads, AMD recommends GCC compiler 7.3 or newer. Older versions, such as 4.8.5 included with RHEL 7.4, aren't recommended. GCC 7.3, and newer, delivers higher performance on HPL, HPCG, and DGEMM tests.

Bash

```
gcc $(OPTIMIZATIONS) $(OMP) $(STACK) $(STREAM_PARAMETERS) stream.c -o stream.gcc
```

Next steps

- Test your knowledge with a learning module on optimizing HPC applications on Azure.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- Learn more about HPC on Azure.

Last updated on 08/22/2024

NVMe Overview

Article • 08/22/2024

NVM Express (NVMe) is a communication protocol that facilitates faster and more efficient data transfer between servers and storage systems by using nonvolatile memory (NVM). With NVMe, data can be transferred at the highest throughput and with the fastest response times. In this overview, learn about NVMe support on virtual machines (VM) created in Azure.

NVMe offers higher input/output operations per second (IOPS) and higher throughput in megabytes per second (MBps). This capability can significantly improve the performance of both temporary (local) and remote NVMe disk storage with Azure managed disks. Higher performance is especially beneficial for I/O-intensive workloads that require fast data transfer to the Azure managed disks.

Benefits

Azure storage performance with NVMe disks

The performance gains that your applications can achieve depend on several factors, including VM type, VM size, and I/O block size. To identify the ideal block size and to achieve peak transfer rates, test on the individual VM sizes. It's important to optimize the I/O size based on the specific requirements of your application. For more information on the performance capabilities of individual VM types, see the documentation about VM sizes in Azure.

Workloads

The VM families that use NVMe disks demonstrate performance benefits compared to Small Computer System Interface (SCSI) across various workloads that require higher I/O and improved storage performance. These workloads include application servers, web servers, databases, data warehousing, and analytics. You can achieve these benefits across VMs that use temporary and/or remote NVMe disks.

Azure Boost & NVMe VM benefits

Azure Boost is a Microsoft-designed system that offloads server virtualization processes traditionally performed by the hypervisor and host OS onto purpose-built software and

hardware. Offloading these processes enables faster storage and networking performance for Azure VM customers.

One of the primary advantages of Azure Boost is its ability to enhance the throughput of Azure managed disks and local storage. This enhancement is enabled by offloading the storage processing tasks to hardware dedicated for Azure Boost.

Azure Boost also optimizes performance by using the industry-standard NVMe interface, which capitalizes on the low latency and internal parallelism of solid-state storage drives. For details on the performance that Azure Boost offers, refer to the Microsoft blog post about general availability of Azure Boost.

SCSI to NVMe

For v6, Ebsv5, and greater VMs; you need to transition from SCSI to NVMe. For more information, read about SCSI to NVMe using Linux.

Supportability

VM sizes

Typically, the older generations of general purpose, memory optimized, and compute optimized VMs (D/Ev5 or Fv2 and older) support SCSI. The newer generations (Ebsv5, Da/Ea/Fav6 and newer) typically support only the NVMe storage interface (or some offering supports SCSI and NVMe, both interfaces, Ebsv5 is an example of such offering). Also, starting from Da/Ea/Fav6 and some of the future offering, we have introduced the NVMe interface for the local SSD disks.

For specifics about which VM generations support which storage types, check the documentation about VM sizes in Azure.

VM image generation

If you think your workload benefits from NVMe performance, you must switch to the Generation 2 VMs. To create a Generation 2 VM, follow the documented steps.

> **Note**
>
> We currently do not support NVMe disks on Generation 1 VMs.

If you attempt to use a Generation 1 VM image, you won't be able to select an NVMe-capable VM or you receive an error message stating `that the selected image is not supported for NVMe.`

Availability of Support

NVMe support is available in more than 50 of the most popular OS images. We're not adding NVMe support to older OS images. However, we continuously improve the OS image coverage. For updates on the latest OS image support for both Linux and Windows, we recommend this documentation on OS image that supports NVMe.

If the OS you want to use doesn't support NVMe, use a VM series that still has SCSI support. Many of the latest Azure VM generations are NVMe-only and require an OS image that supports NVMe.

How to create a VM with the NVMe interface

You can enable NVMe during VM creation by using methods such as:

- Azure portal
- Azure CLI
- Azure PowerShell
- Azure Resource Manager templates

To create an NVMe-enabled VM, you must first enable the NVMe option on a VM and select the NVMe disk controller type for the VM. You can also update a VM to NVMe when stopped and deallocated, if the VM size supports NVMe.

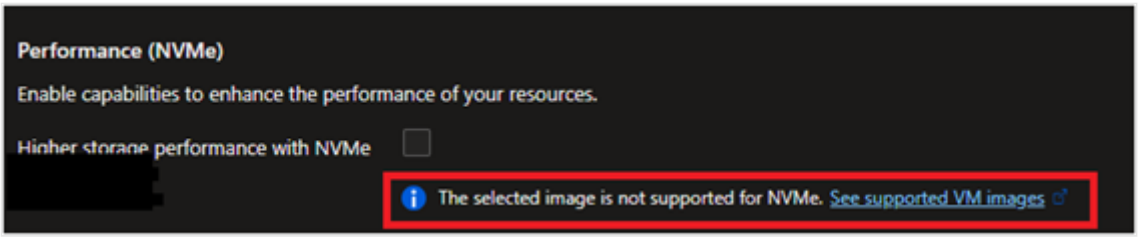
Mark as NVMe capable

You can create NVMe interface enabled VM only by using an image (available in Marketplace, or shared in Azure Compute Gallery within your company) marked as NVMe.

> ⚠ **Warning**
>
> If you use an unmarked image, a SCSI-based VM is created, and the VM might not perform as you intended.

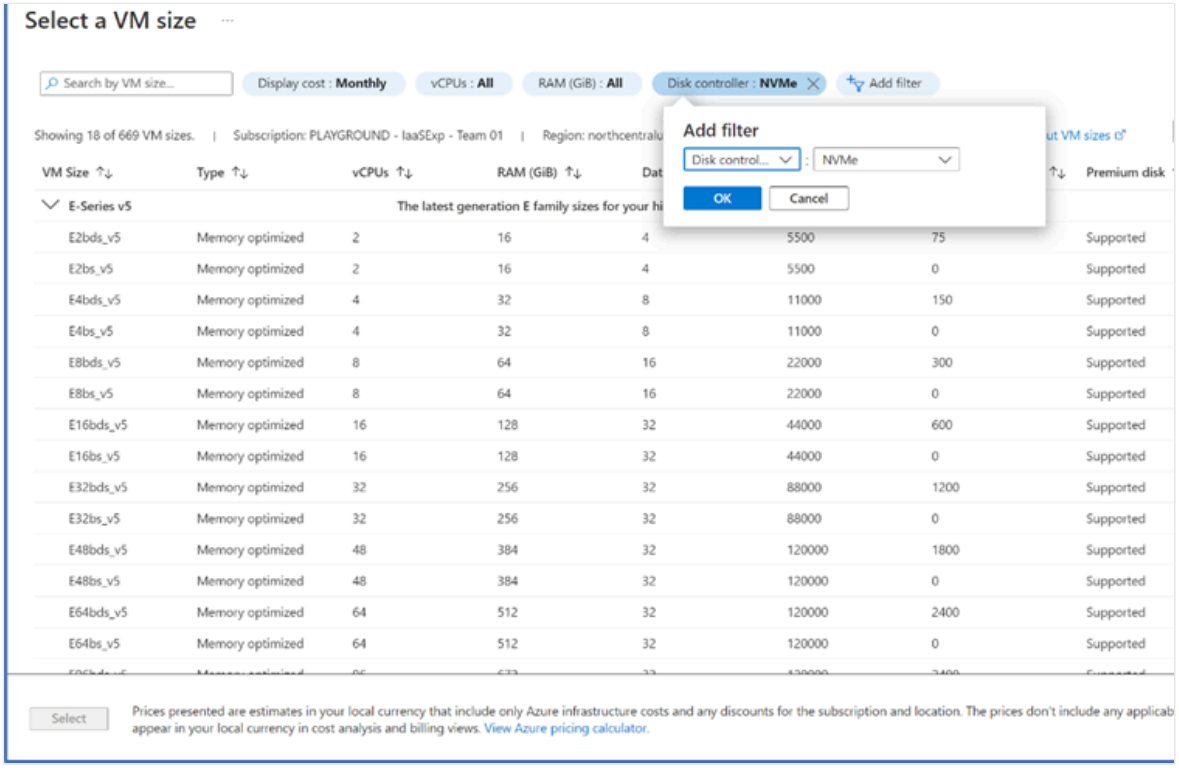
To create VMs with an NVMe interface, it's essential to choose one of the supported OS images marked as NVMe. If your current OS image isn't supported for NVMe, you get

this error message: `"The selected image is not supported for NVMe. See supported VM images."`

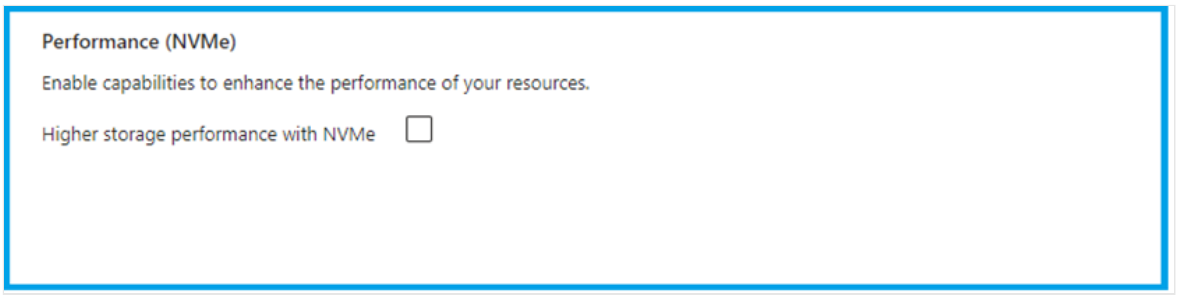


Using Azure portal to enable the NVMe interface during VM creation

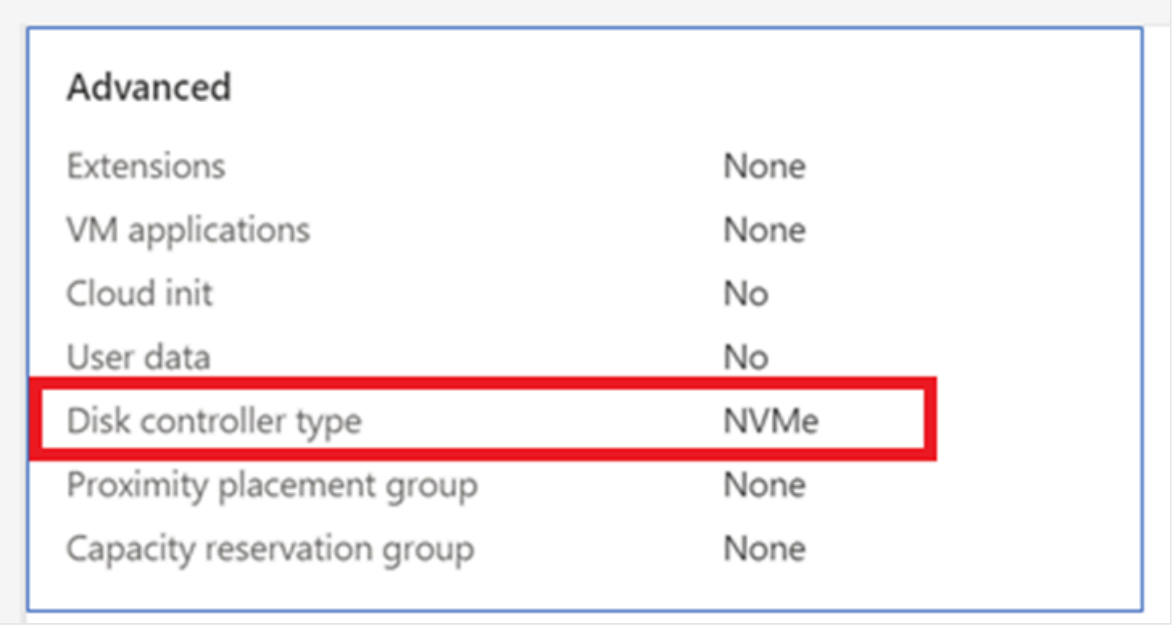
1. Add a disk controller filter. To find the NVMe-eligible sizes, select **See All Sizes**, select the **Disk controller** filter, and then select **NVMe**.



2. Enable the NVMe feature by going to the **Advanced** tab and selecting the checkbox under **Performance (NVMe)**.



3. Verify that the feature is enabled by going to the **Review and Create** tab and confirming that the **Disk controller type** value is **NVMe**.



Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

FAQ for Temp NVMe disks

This article addresses some common questions about support for temporary Non-Volatile Memory Express (NVMe) disks on virtual machines (VMs) created in Azure. In this article, the term "temp disk" is referring to the temporary disk that is created on local SSD.

What are the prerequisites to create temp disk that uses NVMe interface for my VM?

NVMe support for local temp disks varies at a virtual machine (VM) family level. All of the new v6 VMs with local disk support NVMe temp disks, but v5 VMs and earlier versions may not support NVMe local temp disks, as most of those sizes local disks are on the SCSI interface. Some VM families like L-series, include attached NVMe SSDs from Lsv2 generation VMs onward. Refer to the VM family overviews to determine if NVMe temp disks are supported. Also, note your Guest OS needs to have NVMe driver support in order to use local NVMe, which most of the recent OS releases have by default.

How are temp NVMe disks in my Dl/D/E_v6 VMs different from remote NVMe disks?

A key difference is that temp NVMe disks are transient and data on the VMs can disappear, but data on remote NVMe disks stays. Factor in this difference when you're choosing your VM's storage options.

Unlike the previous D/E series VMs, the new v6 VMs come with raw, unformatted NVMe disks. Customers should initialize and format the disks into a file system of their preference after the VM starts up.

The Dl/D/E v6 series VMs are optimized to use the local temp NVMe disk on the node attached directly to the VM rather than previously used local Small Computer System Interface (SCSI) disks. This method allows for higher input/output operations per second and higher throughput for your workloads.

What changes should I prepare for when configuring my VMs with temp NVMe disks?

- Disks should be initialized and formatted after the VM starts up. After user-initiated stops, deallocations, or planned maintenance and Azure-initiated automatic recovery events, the VMs start up with only raw temp NVMe disks. No temp NVMe disks are visible to applications until after they're initialized and formatted.
- You can span separate NVMe drives into one as needed after the VM starts up.
- A Windows pagefile is placed on persistent OS disks (unless you're using the ephemeral OS feature), but you can move it to temp NVMe disks as needed after the VM starts up.
- The ephemeral OS feature, if you use it, converts one or several NVMe disks (depending on the VM and image size) into a slower SCSI/VHD-based disk, similar to how it was on v5 and older VMs. For the larger VM sizes, the remaining drives stay as raw unformatted NVMe.
- The list of resource Stock Keeping Units (SKUs) in the REST API exposes several capabilities of virtual machines. The next iteration of VM sizes uses the faster and more efficient NVMe protocol for local storage, instead of the SCSI protocol that earlier VM sizes use.

  For Azure virtual machines, SCSI-based local storage is a temporary resource disk, and the `MaxResourceVolumeMB` value specifies the size of this disk. In contrast, the `NVMeDiskSizeInMiB` value specifies the size of NVMe-based local storage.

What is the minimum encryption offered by Azure for the temp NVMe disks?

All local NVMe disks on Azure virtual machines (VMs) have built-in encryption enabled by default for data protection. Temp NVMe disks support the feature of data encryption at rest. A unique data encryption key (DEK) encrypts the data for each temp NVMe disk assigned to the VM. A key encryption key (KEK) helps protect the DEK. When you delete a VM, the data on its temp NVMe disk is cryptographically erased. All the v6 VMs that are introduced from 2024 and onward, have their local NVMe drives encrypted by default using hardware based encryption with a Platform Managed Key.

- Data is encrypted using XTS-AES-256, AES-256-GCM, or a stronger cipher.
- Device generates a unique Media Encryption Key (MEK) for each disk in the VM.
- The MEK is:
  - Generated and stored entirely within the device hardware.
  - Never exposed to the host, VM, or any external component including the operator, or Azure system.
- All encryption and decryption operations occur within the device using the MEK.

- The MEK is destroyed when you delete your VM, ensuring that no component can access the data afterward.

What is Enhanced Encryption at Rest?

Enhanced Encryption at rest is another layer of protection that's available on the Azure V6-series and Lsv4-series VMs through support for TCG-OPAL standards. You don't need to take any action for this feature, it's the default for these VM types.

- Azure infrastructure generates and securely stores a PIN value for each MEK.
- The PIN is passed through a Key Derivation Function (KDF) within the storage device to generate a Key Encrypting Key (KEK). The PIN acts as a Platform management key (PMK).
- The MEK is encrypted using the KEK and stored internally within the storage device.
- The KEK isn't stored anywhere and the storage device dynamically generates the KEK when the PIN is used to successfully authenticate it. Without this PIN, the device can't decrypt its own stored data—even if physically accessed.
- The storage device is authenticated when the VM starts. It’s locked when the VM stops, which ensures data is accessible only while the VM is running.

How can I resize a v5 or older VM with a temp disk to v6?

See the FAQ for Azure VM sizes with no local temporary disk.

How can I identify my temp NVMe disk?

You can run the following command for Windows:

PowerShell

```
Get-PhysicalDisk | where { $_.FriendlyName.contains("NVMe Direct Disk")}
```

You can run the following command for Linux:

```
sudo nvme id-ns /dev/nvme0n1 -b | dd bs=1 skip=384 status=none | sed 's/\x00*$//'
```

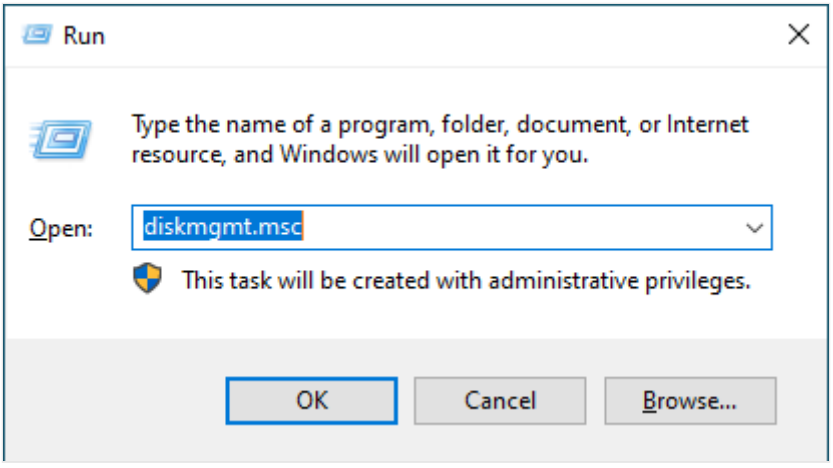
How can I format and initialize temp NVMe disks in Windows when I create a VM?

You can use the GUI or Azure PowerShell.

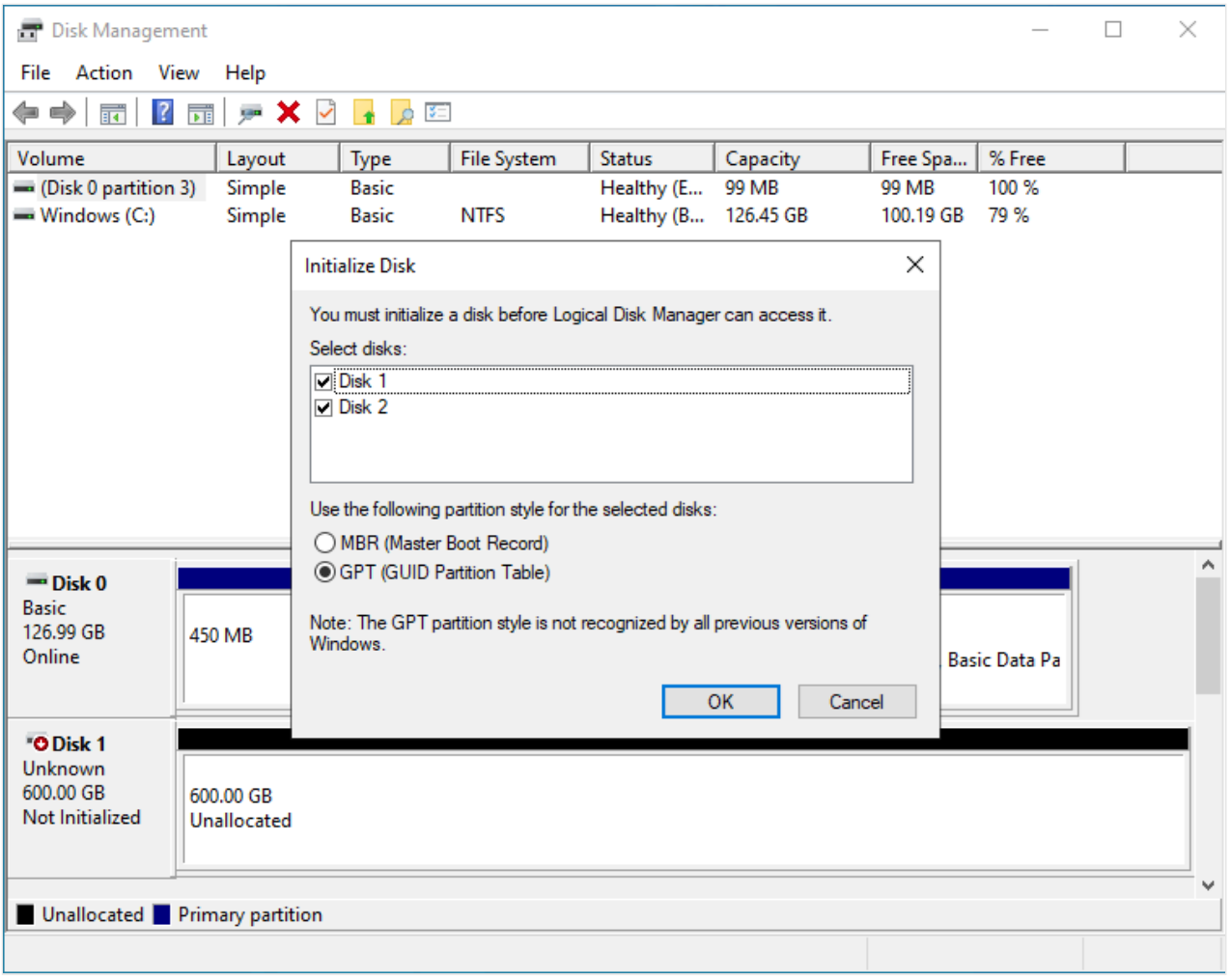
GUI

> **Tip**
>
> The following GUI example is meant for illustrative purposes. We recommend that you create scripts to automate this workflow for production deployments.

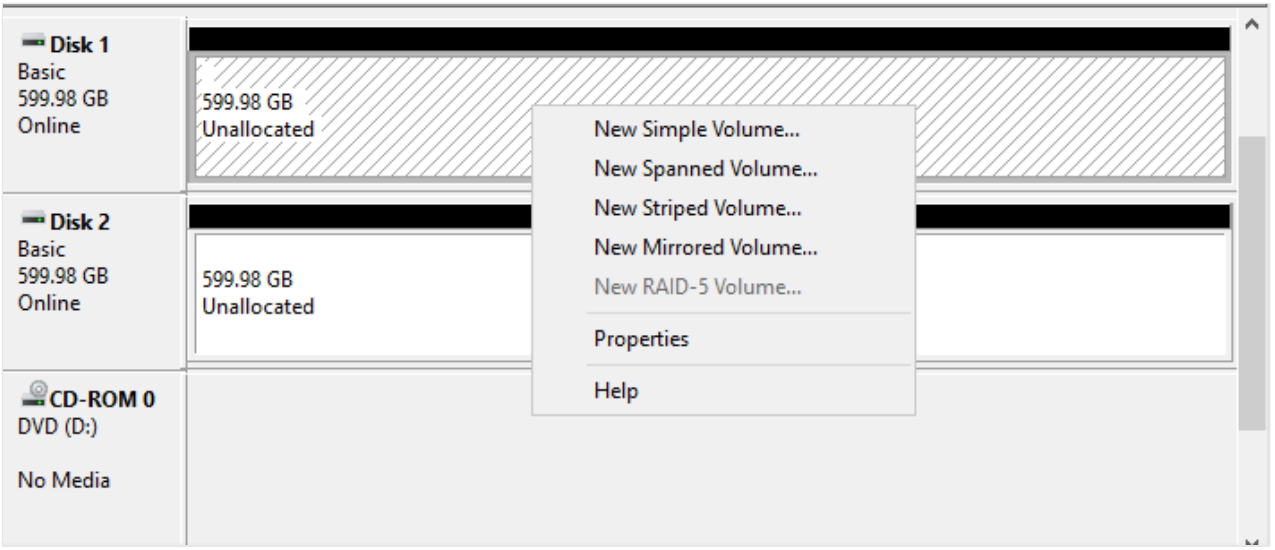
1. Select Windows logo key + R to open the **Run** dialog. Type **diskmgmt.msc** and select the Enter key.



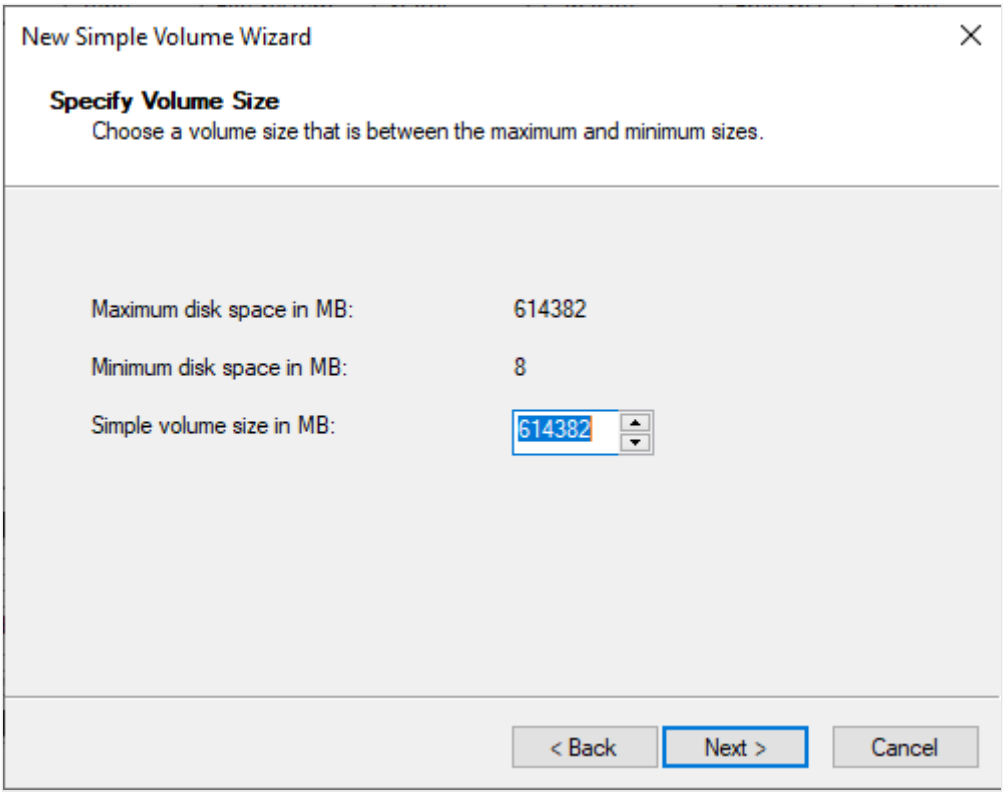
2. Select a partition style (**MBR** or **GPT**) and the disks to be initialized, and then select **OK**.

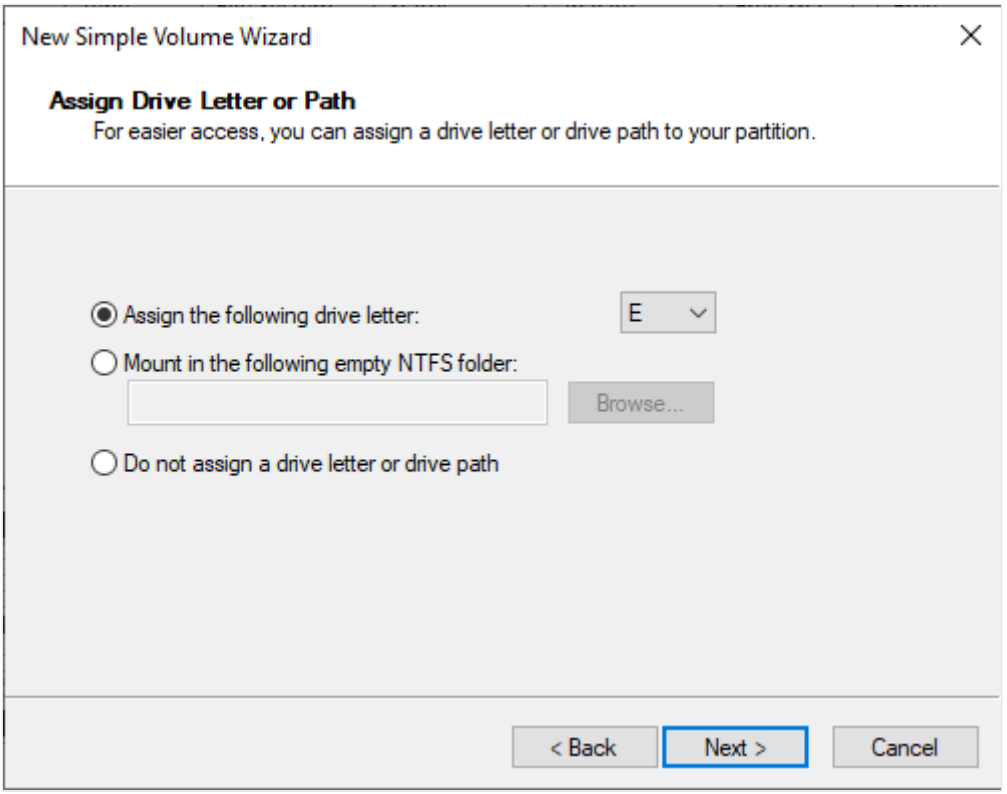


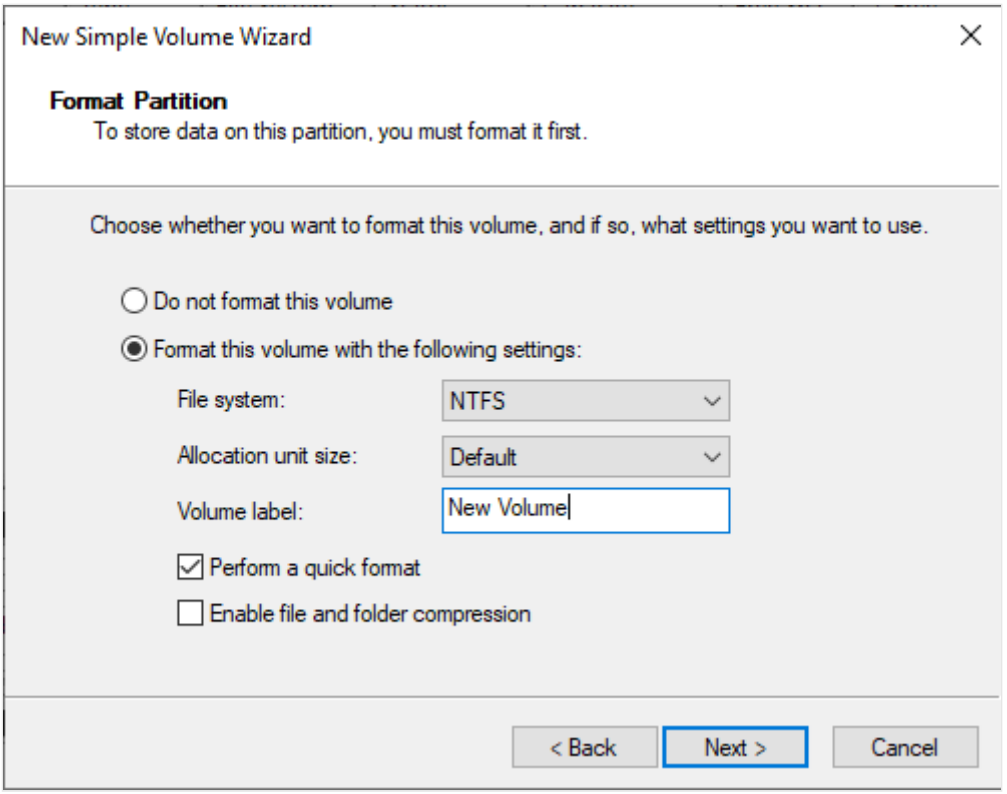
3. Right-click the disk to be partitioned, and then select **New Simple Volume**.



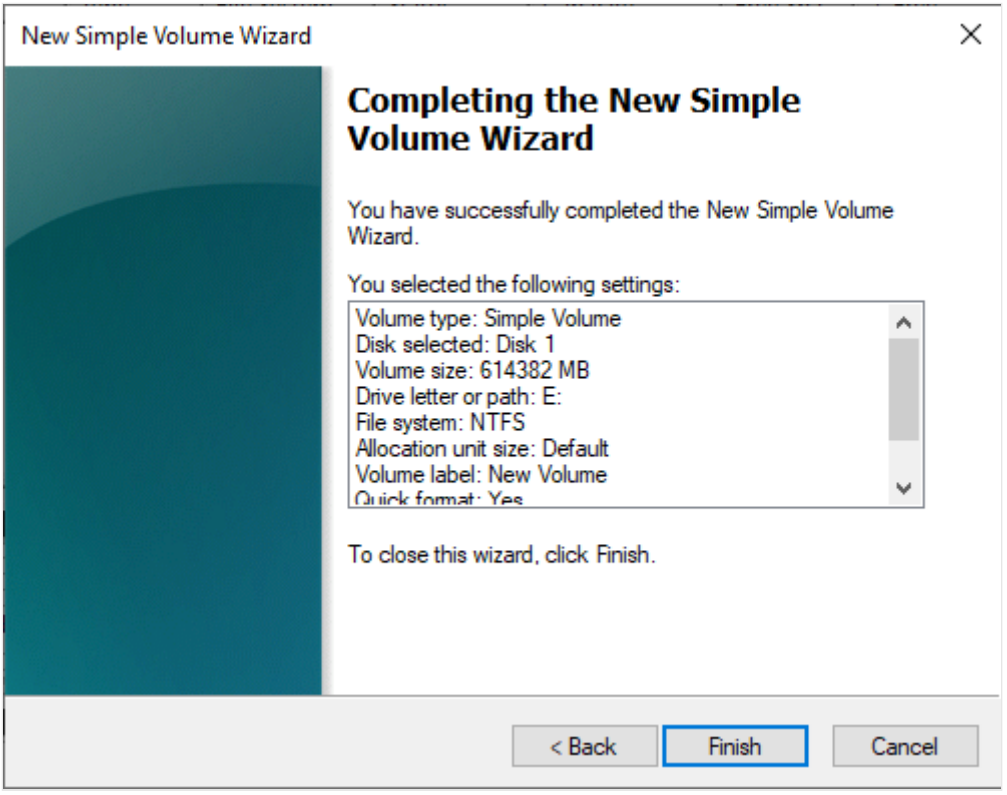
4. To specify items such as volume size, drive letter, file system, and volume label, follow the New Simple Volume Wizard. Select the **Next** button to advance through the wizard.


New Simple Volume Wizard
Specify Volume Size
Choose a volume size that is between the maximum and minimum sizes.
Maximum disk space in MB:
614382
Minimum disk space in MB:
8
Simple volume size in MB:
614382
< Back
Next >
Cancel

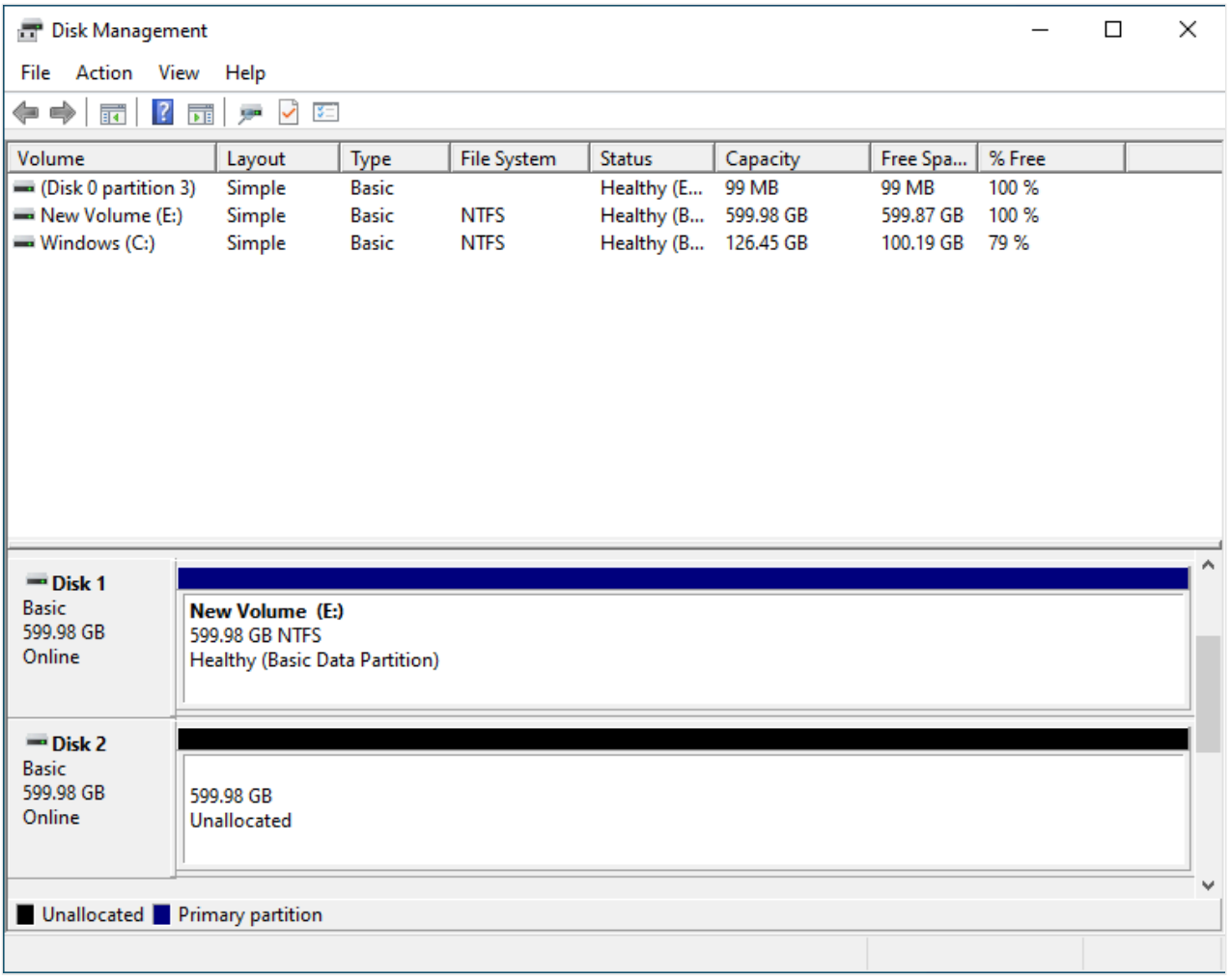

New Simple Volume Wizard
Assign Drive Letter or Path
For easier access, you can assign a drive letter or drive path to your partition.
Assign the following drive letter:
E
Mount in the following empty NTFS folder:
Browse...
Do not assign a drive letter or drive path
< Back
Next >
Cancel



5. To complete the partition and format, review your settings and then select **Finish**.



6. Confirm that the formatted and initialized NVMe disks appear in the Windows Disk Management tool, similar to how **New Volume (E:)** appears in this example.



Azure PowerShell script

> **Tip**
>
> The following script initializes, formats, and assigns drive letters. However, it erases all data on the disk. You should thoroughly test it on your VMs before deploying it in production.

PowerShell

```
# Select the raw NVMe disks to partition and format

$RawNvmeDisks = Get-PhysicalDisk -CanPool $True | where {
$_.FriendlyName.contains("NVMe Direct Disk")}

# Create a pool of the existing disks

New-StoragePool -FriendlyName NVMePool -StorageSubsystemFriendlyName "Windows
Storage*" -PhysicalDisks $RawNvmeDisks -ResiliencySettingNameDefault Simple

#Create a new disk, initialize, partition, and format

$Disk = New-VirtualDisk -FriendlyName NVMeTemporary -StoragePoolFriendlyName
NVMePool -NumberOfColumns @($RawNvmeDisks).count  -PhysicalDiskRedundancy 0 -
```

```
ResiliencySettingName "Simple" -UseMaximumSize

$Disk | Initialize-Disk

#Create a partition and format. Ignore the pop-up.

New-Partition -DiskId $Disk.UniqueId  -AssignDriveLetter -UseMaximumSize | Format-
Volume
```

How can I format and initialize temp NVMe disks in Linux?

The following script provides an example of initializing temp NVMe disks in Linux. The script enumerates all temp NVMe disks on the VM, combines the disks into a single RAID 0 array, and creates a formatted partition of the specified file system.

Temp NVMe disks are distinguished by the model number, which appears as `Microsoft NVMe Direct Disk` to the guest OS. This script uses the `nvme-cli id-ctrl` command to query the model number of each NVMe device.

The script first checks for any already-created temp NVMe disk volumes (identified by the `filesystem` label). If an array appears to be present, the script verifies the array's integrity and repeats the initialization process if necessary. The array reinitializes in the following cases:

- One or more array temp disks are missing.
- There are temp disks not already incorporated into the array.
- The disk array is in an otherwise invalid or unknown state.
- Associated configuration files are missing or invalid.

```
#!/bin/bash

# Script requirements:
#   nvme-cli
#   mdadm
#   gdisk

readonly USAGE="Usage: $(basename "$0") <filesystem> <filesystem mount point
(optional)>"

# Label used to identify the NVMe array file system and associated disks
# Can't exceed 16 characters
readonly RAID0_FILESYSTEM_LABEL="azure_nvme_temp"
# Device path used for the RAID 0 NVMe array
# Choose any unoccupied device path of format /dev/mdX (X = 0 to 99)
readonly RAID0_DEVICE_PATH="/dev/md0"
```

```
# Formatted RAID 0 partition is mounted here
readonly DEFAULT_MOUNT_POINT="/mnt/${RAID0_FILESYSTEM_LABEL}"

filesystem="$1"
if [ ! "$filesystem" ]; then
    printf "No filesystem specified. Usage: $USAGE\n"
    exit 1
fi
if ! [ -x "$(command -v mkfs.$filesystem)" ]; then
    printf "Filesystem \"$filesystem\" not supported by mkfs\n$USAGE\n"
    exit 1
fi

mount_point="$2"
if [ ! "$mount_point" ]; then
    printf "No mount point specified. Using default: $DEFAULT_MOUNT_POINT\n"
    mount_point=$DEFAULT_MOUNT_POINT
fi

# Make sure mdadm.conf is present
mdadm_conf_path=""
if [ -e "/etc/mdadm/mdadm.conf" ]; then
    mdadm_conf_path="/etc/mdadm/mdadm.conf"
elif [ -e "/etc/mdadm.conf" ]; then
    mdadm_conf_path="/etc/mdadm.conf"
else
    printf "Couldn't find mdadm.conf file"
    exit 1
fi

# Enumerate unmounted NVMe direct disks
devices=$(lsblk -p -o NAME,TYPE,MOUNTPOINT | grep "nvme" | awk '$2 == "disk" && $3
== "" {print $1}')
nvme_direct_disks=()
for device in $devices
do
    if nvme id-ctrl "$device" | grep -q "Microsoft NVMe Direct Disk"; then
        nvme_direct_disks+=("$device")
    fi
done
nvme_direct_disk_count=${#nvme_direct_disks[@]}
printf "Found $nvme_direct_disk_count NVMe Direct Disks\n"

# Check if there's already an NVMe Direct Disk RAID 0 disk (or remnant data)
if grep "$RAID0_FILESYSTEM_LABEL" /etc/fstab > /dev/null; then
    fstab_entry_present=true
fi
if grep "$RAID0_FILESYSTEM_LABEL" $mdadm_conf_path > /dev/null; then
    mdadm_conf_entry_present=true
fi
if [ -e $RAID0_DEVICE_PATH ]; then
    nvme_raid0_present=true
fi
if [ "$fstab_entry_present" = true ] || [ "$mdadm_conf_entry_present" = true ] || [
"$nvme_raid0_present" = true ]; then
```

```
    # Check if the RAID 0 volume and associated configurations are still intact or
need to be reinitialized
    #
    # If reinitialization is needed, clear the old RAID 0 information and associated
files

    reinit_raid0=false
    if [ "$fstab_entry_present" = true ] && [ "$mdadm_conf_entry_present" = true ]
&& [ "$nvme_raid0_present" = true ]; then
        # Check RAID 0 device status
        if ! mdadm --detail --test $RAID0_DEVICE_PATH &> /dev/null; then
            reinit_raid0=true
        # Test the NVMe direct disks for valid mdadm superblocks
        else
            for device in "${nvme_direct_disks[@]}"
            do
                if ! mdadm --examine $device &> /dev/null; then
                    reinit_raid0=true
                    break
                fi
            done
        fi
    else
        reinit_raid0=true
    fi

    if [ "$reinit_raid0" = true ]; then
        echo "Errors found in NVMe RAID 0 temp array device or configuration.
Reinitializing."

        # Remove the file system and partition table, and stop the RAID 0 array
        if [ "$nvme_raid0_present" = true ]; then
            if [ -e ${RAID0_DEVICE_PATH}p1 ]; then
                umount ${RAID0_DEVICE_PATH}p1
                wipefs -a -f ${RAID0_DEVICE_PATH}p1
            fi
            sgdisk -o $RAID0_DEVICE_PATH &> /dev/null
            mdadm --stop $RAID0_DEVICE_PATH
        fi

        # Remove any mdadm metadata from all NVMe Direct Disks
        for device in "${nvme_direct_disks[@]}"
        do
            printf "Clearing mdadm superblock from $device\n"
            mdadm --zero-superblock $device &> /dev/null
        done

        # Remove any associated entries in fstab and mdadm.conf
        sed -i.bak "/$RAID0_FILESYSTEM_LABEL/d" /etc/fstab
        sed -i.bak "/$RAID0_FILESYSTEM_LABEL/d" $mdadm_conf_path
    else
        printf "Valid NVMe RAID 0 array present and no additional Direct Disks
found. Skipping\n"
        exit 0
    fi
```

```
fi

if [ "$nvme_direct_disk_count" -eq 0 ]; then
    printf "No NVMe Direct Disks found\n"
    exit 1
elif [ "$nvme_direct_disk_count" -eq 1 ]; then
    additional_mdadm_params="--force"
fi

# Initialize enumerated disks as RAID 0
printf "Creating RAID 0 array from:\n"
printf "${nvme_direct_disks[*]}\n\n"
if ! mdadm --create $RAID0_DEVICE_PATH --verbose $additional_mdadm_params --
name=$RAID0_FILESYSTEM_LABEL --level=0 --raid-devices=$nvme_direct_disk_count
${nvme_direct_disks[*]}; then
    printf "Failed to create RAID 0 array\n"
    exit 1
fi

# Create a GPT partition entry
readonly GPT_PARTITION_TYPE_GUID="0FC63DAF-8483-4772-8E79-3D69D8477DE4"
printf "\nCreating GPT on $RAID0_DEVICE_PATH..\n"
sgdisk -o $RAID0_DEVICE_PATH &> /dev/null
if ! sgdisk --new 1::0 --typecode 1:$GPT_PARTITION_TYPE_GUID $RAID0_DEVICE_PATH  &>
/dev/null; then
    printf "Failed to create partition on $RAID0_DEVICE_PATH\n"
    exit 1
fi

# Format the partition
partition_path="${RAID0_DEVICE_PATH}p1"
printf "\nCreating $filesystem filesystem..\n"
if ! mkfs.$filesystem -q -L $RAID0_FILESYSTEM_LABEL $partition_path; then
    printf "Failed to create $filesystem filesystem\n"
    exit 1
fi
printf "The operation has completed successfully.\n"

# Add the partition to /etc/fstab
echo "LABEL=$RAID0_FILESYSTEM_LABEL $mount_point $filesystem defaults,nofail 0 0" >>
/etc/fstab

# Add RAID 0 array to mdadm.conf
mdadm --detail --scan >> $mdadm_conf_path
update-initramfs -u

# Mount the partition
printf "\nMounting filesystem to $mount_point..\n"
mkdir $mount_point &> /dev/null
if ! mount -a; then
    printf "Failed to automount partition\n"
    exit 1
fi
printf "The operation has completed successfully.\n"
```

```
exit 0
```

For RHEL/CentOS based distributions, replace `update-initramfs -u` to `dracut --force` to regenerate initramfs file. If script stops execution with "Couldn't find mdadm.conf file", then you can create mdadm.conf file using the following command:

```
sudo touch /etc/mdadm/mdadm.conf
```

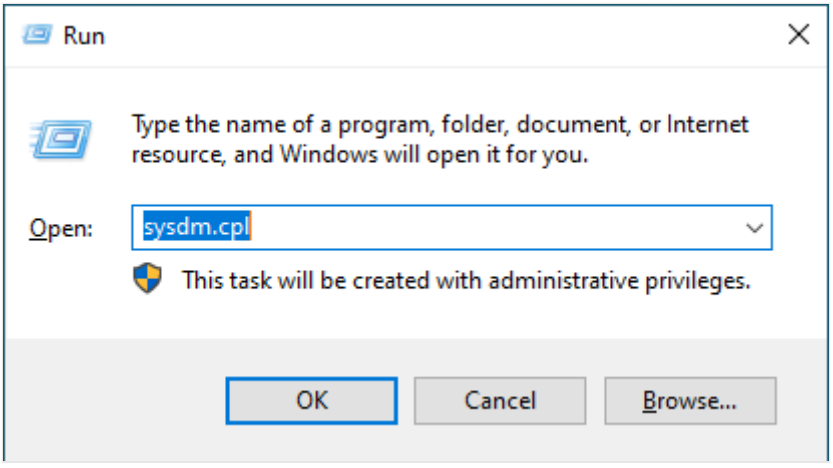
How can I move the Windows pagefile from an OS disk to a temp NVMe disk?

You can use the GUI or Azure PowerShell.

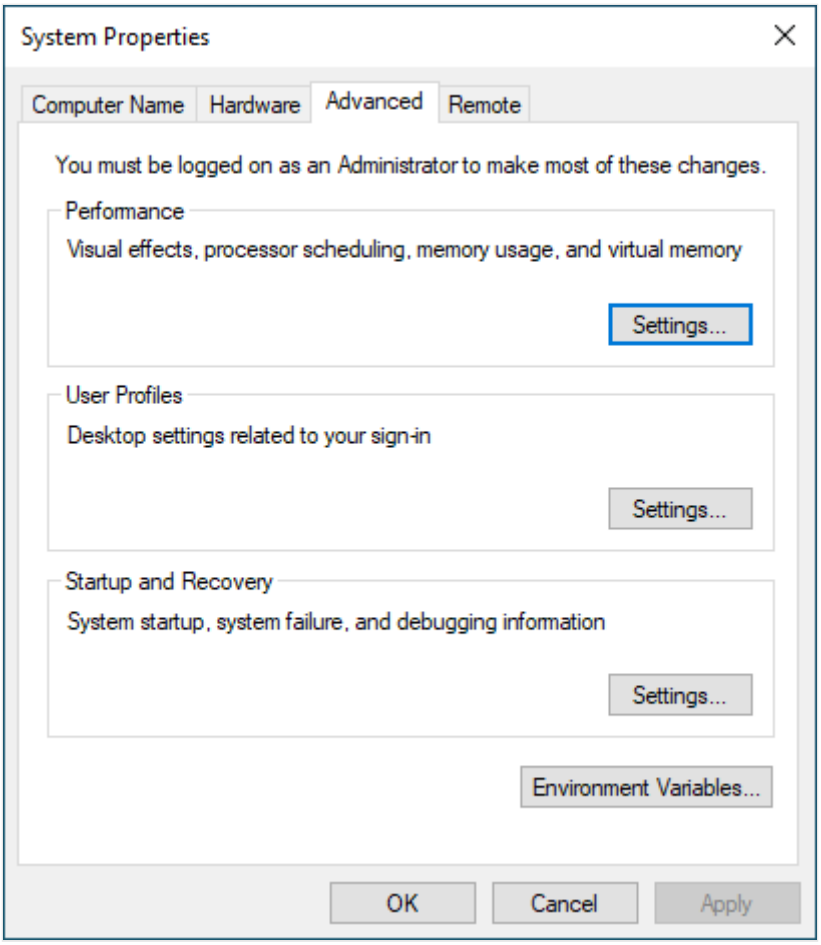
GUI

> **Tip**
>
> The following GUI example is meant for illustrative purposes. We recommend that you create scripts to automate this workflow for production deployments.

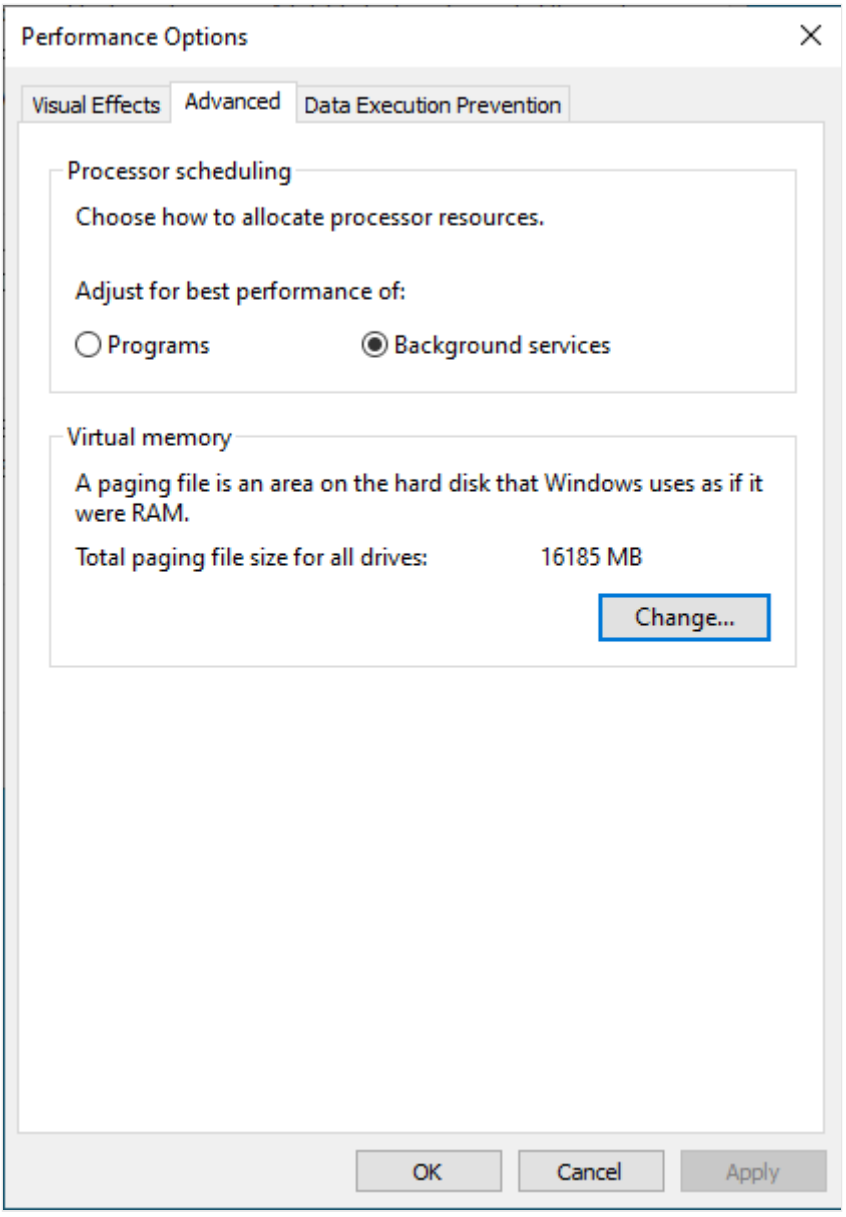
1. Select Windows logo key + R to open the **Run** dialog. Type **sysdm.cpl** and select the Enter key.



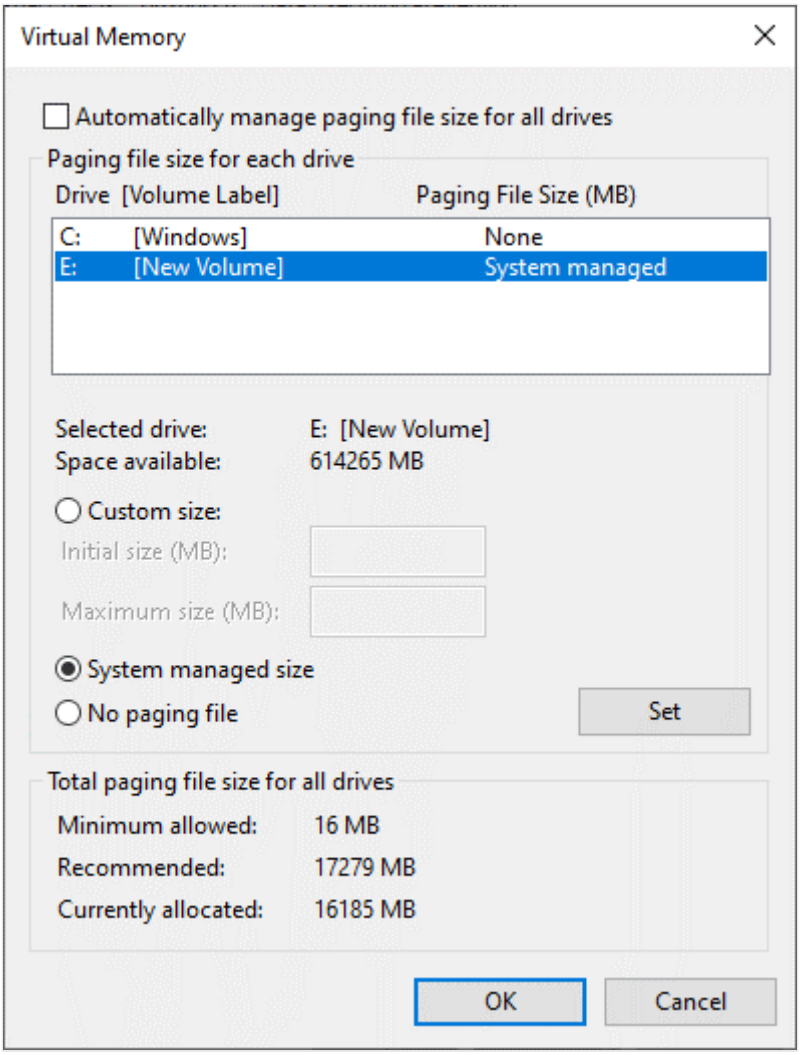
2. Select the **Advanced** tab, and then select the **Settings** button in the **Performance** section.



3. In the **Performance Options** dialog, select the **Advanced** tab, and then select the **Change** button in the **Virtual memory** section.



4. Clear the **Automatically manage paging file size for all drives** checkbox. If a pagefile is set to your OS disk, select your OS disk, select the **No paging file** option, and then select the **Set** button. Select your local temp NVMe drive, select the **System managed size** option, and then select the **Set** button. Select **OK** to close all dialogs.



You might need to restart your VM for these changes to take effect.

Azure PowerShell script

PowerShell

```
$OsDisk = "C:"
# This value can vary, depending on which drive letter is assigned to the disk
$NVMeDisk = "E:"

# Disable automatic pagefile management
$Computer = Get-WmiObject Win32_computersystem -EnableAllPrivileges
$Computer.AutomaticManagedPagefile = $false
$Computer.Put()

# Delete the pagefile on the OS disk
$PageFile = Get-WmiObject -Query "select * from Win32_PageFileSetting where
name='$OsDisk\\pagefile.sys'"
$PageFile.Delete()

# Create a new pagefile on the NVMe drive with system-managed size
Set-WMIInstance -Class Win32_PageFileSetting -Arguments
@{name="$NVMeDisk\\pagefile.sys"; InitialSize=0; MaximumSize=0} -EnableAllPrivileges
```

You might need to restart your VM for these changes to take effect.

How can I move the Linux swap file from an OS disk to a temp NVMe disk?

1. Select the NVMe disk used for swap space:

```
root@D2adsv6:/ # lsblk -p
    NAME              MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
    /dev/nvme1n1      259:0    0   30G  0 disk
    ├─/dev/nvme1n1p1  259:1    0 29.9G  0 part /
    └─/dev/nvme1n1p15 259:3    0  106M  0 part /boot/efi
    /dev/nvme0n1      259:4    0   75G  0 disk
```

2. Create swap space on the disk:

```
root@D2adsv6:/ # mkswap /dev/nvme0n1
Setting up swapspace version 1, size = 75 GiB (80530632704 bytes)
no label, UUID=064bdcfb-86ae-49f3-bf9d-b956493e2a1d
```

3. Enable the swap space:

```
root@D2adsv6:/ # swapon /dev/nvme0n1
```

4. Check that the swap space is set up properly:

```
root@D2adsv6:/ # swapon -s
Filename                                Type            Size    Used
Priority
/dev/nvme0n1                            partition       78643196        0
-2
```

5. Append the swap space to `/etc/fstab` to make it persistent across restarts:

```
root@D2adsv6:/ # echo '/dev/nvme0n1 swap swap defaults 0 0' >> /etc/fstab
```

What considerations do I need to be aware of for maintenance events, VM redeployments, and VM restarts?

Temp NVMe disks attached to V6 VMs are ephemeral, similar to temp drives on other VM series. This means that you lose all data on the VM after redeployment or during a maintenance event. For more information on maintenance events and downtime, see Understand VM reboots: Maintenance vs. downtime.

If the VM moved to new hardware, these drives unmount, and new unmounted disks are presented to the operating system when it comes back up. If the VM didn't change hardware, the temp NVMe disks might still be present. The preceding script for mounting the drives checks for unmounted drives before it tries to mount.

It's a best practice to run the mounting script automatically every time the VM starts and delay any other startup scripts that need the drives until after the mounting script finishes running.

Why are my temporary disk write speeds different than published specifications?

Temporary disk performance is influenced by several factors, such as block size, workload read/write patterns, queue depth (QD). Published performance specifications represent best-case scenarios, based on 4k random block sizes with multiple workers (≥8) at QD≥64 per worker for peak IOPS, and large sequential block sizes (≥128k) for maximum throughput. Write performance may also be affected by data patterns, SSD state (overprovisioned capacity, percent of LBA written), and background activities such as garbage collection. Performance figures assume the device is in a fresh-out-of-box (FOB) state, which provides the highest speeds. Once the disk reaches steady state after SNIA-standard preconditioning specifications, write performance may be lower than FOB levels.

FAQ for remote NVMe disks

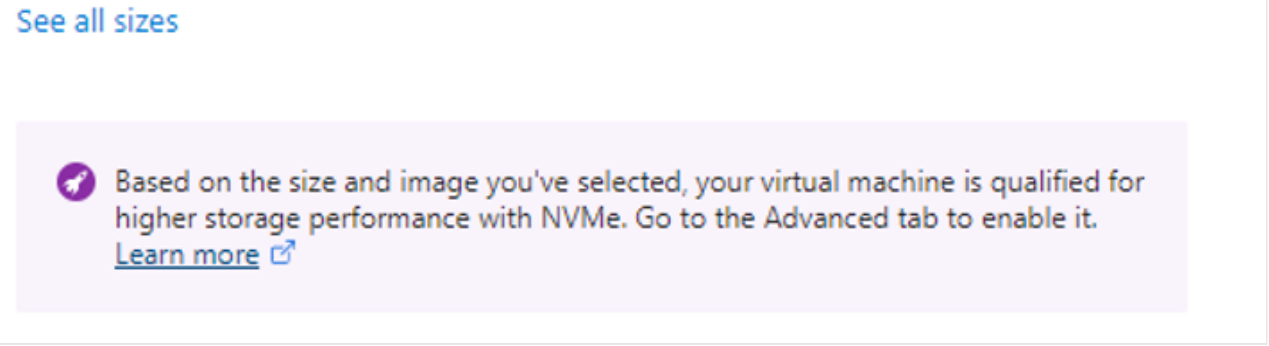
This article addresses some common questions about support for remote NVM Express (NVMe) disks on virtual machines (VMs) created in Azure.

What are the prerequisites to enable the remote NVMe interface on my VM?

The `DiskControllerTypes` is defined during VM configuration and is determined by the selected VM size as either NVMe or Small Computer System Interface (SCSI). If you don't specify a `DiskControllerTypes` value, the platform automatically chooses the default controller based on the VM size configuration.

To enable the remote NVMe interface on your VM, you must meet the following prerequisites:

- Most modern Azure VM sizes support the NVMe disk controller type for remote storage. Support begins with Ebsv5 VM sizes and later generations (v6, v7).
- Select the operating system image that's tagged with NVMe support. For VM sizes that support the NVMe interface, Azure automatically configures the NVMe disk controller type during VM creation. The NVMe setting in the Advanced tab is selected by default and cannot be changed.



- Opt in to NVMe by selecting the NVMe disk controller type in the Azure portal or in the Azure Resource Manager, Azure CLI, or Azure PowerShell template. For step-by-step instructions, refer to the general NVMe FAQ.

How can I resize a SCSI-based VM to a remote NVMe-enabled VM of a different size?

You can use the following process to either:

- Resize a SCSI-based VM created using an untagged image to an NVMe-enabled VM of a different size without re-creating the VM configurations and without tagging the image.
- Resize a SCSI-based VM to an NVMe-enabled VM of a different size without re-creating the VM configurations.

The source VM can be either:

- An untagged OS image that supports remote NVMe.
- An NVMe-tagged OS image.

To resize the VM, use the following command to run an Azure PowerShell script that sets the destination `discontrollertype` value of the VM as NVMe:

PowerShell

```
azure-nvme-VM-update.ps1 [-subscription_id] <String> [-resource_group_name]
<String> [-vm_name] <String> [[-disk_controller_change_to] <String>] [-
vm_size_change_to] <String> [[-start_vm_after_update] <Boolean>] [[-write_logfile]
<Boolean>]
```

For more details, refer to SCSI to NVMe for Linux VMs

How can I check if an image is tagged as NVMe?

To check if an image is tagged as NVMe, use the following command:

```
az vm image show --urn URN_OF_IMAGE
```

How do I create an image definition that supports NVMe for remote disks?

To create an image definition that supports NVMe for remote disks, complete the following steps:

1. Upload an NVMe-supported virtual hard disk (VHD) to your storage account. AzCopy is a fast way, but you can also use the portal to upload.

```
azcopy copy <local path to your VHD> <container in your storage account>
```

2. Create an image gallery by using Azure PowerShell, the portal, or the Azure CLI.

3. Create an image definition. Be sure to include `--feature DiskControllerTypes=SCSI,NVMe`.

   Here's an Azure CLI example:

```
az sig image-definition create --resource-group <resourceGroupName> --gallery-name <galleryName> --gallery-image-definition <imageName> --publisher <publisher> --offer <offerName> --sku <skuName> --os-type <osType> --os-state <osState> --feature DiskControllerTypes=SCSI,NVMe
```

4. Create the image version with the NVMe-supported VHD.

   Here's an Azure CLI example:

```
az sig image-version create --resource-group <resourceGroupName> --gallery-name <galleryName> --gallery-image-definition <imageName> --gallery-image-version <version> --target-regions <region1> <region2> --replica-count <replicaCount> --os-vhd-uri <NVMe-supported vhd uri> --os-vhd-storage-account <storageAccount>
```

Which Azure disk storage options are compatible with remote NVMe disks?

NVMe sizes can be combined with Azure Standard HDD, Standard SSD, Premium SSD v1, Premium SSD v2, and Ultra Disk Storage. For more information on Azure disk offerings, see Azure managed disk types.

Does Azure support live resizing on disks with NVMe VM sizes?

Live resizing of NVMe is supported on Azure Premium SSD v1 disks, Premium SSD v2 disks, Standard SSDs, and Standard HDDs. You can also add remote NVMe disks without restarting the VM.

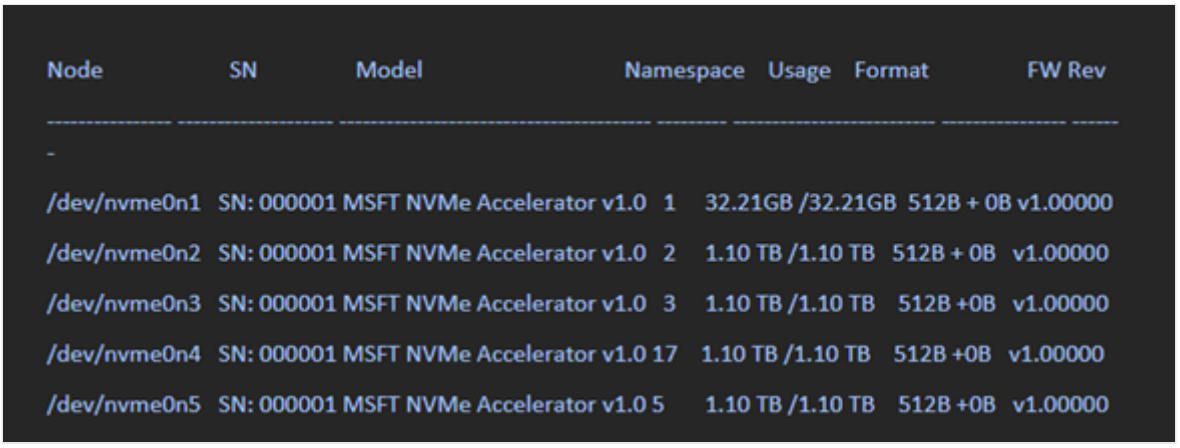
How can I identify remote NVMe disks on a Linux VM?

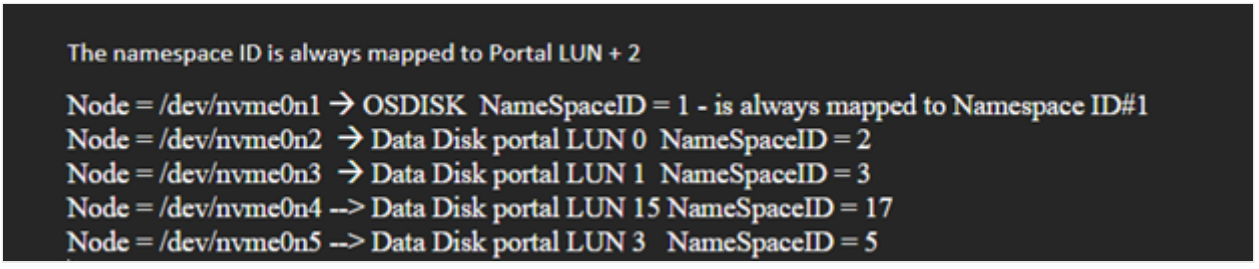
1. Get the `nvme-cli` package:

```
sudo apt install nvme-cli
```

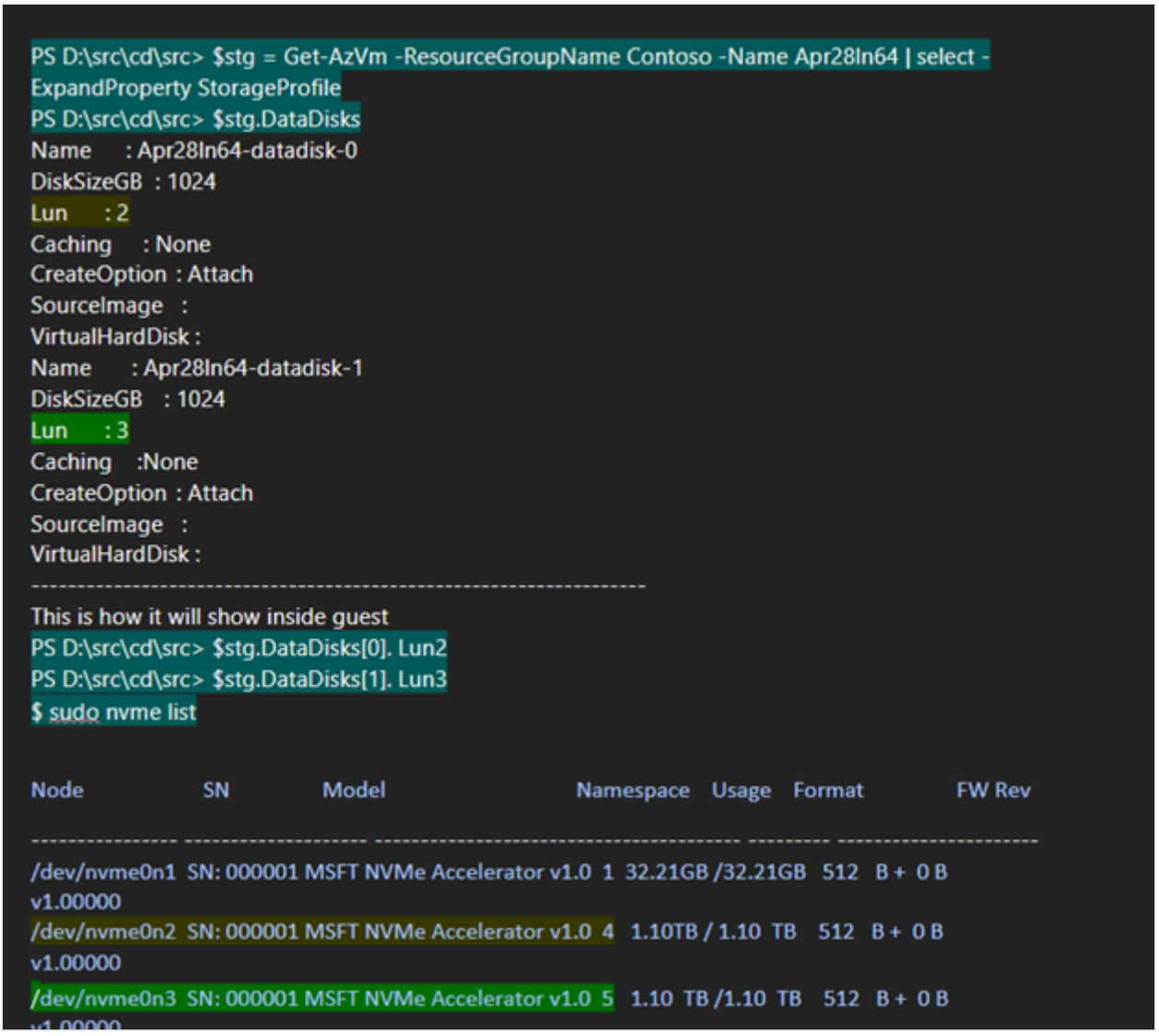
2. Run the NVMe `list` command to fetch NVMe disk details:

```
sudo nvme list
```





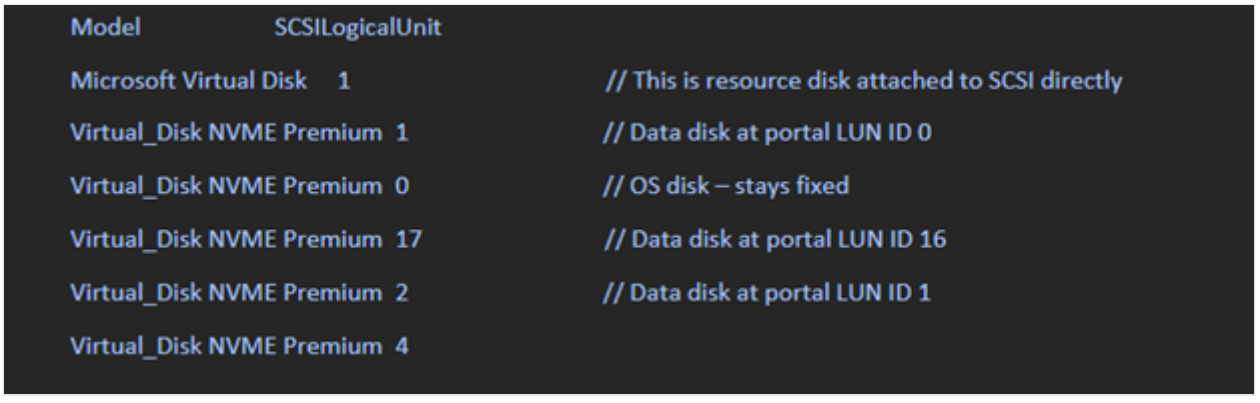
Here's how the data appears in response to Azure PowerShell commands:



How can I identify NVMe disks on a Windows VM?

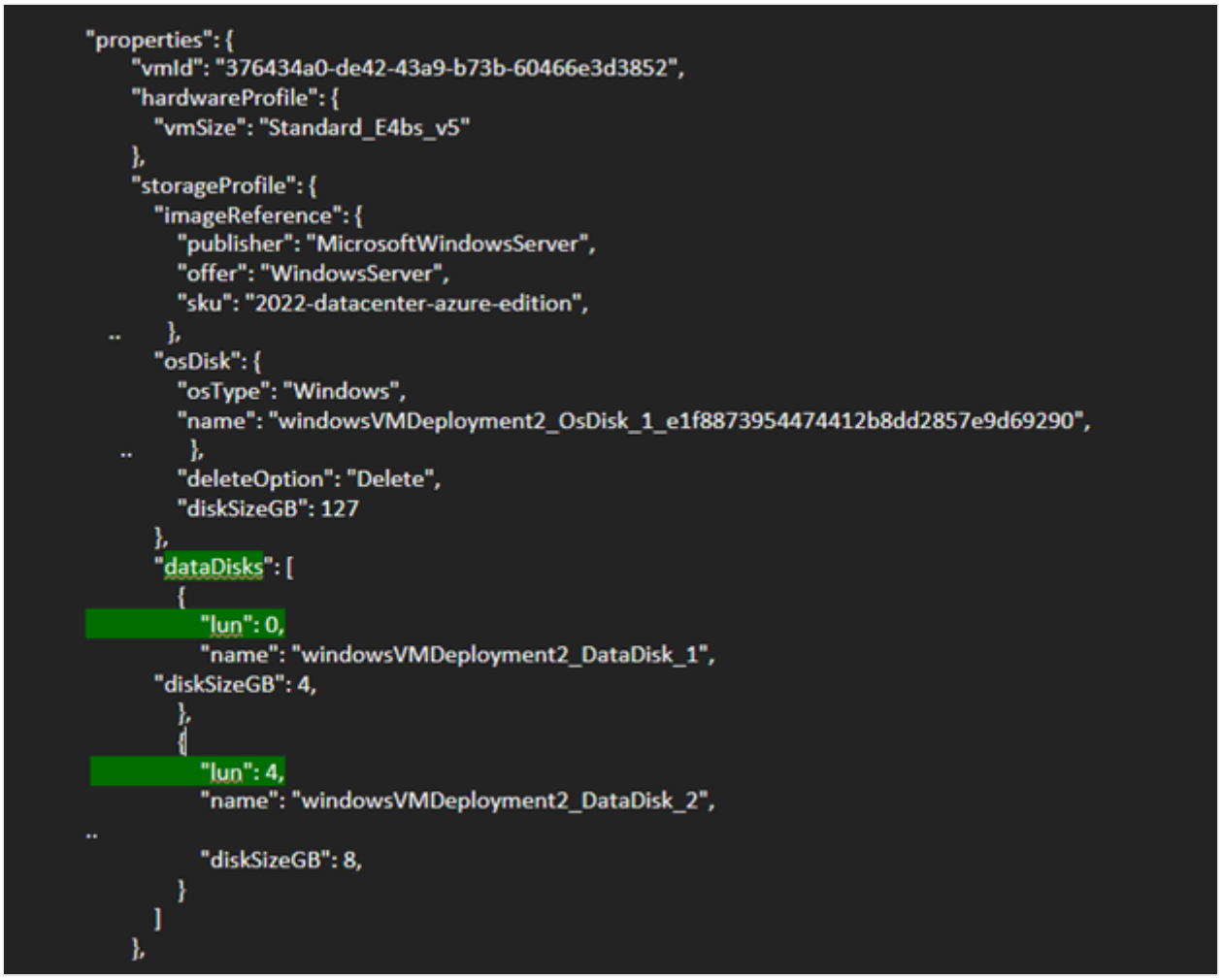
Open Azure PowerShell and use the following command:

```
wmic diskdrive get model,scsilogicalunit
```

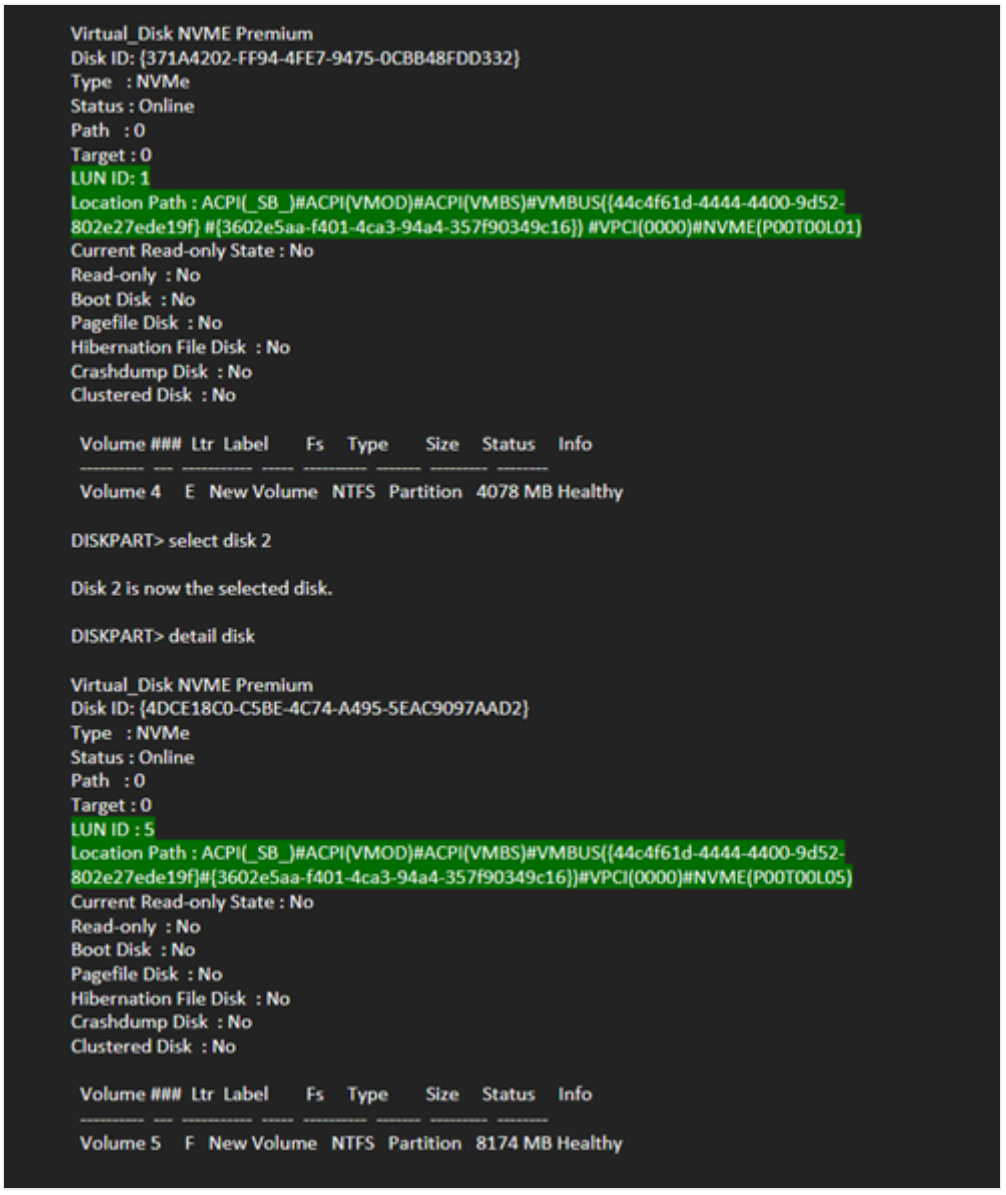


The ASAP attached disks are presented in the guest with the model string `Virtual_Disk NVMe Premium`. The SCSI logical unit has the value for the portal-visible LUN ID incremented by 1.

Here's a snapshot of how NVMe disks appear in an NVMe-enabled Windows VM:



The following snapshot shows guest output for data disks attached at LUN 0 and LUN 4 (CRP). The LUN ID is equivalent to the namespace ID.



Are shared disks in remote disks supported with NVMe-enabled VMs?

The shared disk feature is supported for Premium SSD, Premium SSD v2, and Ultra Disk Storage disks. Shared Disks using NVMe isn't supported with the OS Windows Server 2019.

Can a data disk be detached from a SCSI-based VM and then attached to an NVMe-based VM?

Yes. A data disk can be detached from a SCSI-based VM and attached to an NVMe-based VM. Once attached, the disk type will automatically convert to NVMe.

Will all my VM's disks be attached to one NVMe controller, or are they distributed across multiple controllers?

Older Azure VM types (such as Ebsv6) attach all disks to a single NVMe controller. Select newer VM sizes (V7 and above, running on Intel and ARM hardware) automatically distribute disks across multiple controllers and separate cached disks (including the OS disk) and uncached data disks for improved performance and reliability.

How are disks assigned to controllers, and what should I know about disk management?

Boot and cached data disks are assigned to the cached controller and uncached data disks go to the uncached controller. Controller assignment is automatic based on disk caching policy selected in the VM settings.

If you need to change the caching policy of a disk, it is recommended to stop the VM, change the caching policy, and then start the VM again for stable operation. This helps avoid inconsistent states or remapping issues. OS disk caching changes are non-functional and will be disabled in future updates.

How can I identify and manage disks across controllers in Linux and Windows?

> **Note**
>
> Using UUIDs (Linux) or GUIDs (Windows) ensures disks are correctly identified and remounted after VM events, upgrades, or controller changes. If device names change after reboots or disk operations, rely on UUIDs/GUIDs or persistent naming for automation and scripting.

Windows

- Open Disk Management (`diskmgmt.msc`) to see all disks and their controller assignments.

- Or use PowerShell for more details:

PowerShell

```
Get-PhysicalDisk | Select-Object DeviceID, FriendlyName, SerialNumber, BusType
```

This lists all physical disks, including NVMe devices and their properties.

Supported OS images for remote NVMe

The following lists provide up-to-date information on which OS images are tagged as supported for remote NVM Express (NVMe).

> **Important**
>
> Only Gen2 VM images with security type "Standard" or "Trusted Launch" support NVMe. For more information, see **FAQ for NVMe**. Older operating systems cannot support NVMe due to OS driver limitations. If an older version of the OS is not included in the list below, it means that NVMe support is not present for that OS.

For specifics about which virtual machine (VM) generations support which storage types, check the documentation about VM sizes in Azure.

For more information about enabling the NVMe interface on virtual machines created in Azure, review the FAQ for remote NVMe disks.

Supported Linux OS images

> **Note**
>
> For Linux virtual machines using NVMe-attached storage, Microsoft recommends setting the kernel parameter `nvme_core.io_timeout` to `240` seconds. This setting effectively disables OS-level NVMe IO timeouts, ensuring that Azure's host-level timeout mechanism takes precedence to handle disk failures or interruptions. Some older Linux images may have the default `io_timeout` set to 30 seconds, which can cause the OS to timeout IOs before Azure can intervene.

- Azure Linux 3.0
- Almalinux 8.x
- Almalinux 9.x
- Debian 11
- Debian 12
- Oracle Linux 7.9
- Oracle Linux 8.5 and newer
- Oracle Linux 9.0 and newer
- RHEL 7.9
- RHEL 8.6 and newer
- RHEL 9.0 and newer

- SLES 15 SP4 and newer
- SLES-for-SAP 15 SP3 and newer
- Ubuntu 18.04
- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04
- Rocky Linux 8.10
- Rocky Linux 9.6

Supported Windows OS images

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019*
- Windows 10
- Windows 11

*Shared Disks on NVME are not compatible with Windows Server 2019

To download an image, go to Azure Marketplace .

Last updated on 01/23/2026

Converting Virtual Machines Running Linux from SCSI to NVMe

07/16/2025

In this article, we discuss the process of converting virtual machines (VM) running Linux from SCSI to NVMe storage. By migrating to NVMe, you can take advantage of its improved performance and scalability.

SCSI vs NVMe

Azure VMs support two types of storage interfaces: Small Computer System Interface (SCSI) and NVMe. The SCSI interface is a legacy standard that provides physical connectivity and data transfer between computers and peripheral devices. NVMe is similar to SCSI in that it provides connectivity and data transfer, but NVMe is a faster and more efficient interface for data transfer between servers and storage systems.

Support for SCSI interface VMs

Azure continues to support the SCSI interface on the versions of VM offerings that provide SCSI storage. However, not all new VM series have SCSI storage as an option going forward.

What is changing for your VM?

Changing the host interface from SCSI to NVMe doesn't change the remote storage (OS disk or data disks), but change the way the operating systems sees the disks.

Expand table

Disk	SCSI enabled VM	NVMe VM with SCSI tempdisk (e.g. Ebds_v5)	NVMe VM with NVMe tempdisk
OS disk	/dev/sda	/dev/nvme0n1	/dev/nvme0n1
Temp Disk	/dev/sdb	/dev/sda	/dev/nvme1n1
First Data Disk	/dev/sdc	/dev/nvme0n2	/dev/nvme0n2

Tip

Some VM types have more than one temporary disk (e.g. E64ds_v6)

In the following sections, we provide a guide to convert your Azure VM from SCSI to NVMe using Azure Boost ensuring you can take full advantage of these performance improvements and maintain a competitive edge in the cloud computing landscape.

Migrate your virtual machine (VM) from SCSI to NVMe

In order to migrate from SCSI to NVMe, some steps need to be followed:

1. Check if your virtual machine series supports NVMe
2. Check your operating system for NVMe readiness
3. Convert your virtual machine to NVMe
4. Check your operating system

1. Check if your virtual machine series supports NVMe

The supported virtual machines to support NVMe attached disks is described on the Azure Boost overview site in the availability table.

2. Check your operating system for NVMe readiness

The operating system needs to support NVMe devices, includes for example, device drivers and initrdm, the temporary file system used during boot, need to be prepared. In addition to that you need to validate the mount points of the file systems as they check if you use the SCSI device name (/dev/sdX).

The migration script automatically can take care of these readiness checks for you when using the `-FixOperatingSystemSettings`.

2.1 Check Controller Type of VM

2.1.1 Check Controller Type using PowerShell

PowerShell

```
PS C:\Users\user1> $vm = Get-AzVM -name [your-vm-name]
PS C:\Users\user1> $vm.StorageProfile.DiskControllerType
```

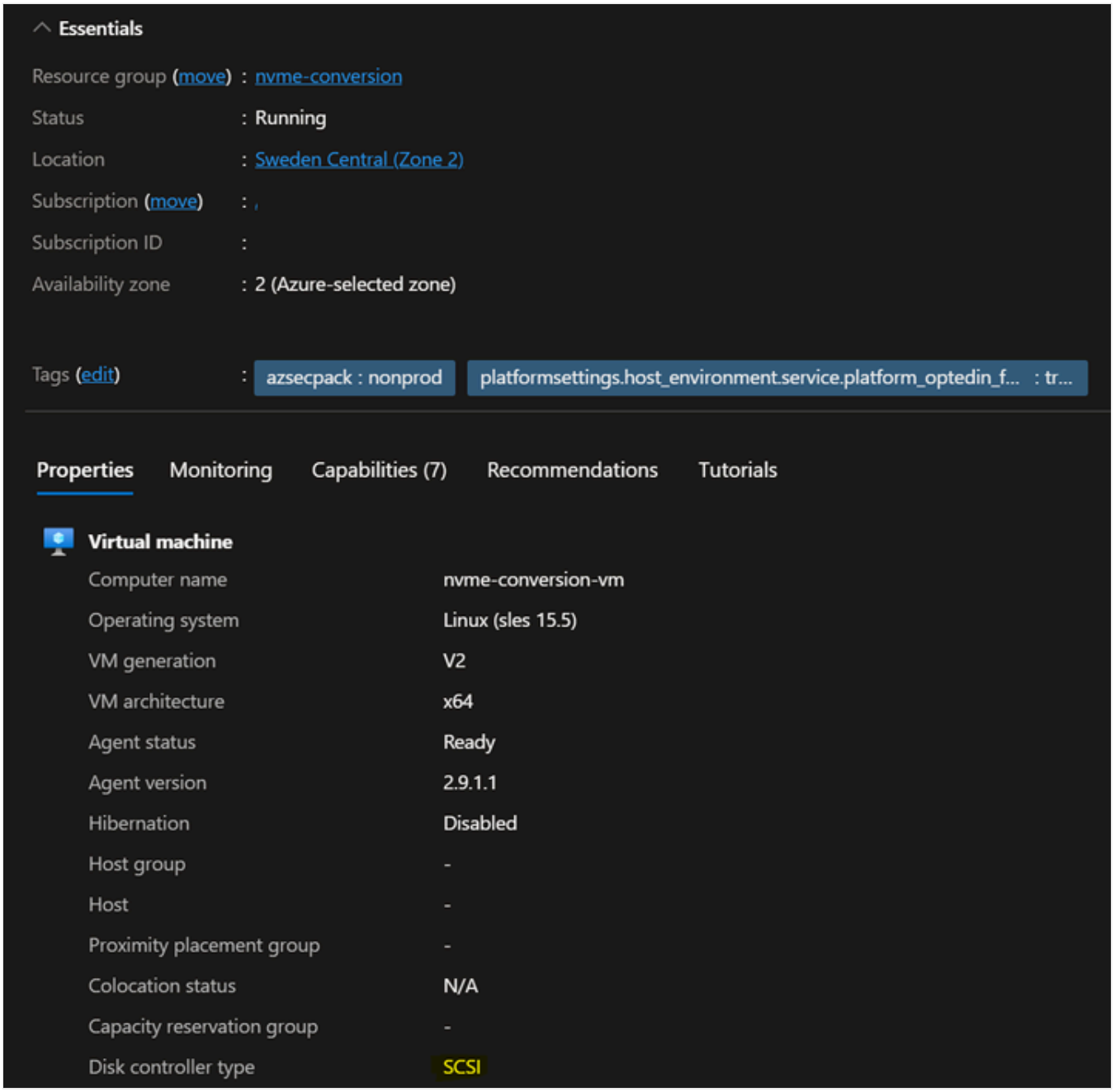
```
SCSI
PS C:\Users\user1>
```

2.1.2 Check Controller Type using Azure CLI

PowerShell

```
$ az vm show --name [your-vm-name] --resource-group [your-resource-group-name]
{
"additionalCapabilities": {
...
 "storageProfile": {
 ...
   "diskControllerType": "SCSI",
 ...
```

2.1.3 Check Controller Type using Azure portal



2.2 Prepare for migration

The Migration script can automatically take care about the prerequisites when using the `-FixOperatingSystemSettings` parameter.

If you want to manually take care of the required changes validate

- NVMe modules installed and part of initrd/initramfs
- GRUB configuration includes the parameter nvme_core.io_timeout=240
- /etc/fstab checks for devices

Please check back with your OS vendor to cover all required commands to update initrd/initramfs.

2.2.1 Prepare PowerShell

> **Tip**
>
> This step is not required when running the script in Azure CloudShell

1. Install PowerShell using https://aka.ms/powershell
2. Connect to Azure using `Connect-AzAccount` and select the correct subscription using `Select-AzSubscription -Subscription [your-subscription-id]`
3. Set the Execution Policy using `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`

2.2.2 Download the script

You can download the script using a PowerShell command

PowerShell

```
Invoke-WebRequest -Uri "https://raw.githubusercontent.com/Azure/SAP-on-Azure-Scripts-and-Utilities/refs/heads/main/Azure-NVMe-Utils/Azure-NVMe-Conversion.ps1" -OutFile ".\NVMe-Conversion.ps1"
```

2.3 Run the migration

The script has multiple parameters available:

Expand table

Parameter	Description	Required
`-ResourceGroupName`	The Resource Group Name of your VM	Yes
`-VMName`	The name of your Virtual Machine on Azure	Yes
`-NewControllerType`	The storage controller type the VM should get converted to (NVMe or SCSI)	Yes
`-VMSize`	Azure VM SKU you want to convert the VM to	Yes
`-StartVM`	Start the VM after conversion	No
`-IgnoreSKUCheck`	Ignore the check of the VM SKU	No
`-IgnoreWindowsVersionCheck`	Ignore the Windows Version check	No

Parameter	Description	Required
`-FixOperatingSystemSettings`	Automatically fix the OS settings using Azure RunCommands	No
`-WriteLogfile`	Create a Log File	No
`-IgnoreAzureModuleCheck`	Do not run the check for installed Azure modules	No
`-IgnoreOSCheck`	Do not check for OS readiness, expectation is that the OS is ready	No
`-SleepSeconds`	Time for Azure to settle changes before starting the VM	No

Sample Command:

PowerShell

```
# Example usage
.\Azure-NVMe-Conversion.ps1 -ResourceGroupName <your-RG> -VMName <your-VMname> -
NewControllerType <NVMe/SCSI> -VMSize <new-VM-SKU> -StartVM -
FixOperatingSystemSettings
```

> **Tip**
>
> You can always revert back to SCSI, the script will share a command with you to directly revert to your original configuration.

2.3.1 Sample output

PowerShell

```
PS /home/philipp> ./NVMe-Conversion.ps1 -ResourceGroupName testrg -VMName testvm -
NewControllerType NVMe -VMSize Standard_E4bds_v5 -StartVM -
FixOperatingSystemSettings
00:00 - INFO      - Starting script Azure-NVMe-Conversion.ps1
00:00 - INFO      - Script started at 06/27/2025 15:41:39
00:00 - INFO      - Script version: 2025062704
00:00 - INFO      - Script parameters:
00:00 - INFO      -   ResourceGroupName -> testrg
00:00 - INFO      -   VMName -> testvm
00:00 - INFO      -   NewControllerType -> NVMe
00:00 - INFO      -   VMSize -> Standard_E4bds_v5
00:00 - INFO      -   StartVM -> True
00:00 - INFO      -   FixOperatingSystemSettings -> True
00:00 - INFO      - Script Version 2025062704
00:00 - INFO      - Module Az.Compute is installed and the version is correct.
00:00 - INFO      - Module Az.Accounts is installed and the version is correct.
```

```
00:00 - INFO      - Module Az.Resources is installed and the version is correct.
00:00 - INFO      - Connected to Azure subscription name: AG-GE-CE-PHLEITEN
00:00 - INFO      - Connected to Azure subscription ID: 232b6759-xxxx-yyyy-zzzz-
757472230e6c
00:00 - INFO      - VM testvm found in Resource Group testrg
00:01 - INFO      - VM testvm is running
00:01 - INFO      - VM testvm is running Linux
00:01 - INFO      - VM testvm is running SCSI
00:02 - INFO      - Running in Azure Cloud Shell
00:02 - INFO      - Authentication token is a SecureString
00:02 - INFO      - Authentication token received
00:02 - INFO      - Getting available SKU resources
00:02 - INFO      - This might take a while ...
00:06 - INFO      - VM SKU Standard_E4bds_v5 is available in zone 1
00:06 - INFO      - Resource disk support matches between original VM size and new
VM size.
00:06 - INFO      - Found VM SKU - Checking for Capabilities
00:06 - INFO      - VM SKU has supported capabilities
00:06 - INFO      - VM supports NVMe
00:06 - INFO      - Pre-Checks completed
00:06 - INFO      - Entering Linux OS section
00:37 - INFO      -    Script output: Enable succeeded:
00:37 - INFO      -    Script output: [stdout]
00:37 - INFO      -    Script output: [INFO] Operating system detected: sles
00:37 - INFO      -    Script output: [INFO] Checking if NVMe driver is included
in initrd/initramfs...
00:37 - INFO      -    Script output: [INFO] NVMe driver found in
initrd/initramfs.
00:37 - INFO      -    Script output: [INFO] Checking nvme_core.io_timeout
parameter...
00:37 - INFO      -    Script output: [INFO] nvme_core.io_timeout is set to 240.
00:37 - INFO      -    Script output: [INFO] Checking /etc/fstab for deprecated
device names...
00:37 - INFO      -    Script output: [INFO] /etc/fstab does not contain
deprecated device names.
00:37 - INFO      -    Script output:
00:37 - INFO      -    Script output: [stderr]
00:37 - INFO      -    Script output:
00:37 - INFO      - Errors: 0 - Warnings: 0 - Info: 7
00:37 - INFO      - Shutting down VM testvm
01:18 - INFO      - VM testvm stopped
01:18 - INFO      - Checking if VM is stopped and deallocated
01:19 - INFO      - Setting OS Disk capabilities for
testvm_OsDisk_1_165411276cbe459097929b981eb9b3e2 to new Disk Controller Type to
NVMe
01:19 - INFO      - generated URL for OS disk update:
01:19 - INFO      - https://management.azure.com/subscriptions/232b6759-xxxx-yyyy-
zzzz-
757472230e6c/resourceGroups/testrg/providers/Microsoft.Compute/disks/testvm_OsDisk
_1_165411276cbe459097929b981eb9b3e2?api-version=2023-04-02
01:19 - INFO      - OS Disk updated
01:19 - INFO      - Setting new VM Size from Standard_E4s_v3 to Standard_E4bds_v5
and Controller to NVMe
01:19 - INFO      - Updating VM testvm
01:54 - INFO      - VM testvm updated
```

```
01:54 - INFO      - Start after update enabled for VM testvm
01:54 - INFO      - Waiting for 15 seconds before starting the VM
02:09 - INFO      - Starting VM testvm
03:31 - INFO      - VM testvm started
03:31 - INFO      - As the virtual machine got started using the script you can
check the operating system now
03:31 - INFO      - If you have any issues after the conversion you can revert the
changes by running the script with the old settings
03:31 - IMPORTANT - Here is the command to revert the changes:
03:31 - INFO      -    .\Azure-NVMe-Conversion.ps1 -ResourceGroupName testrg -
VMName testvm -NewControllerType SCSI -VMSize Standard_E4s_v3 -StartVM
03:31 - INFO      - Script ended at 06/27/2025 15:45:11
03:31 - INFO      - Exiting
PS /home/philipp>
```

If you have challenges accessing the operating system afterwards try to check

- Serial Console for Linux operating systems
- Screenshot from the operating system in Azure portal

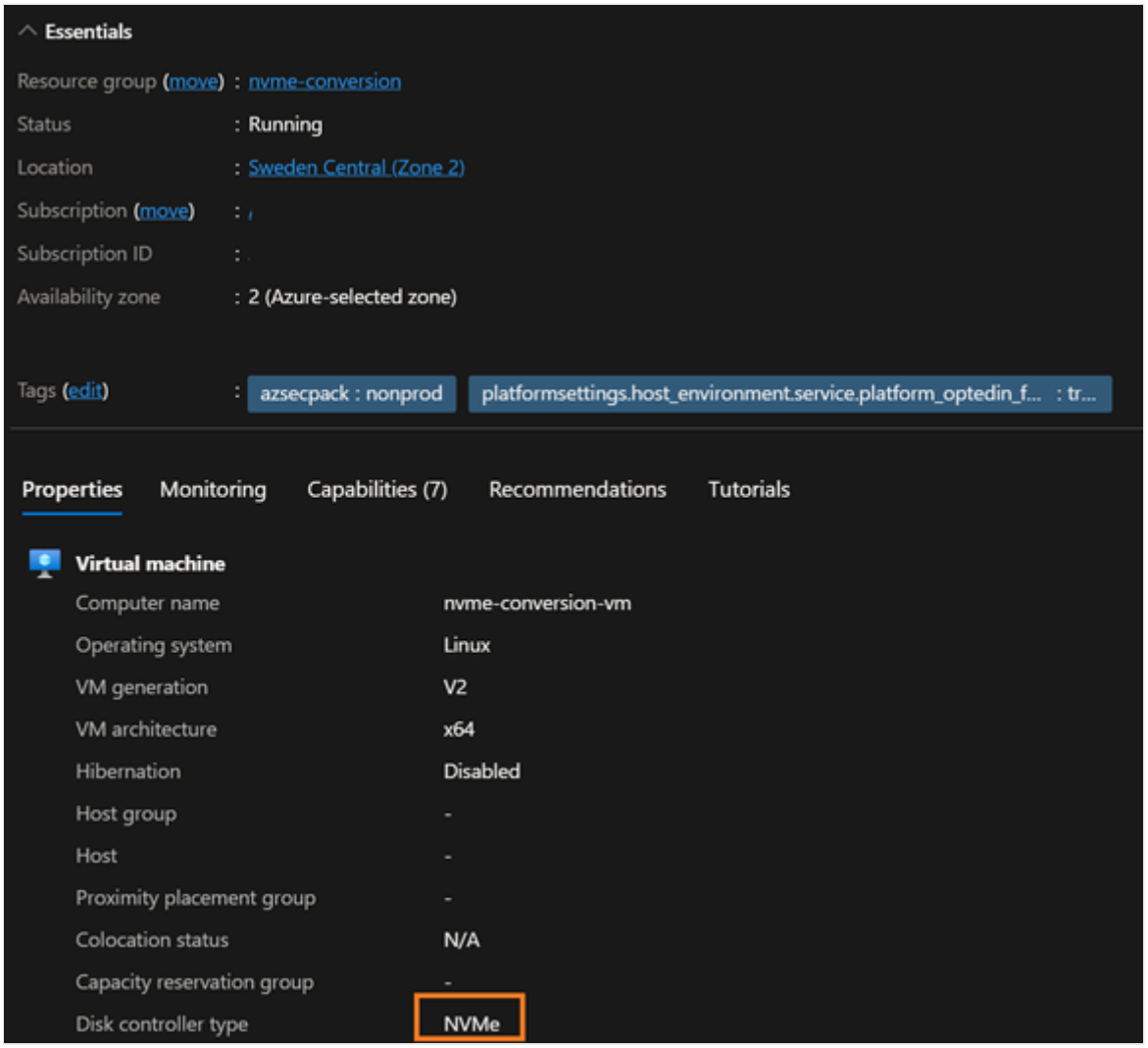
When something happens you can always revert back to SCSI using the command shown at the end of the script:

PowerShell

```
.\Azure-NVMe-Conversion.ps1 -ResourceGroupName testrg -VMName testvm -
NewControllerType SCSI -VMSize Standard_E4s_v3 -StartVM
```

2.4 Check the result

2.4.1 Check result in Azure portal



2.4.2 Check result in PowerShell

Powershell

```
PS C:\Users> $vm = Get-AzVM -name [your-vm-name]
PS C:\Users> $vm.StorageProfile.DiskControllerType
NVMe
PS C:\Users>
```

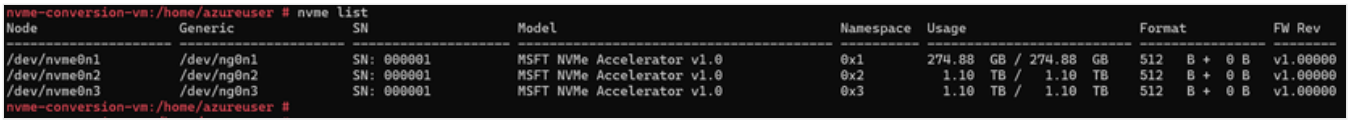
3. Check your operating system

3.1 Check devices

You can check the devices using nvme command, if the nvme command is missing, install the "nvme-cli" package.

`nvme list`

The output should show the OS disk and the data disks.



3.2 Get udev file for NVMe (Optional)

On SCSI virtual machines, the udev rules integrated in waagent (Azure agent) created links in `/dev/disk/azure/scsi1/lunX` to identify the data disks. As SCSI isn't used anymore, the rules don't apply.

With one of the two available options to deploy NVMe enabled udev rules you see new symbolic links in the directory `/dev/disk/azure/data/by-lun`. This directory is the replacement for `/dev/disk/azure/scsi1`.

Bash

```
nvme-conversion-vm:/usr/lib/udev/rules.d # ls -l /dev/disk/azure/data/by-lun/
total 0
lrwxrwxrwx 1 root root 19 Jun 7 13:52 0 -> ../../../../nvme0n2
lrwxrwxrwx 1 root root 19 Jun 7 13:52 1 -> ../../../../nvme0n3
nvme-conversion-vm:/usr/lib/udev/rules.d #
```

3.2.1 Manual download of udev file

To download the new udev rules file, use this command: `curl https://raw.githubusercontent.com/Azure/SAP-on-Azure-Scripts-and-Utilities/refs/heads/main/NVMe-Preflight-Check/88-azure-nvme-data-disk.rules` and then run `udevadm control --reload-rules && udevadm trigger` to reload the udev rules.

3.2.2 Ready to install packages using Azure VM utils

There are precompiled packages available on Index of /results/cjp256/azure-vm-utils/ for multiple distributions.

Multiple distributions started already to integrate the package. You can directly install it from their repository.

Expand table

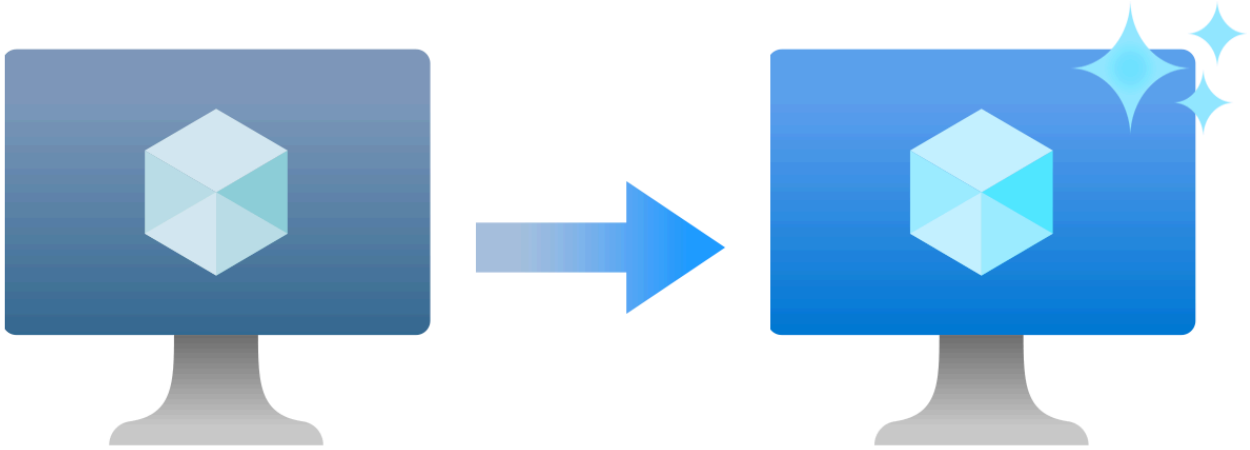
Distribution	Minimum version
SUSE	SLES 15 SP5 or newer

Distribution	Minimum version
RedHat	RHEL 9.6 or newer
Ubuntu	Ubuntu 25.04 or newer

Previous-gen and retired VM size series

03/17/2025

Azure virtual machine sizes are the amount of resources allocated to a virtual machine in the cloud. These resources are a portion of the physical server's hardware capabilities. A *size series* is a collection of all sizes that are available within a single physical server's hardware. As a size series' physical hardware ages and newer components are released, Microsoft stops deploying more of previously established series' hardware. Once users migrate off of said hardware or the hardware becomes sufficiently outdated, it's retired to make room for new infrastructure.



When hardware becomes *previous-gen* or begins the retirement process, we recommend migrating workloads to newer generation hardware. Newer hardware provides better performance and future scalability. Migrating also helps you to avoid any potential issues that may arise from using outdated hardware. By keeping your hardware up-to-date, you can ensure that your workloads are running smoothly and efficiently.

This article describes the various states of older hardware and explains the processes behind these states.

Previous-gen sizes

Previous generation size series **are not currently retired** and can still be used. These sizes are still fully supported, but they aren't guaranteed to receive more capacity. We recommend migrating to the latest generation replacements as soon as possible, especially when trying to scale up your installations.

There are two types of previous-gen sizes; *next-gen available* and *capacity limited*.

Next-gen available

Size series listed as *next-gen available* means that while no capacity limitation plans have been made, there are sufficient deployments of newer alternative sizes to justify considering these series "previous-gen". *Next-gen available* series have no formal announcement of retirement timelines, but they'll eventually be retired.

For a list of previous-gen size series that are considered *next-gen available*, see the list of previous-gen sizes.

Capacity limited

Size series listed as *capacity limited* means no extra capacity for this series will be deployed. Similar to *next-gen available* previous-gen series, there are sufficient deployments of newer alternative sizes available to replace *capacity limited* size deployments.

If you had one VM running on a *capacity limited* size and require another VM of a similar size in the same series, there is no guarantee that capacity will be able to fulfill this request. Any new VMs should be created using sizes in series that aren't considered previous-gen. Migrating to sizes considered *next-gen available* is not recommended since these sizes are also similarly outdated.

For a list of previous-gen size series that are considered *capacity limited*, see the list of previous-gen sizes.

Retired sizes

Retired size series are **no longer available** and can't be used. If you have VMs running on sizes with **retirement announced** status, migrate to a newer next-gen size as soon as possible.

Retiring hardware is necessary over time to ensure that the latest and greatest technology is available on Azure. Retirement ensures that the hardware is reliable, secure, and efficient. It also allows for the latest features and capabilities that may not be present on previous generations of hardware.

Hardware which is currently listed as *previous-gen* are often preparing for retirement which will be announced formally prior to the retirement date.

For information on size series that have been retired or are scheduled for retirement, see the list of retired sizes.

Migrate to newer sizes

Migrating to newer sizes allows you to keep up with the latest hardware and features available on Azure.

Some size series have specific migration instructions due to the unique hardware or software features. To see a list of migration guide, see retired size migration guides and previous-gen size migration guides.

For sizes without specific instructions, you can resize your VM to a newer size using the Azure portal, Azure PowerShell, Azure CLI, or Terraform. Make sure that the new size has support for all features that your workload requires.

Next steps

- For more information on VM sizes, see Sizes for virtual machines in Azure.
- For a list of retired sizes, see Retired Azure VM sizes.
- For a list of previous-gen sizes, see Previous generation Azure VM sizes.

Previous generation Azure VM size series

This article provides a list of all size series that are considered *previous-gen*. Status is listed as *next-gen available* or *capacity limited* based on capacity. For sizes that require it there are *migration guides* to help move to replacement sizes.

> **Note**
>
> Series listed as *previous-gen* are **not currently retired** and can still be used. However, series announced for retirement will have restrictions when deploying through new subscriptions. To view retired size series, see Retired Azure VM size series.

What are previous-gen size series?

Previous generation virtual machine size series are running on older hardware. While they can still be used and are fully supported, there are newer generations available. It's recommended to migrate to the latest generation replacements.

To learn more about previous-gen sizes, retirement, and the status states of previous-gen size series, see the size series retirement overview.

General purpose previous-gen sizes

Expand table

Series name	Status	Migration guide
B-series (V1)	Capacity limited	Retirement Announced
Standard D-series	Capacity limited	Retirement Announced
Preview DC-series	Capacity limited	
DS-series	Capacity limited	Retirement Announced
Dv1 and Dsv1-series	Capacity limited	Retirement Announced
Dv2 and Dsv2-series	Capacity limited	Retirement Announced
Av2 and Amv2-series	Capacity limited	Retirement Announced

For a list of general purpose sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired general purpose sizes.

Compute optimized previous-gen sizes

Expand table

Series name	Status	Migration guide
F-series	Capacity limited	Retirement Announced
Fs-series	Capacity limited	Retirement Announced
Fsv2-series	Capacity limited	Retirement Announced

For a list of compute optimized sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired compute optimized sizes.

Memory optimized previous-gen sizes

Expand table

Series name	Replacement series	Migration guide
Ev3 and Esv3-series	Next-gen available	
Ev4 and Esv4-series	Next-gen available	
Eav4 and Easv4-series	Next-gen available	
Edv4 and Edsv4-series	Next-gen available	
GS-series	Capacity limited	Retirement Announced
G-series	Capacity limited	Retirement Announced
Memory-optimized D-series	Capacity limited	Retirement Announced
Memory-optimized DS-series	Capacity limited	Retirement Announced

For a list of memory optimized sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired memory optimized sizes.

Storage optimized previous-gen sizes

Expand table

Series name	Replacement series	Migration guide
Lsv1-series	Capacity limited	Retirement Announced
Lsv2-series	Capacity limited	Retirement Announced

For a list of storage optimized sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired storage optimized sizes.

GPU accelerated previous-gen sizes

Currently there are no previous-gen or capacity limited GPU accelerated sizes.

For a list of GPU accelerated sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired GPU accelerated sizes.

FPGA accelerated previous-gen sizes

Currently there are no previous-gen or capacity limited FPGA accelerated sizes.

For a list of FPGA accelerated sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired fpga accelerated sizes.

HPC previous-gen sizes

Currently there are no previous-gen or capacity limited HPC sizes.

For a list of HPC sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired HPC sizes.

ADH previous-gen sizes

Currently there are no previous-gen or capacity limited ADH sizes.

For a list of ADH sizes listed as "retired" and "announced for retirement" (sizes that are no longer available or soon to be unavailable for use), see retired ADH sizes.

Next steps

- For a list of retired sizes, see Retired Azure VM sizes.
- For more information on VM sizes, see Sizes for virtual machines in Azure.

Last updated on 10/28/2025

Av1-series retirement

03/17/2025

On August 31, 2024, we retire Basic and Standard A-series virtual machines (VMs). Before that date, migrate your workloads to Av2-series VMs, which provide more memory per vCPU and faster storage on solid-state drives (SSDs). The remaining VMs with these specific sizes on your subscription will be set to a deallocated state. These VMs will be stopped and removed from the host. These VMs will no longer be billed in the deallocated state.

> **Note**
>
> In some cases, you must deallocate the VM prior to resizing. This can happen if the new size is not available on the hardware cluster that is currently hosting the VM.

Migrate workloads to Av2-series VMs

You can resize your virtual machines to the Av2-series using the Azure portal, PowerShell, or the CLI. Below are examples on how to resize your VM using the Azure portal and PowerShell.

> **Important**
>
> Resizing a virtual machine results in a restart. We recommend that you perform actions that result in a restart during off-peak business hours.

Azure portal

1. Open the Azure portal.
2. Type *virtual machines* in the search.
3. Under **Services**, select **Virtual machines**.
4. In the **Virtual machines** page, select the virtual machine you want to resize.
5. In the left menu, select **size**.
6. Pick a new Av2 size from the list of available sizes and select **Resize**.

Azure PowerShell

1. Set the resource group and VM name variables. Replace the values with information of the VM you want to resize.

```
```

```powershell
$resourceGroup = "myResourceGroup"
$vmName = "myVM"
```

2. List the VM sizes that are available on the hardware cluster where the VM is hosted.

```powershell
Get-AzVMSize -ResourceGroupName $resourceGroup -VMName $vmName
```

3. Resize the VM to the new size.

```powershell
$vm = Get-AzVM -ResourceGroupName $resourceGroup -VMName $vmName
$vm.HardwareProfile.VmSize = "<newAv2VMsize>"
Update-AzVM -VM $vm -ResourceGroupName $resourceGroup
```

Help and support

If you have questions, ask community experts in Microsoft Q&A. If you have a support plan and need technical help, create a support request:

1. In the Help + support page, select **Create a support request**. Follow the **New support request** page instructions. Use the following values:

   - For **Issue type**, select **Technical**.
   - For **Service**, select **My services**.
   - For **Service type**, select **Virtual Machine running Windows/Linux**.
   - For **Resource**, select your VM.
   - For **Problem type**, select **Assistance with resizing my VM**.
   - For **Problem subtype**, select the option that applies to you.

Follow instructions in the **Solutions** and **Details** tabs, as applicable, and then **Review + create**.

Next steps

Learn more about the Av2-series VMs

Bv1 sizes series

Summarize this article for me

> **Note**
>
> These virtual machine sizes are a previous generation series. While older VM sizes are supported until further notice, we recommended using newer generations for improved performance and security. Check out the sizes overview's list of VM size families by type for a selection of newer sizes.

The B-series VMs can be deployed on various hardware types and processors, so competitive bandwidth allocation is provided. B-series run on the third Generation Intel® Xeon® Platinum 8370C (Ice Lake), the Intel® Xeon® Platinum 8272CL (Cascade Lake), the Intel® Xeon® 8171M 2.1 GHz (Skylake), the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors. B-series VMs are ideal for workloads that don't need the full performance of the CPU continuously, like web servers, proof of concepts, small databases, and development build environments. These workloads typically have burstable performance requirements. To determine the physical hardware on which this size is deployed, query the virtual hardware from within the virtual machine. The B-series provides you with the ability to purchase a VM size with baseline performance that can build up credits when its using less than its baseline. When the VM has accumulated credits, the VM can burst above the baseline using up to 100% of the vCPU when your application requires higher CPU performance.

Read more about the B-series CPU credit model.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 20 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64]
Memory	0.5 - 80 GiB	
Local Storage	1 Disk	4 - 160 GiB IOPS

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
		MBps
Remote Storage	2 - 32 Disks	160 - 4,000 IOPS 10 - 50 MBps
Network	2 - 8 NICs	Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_B1ls2	1	0.5
Standard_B1s	1	1
Standard_B1ms	1	2
Standard_B2s	2	4
Standard_B2ms	2	8
Standard_B4ms	4	16
Standard_B8ms	8	32
Standard_B12ms	12	48
Standard_B16ms	16	64
Standard_B20ms	20	80

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

> **Note**
>
> Accelerated Networking is only supported for Standard_B12ms, Standard_B16ms and Standard_B20ms.

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dv3 sizes series

Summarize this article for me

> **Note**
>
> These virtual machine sizes are a previous generation series. While older VM sizes are supported until further notice, we recommended using newer generations for improved performance and security. Check out the sizes overview's list of VM size families by type for a selection of newer sizes.

The Dv3-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads. Memory has been expanded (from ~3.5 GiB/vCPU to 4 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Dv3-series no longer has the high memory VM sizes of the D/Dv2-series, those have been moved to the memory optimized Ev3 and Esv3-series. Example D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	1 Disk	50 - 1,600 GiB 3,000 - 96,000 IOPS 46 - 1,000 MBps
Remote Storage	4 - 32 Disks	Disk Types: Standard SDD/HDD

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - 8 NICs	1,000 - 30,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2_v3	2	8
Standard_D4_v3	4	16
Standard_D8_v3	8	32
Standard_D16_v3	16	64
Standard_D32_v3	32	128
Standard_D48_v3	48	192
Standard_D64_v3	64	256

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dsv3 sizes series

> **Note**
>
> These virtual machine sizes are a previous generation series. While older VM sizes are supported until further notice, we recommended using newer generations for improved performance and security. Check out the sizes overview's list of VM size families by type for a selection of newer sizes.

The Dv3-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads. Memory has been expanded (from ~3.5 GiB/vCPU to 4 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Dv3-series no longer has the high memory VM sizes of the D/Dv2-series, those have been moved to the memory optimized Ev3 and Esv3-series. Example D-series use cases include enterprise-grade applications, relational databases, in-memory caching, and analytics.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	2 - 64 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	8 - 256 GiB	
Local Storage	1 Disks	16 - 512 GiB 4,000 - 128,000 IOPS 32 - 1,024 MBps

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Remote Storage	4 - 32 Disks	4,000 - 80,000 IOPS 48 - 1,200 MBps Disk Types: Standard SDD/HDD, Premium SSD, Premium SSD v2, Ultra Disk
Network	2 - 8 NICs	1,000 - 30,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D2s_v3	2	8
Standard_D4s_v3	4	16
Standard_D8s_v3	8	32
Standard_D16s_v3	16	64
Standard_D32s_v3	32	128
Standard_D48s_v3	48	192
Standard_D64s_v3	64	256

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Supported
Ephemeral OS Disks	Supported
Nested Virtualization	Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 04/02/2026

Dv2 sizes series

Summarize this article for me

> **Note**
>
> These virtual machine sizes are a previous generation series. While older VM sizes are supported until further notice, we recommended using newer generations for improved performance and security. Check out the sizes overview's list of VM size families by type for a selection of newer sizes.

The Dv2 and DSv2-series, a follow-on to the original D-series, feature a more powerful CPU and optimal CPU-to-memory configuration making them suitable for most production workloads. The Dv2-series is about 35% faster than the D-series. Dv2-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with the Intel Turbo Boost Technology 2.0. The Dv2-series has the same memory and disk configurations as the D-series.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 16 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	3.5 - 56 GiB	
Local Storage	1 Disk	50 - 800 GiB 3,000 - 48,000 IOPS 46 - 750 MBps
Remote Storage	4 - 64 Disks	4x500 - 64x500 IOPS Disk Types: Standard SDD/HDD

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - NICs	750 - 12,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_D1_v2	1	3.5
Standard_D2_v2	2	7
Standard_D3_v2	4	14
Standard_D4_v2	8	28
Standard_D5_v2	16	56

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Not Supported

Feature name	Support status
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Dsv2 sizes series

Summarize this article for me

> **Note**
>
> These virtual machine sizes are a previous generation series. While older VM sizes are supported until further notice, we recommended using newer generations for improved performance and security. Check out the sizes overview's list of VM size families by type for a selection of newer sizes.

The Dv2 and DSv2-series, a follow-on to the original D-series, feature a more powerful CPU and optimal CPU-to-memory configuration making them suitable for most production workloads. The Dv2-series is about 35% faster than the D-series. Dv2-series run on the Intel® Xeon® Platinum 8573C (Emerald Rapids), Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1GHz (Skylake), Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors with the Intel Turbo Boost Technology 2.0. The Dv2-series has the same memory and disk configurations as the D-series.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	1 - 16 vCPUs	Intel Xeon Platinum 8370C (Ice Lake) [x86-64] Intel Xeon Platinum 8272CL (Cascade Lake) [x86-64] Intel Xeon 8171M (Skylake) [x86-64] Intel Xeon E5-2673 v4 (Broadwell) [x86-64] Intel Xeon E5-2673 v3 (Haswell) [x86-64] Intel Xeon Platinum 8573C (Emerald Rapids) [x86-64]
Memory	3.5 - 56 GiB	
Local Storage	1 Disk	7 - 112 GiB 4,000 - 64,000 IOPS 32 - 512 MBps
Remote Storage	4 - 64 Disks	3,000 - 51,000 IOPS 48 - 768 MBps Disk Types: Standard SDD/HDD

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Network	2 - 8 NICs	750 - 12,000 Mbps Interfaces: NetVSC, ConnectX, MANA
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_DS1_v2	1	3.5
Standard_DS2_v2	2	7
Standard_DS3_v2	4	14
Standard_DS4_v2	8	28
Standard_DS5_v2	16	56

VM Basics resources

- What are vCPUs
- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported

Feature name	Support status
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/11/2026

Memory optimized Dv2 and Dsv2-series

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

> **Note**
>
> These virtual machine sizes are a previous generation series. While older VM sizes are supported until further notice, we recommended using newer generations for improved performance and security. Check out the sizes overview's list of VM size families by type for a selection of newer sizes.

Dv2 and Dsv2-series, a follow-on to the original D-series, features a more powerful CPU. DSv2-series sizes run on the third Generation Intel® Xeon® Platinum 8370C (Ice Lake), the Intel® Xeon® Platinum 8272CL (Cascade Lake), the Intel® Xeon® 8171M 2.1 GHz (Skylake), the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors. The Dv2-series has the same memory and disk configurations as the D-series.

Dv2-series 11-15

Dv2-series sizes run on the third Generation Intel® Xeon® Platinum 8370C (Ice Lake), the Intel® Xeon® Platinum 8272CL (Cascade Lake), the Intel® Xeon® 8171M 2.1 GHz (Skylake), the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors.

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max temp storage throughput: IOPS/Read MBps/Write MBps	Max data disks/throughput: IOPS	Max NICs	Expected network bandwidth (Mbps)
Standard_D11_v2	2	14	100	6000/93/46	8/8x500	2	1500
Standard_D12_v2	4	28	200	12000/187/93	16/16x500	4	3000
Standard_D13_v2	8	56	400	24000/375/187	32/32x500	8	6000
Standard_D14_v2	16	112	800	48000/750/375	64/64x500	8	12000

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max temp storage throughput: IOPS/Read MBps/Write MBps	Max data disks/throughput: IOPS	Max NICs	Expected network bandwidth (Mbps)
Standard_D15_v2 [1]	20	140	1000	60000/937/468	64/64x500	8	25000 [2]

[1] Instance is isolated to hardware dedicated to a single customer.
[2] 25000 Mbps with Accelerated Networking.

Expand table

Feature Name	Support status
Premium Storage	Not Supported
Premium Storage caching	Not Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1
Accelerated Networking	Supported
Ephemeral OS Disks	Not Supported
Nested Virtualization	Not Supported

DSv2-series 11-15

DSv2-series sizes run on the third Generation Intel® Xeon® Platinum 8370C (Ice Lake), the Intel® Xeon® Platinum 8272CL (Cascade Lake), the Intel® Xeon® 8171M 2.1 GHz (Skylake), the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell), or the Intel® Xeon® E5-2673 v3 2.4 GHz (Haswell) processors.

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max cached and temp storage throughput: IOPS/MBps (cache size in GiB)	Max uncached disk throughput: IOPS/MBps	Max NICs	Expected network bandwidth (Mbps)
Standard_DS11_v2 [3]	2	14	28	8	8000/64 (72)	6400/96	2	1500
Standard_DS12_v2 [3]	4	28	56	16	16000/128 (144)	12800/192	4	3000
Standard_DS13_v2 [3]	8	56	112	32	32000/256 (288)	25600/384	8	6000
Standard_DS14_v2 [3]	16	112	224	64	64000/512 (576)	51200/768	8	12000
Standard_DS15_v2 [2]	20	140	280	64	80000/640 (720)	64000/960	8	25000 [4]

1. Maximum disk throughput (IOPS or MBps) possible with a DSv2 series VM are limited by the number, size, and striping of the attached disks
2. Instance is isolated to the Intel Haswell based hardware and dedicated to a single customer.
3. Constrained core sizes available.
4. 25,000 Mbps with Accelerated Networking.

Expand table

Feature Name	Support status
Premium Storage	Supported
Premium Storage caching	Supported
Live Migration	Supported
Memory Preserving Updates	Supported
VM Generation Support	Generation 1 and 2
Accelerated Networking	Supported
Ephemeral OS Disks	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

DCsv2-series retirement

This migration guide is designed for users of DCsv2-series virtual machines (VMs), which are scheduled for retirement on **June 30, 2026**. To ensure minimal disruption and to continue optimizing cost and performance, this guide helps you transition to the latest series VMs.

This document covers:

- Recommended options for migration
- Detailed migration steps
- Frequently Asked Questions

By migrating to newer VM series, you gain access to improved price-performance ratios, broader regional availability, and the latest hardware capabilities.

Recommended options for migration

Before June 30, 2026, migrate your workloads to one of the following options that best aligns with your business needs:

- If you want to continue using the enclave-based offering with Intel SGX technology, migrate your workloads to DCdsv3 virtual machines. The DCdsv3 VMs offer significant improvements, including enhanced performance and increased memory capacity, making the DCdsv3-series a more robust and efficient choice.
- To lift and shift into a VM based programming model, consider DCasv5/DCadsv5/ECasv5/ECadsv5 VMs, DCasv6/ECasv6 series (currently in preview), or DC/ECesv6 (currently in preview) confidential VMs (CVMs).
- If you already have or plan to transition to containerized workloads and want to lift and shift your containerized applications consider using Azure Confidential Container Instances (C-ACI) serverless infrastructure. If you need to orchestrate containerized workloads, consider using Virtual nodes on Azure Container Instances (C-VN2) for Azure Kubernetes Service (AKS).

Additionally, there may be changes to your Azure Virtual Machines billing because of this retirement. Refer to our Azure Virtual Machines pricing page for more information.

Migration Steps

Start planning your migration from DCsv2-series today.

Identify the Target Migration Option

- Learn more about the different migration options and their benefits in the section *Recommended options for migration*.
- Evaluate your current VM's workload and performance requirements and identify the target migration option.

Check and Request Quota Increases

- Before resizing and migration, verify that your subscription has sufficient quota for the target VM series.
- Request more quota through the Azure portal if needed.

Complete Migration

- Complete migration as soon as possible to prevent business impact and to take advantage of the improved performance, and extensive regional coverage of our new confidential computing offerings.
- Dependent on chosen migration option follows the documentation below.
- For technical questions, issues, and help get answers from community experts in Microsoft Q&A.

Migrate workloads to DCdsv3-series VMs

To continue using the enclave-based offering with Intel SGX technology, migrate your workloads to DCdsv3 virtual machines. First, check if your desired DCdsv3-series SKU is available in your current region. If it is, you can resize your virtual machines to the DCdsv3-series using the Azure portal, PowerShell, or the CLI. Below are examples of how to resize your VM using the Azure portal, PowerShell, and Azure CLI.

> **ⓘ Important**
>
> Resizing a virtual machine results in a restart. We recommend that you perform actions that result in a restart during off-peak business hours.
>
> Deallocating the VM also releases any dynamic IP addresses assigned to the VM. The OS and data disks are not affected.
>
> Additionally, if your DCsv2 VMs are running in North Central US, Canada East, Australia Southeast, or UK West, please follow the instructions in the next section.

Azure portal

1. Open the Azure portal.
2. Type *virtual machines* in the search.
3. Under **Services**, select **Virtual machines**.
4. In the **Virtual machines** page, select the virtual machine you want to resize.
5. Stop (deallocate) the VM.
6. In the left menu under Availability + scale, select **Size** option.
7. Pick a new DCdsv3 size from the list of available sizes and select **Resize**.
8. **Start the VM** after resizing.

Refer to the full Azure VM resizing guide for more detailed instructions.

Azure PowerShell

1. Open a PowerShell session

2. Authenticate your device and set your subscription.

```PowerShell
Connect-AzAccount –UseDeviceAuthentication
Set-AzContext -SubscriptionName "Your-Subscription-Name"
```

3. Set the resource group and VM name variables. Replace the values with information of the VM you want to resize.

```PowerShell
$resourceGroup = "myResourceGroup"
$vmName = "myVM"
```

4. Stop the VM.

```PowerShell
Stop-AzVM -Name $vmName -ResourceGroupName $resourceGroup
```

5. List the VM sizes that are available on the hardware cluster where the VM is hosted.

```PowerShell
Get-AzVMSize -ResourceGroupName $resourceGroup -VMName $vmName
```

6. Resize the VM to the new size.

PowerShell

```
$vm = Get-AzVM -ResourceGroupName $resourceGroup -VMName $vmName
$vm.HardwareProfile.VmSize = "<newDCdsv3VMsize>"
Update-AzVM -VM $vm -ResourceGroupName $resourceGroup
Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
```

Azure CLI

1. Open a terminal session and ensure you have Azure CLI installed (or use the portal cli)

2. Authenticate your device and set your subscription.

   PowerShell

   ```
   az login
   az account set --subscription "<subscriptionIdHere>”
   ```

3. Set the resource group, VM name and new size variables. Replace the values with information of the VM you want to resize.

   PowerShell

   ```
   resourceGroup="myResourceGroup"
   vmName="myVM"
   newSize=<newDCdsv3VMsize> # ex: Standard_DC24ds_v3
   ```

4. Resize the VM to the new size.

   PowerShell

   ```
   az vm deallocate --resource-group $resourceGroup --name $vmName
   az vm resize --resource-group $resourceGroup --name $vmName --size $newSize
   az vm start --resource-group $resourceGroup --name $vmName
   ```

Important Notice for DCsv2 VMs running in North Central US, Canada East, Australia Southeast, or UK West

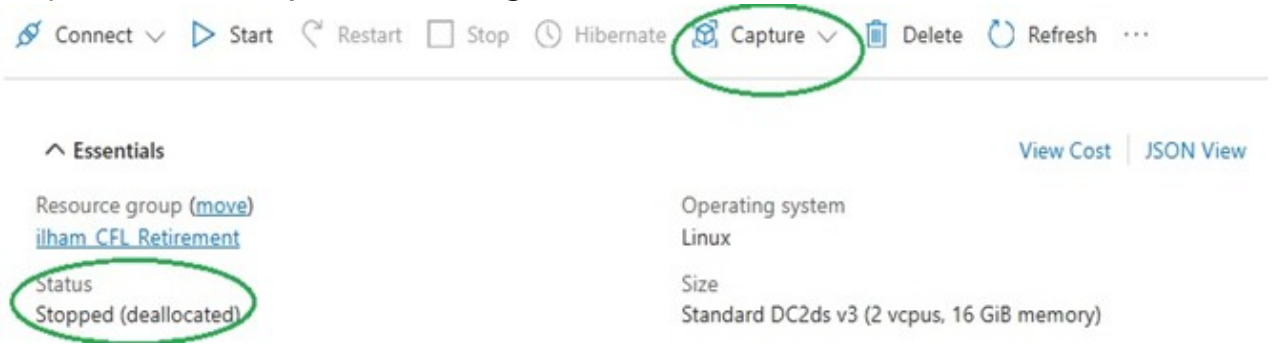
If your DCsv2 VMs are running in North Central US, Canada East, Australia Southeast, or UK West, please consider migrating them to the available regions listed below where the DCdsv3 SKU is available or explore our new generation of confidential computing offerings.

> ⓘ **Note**

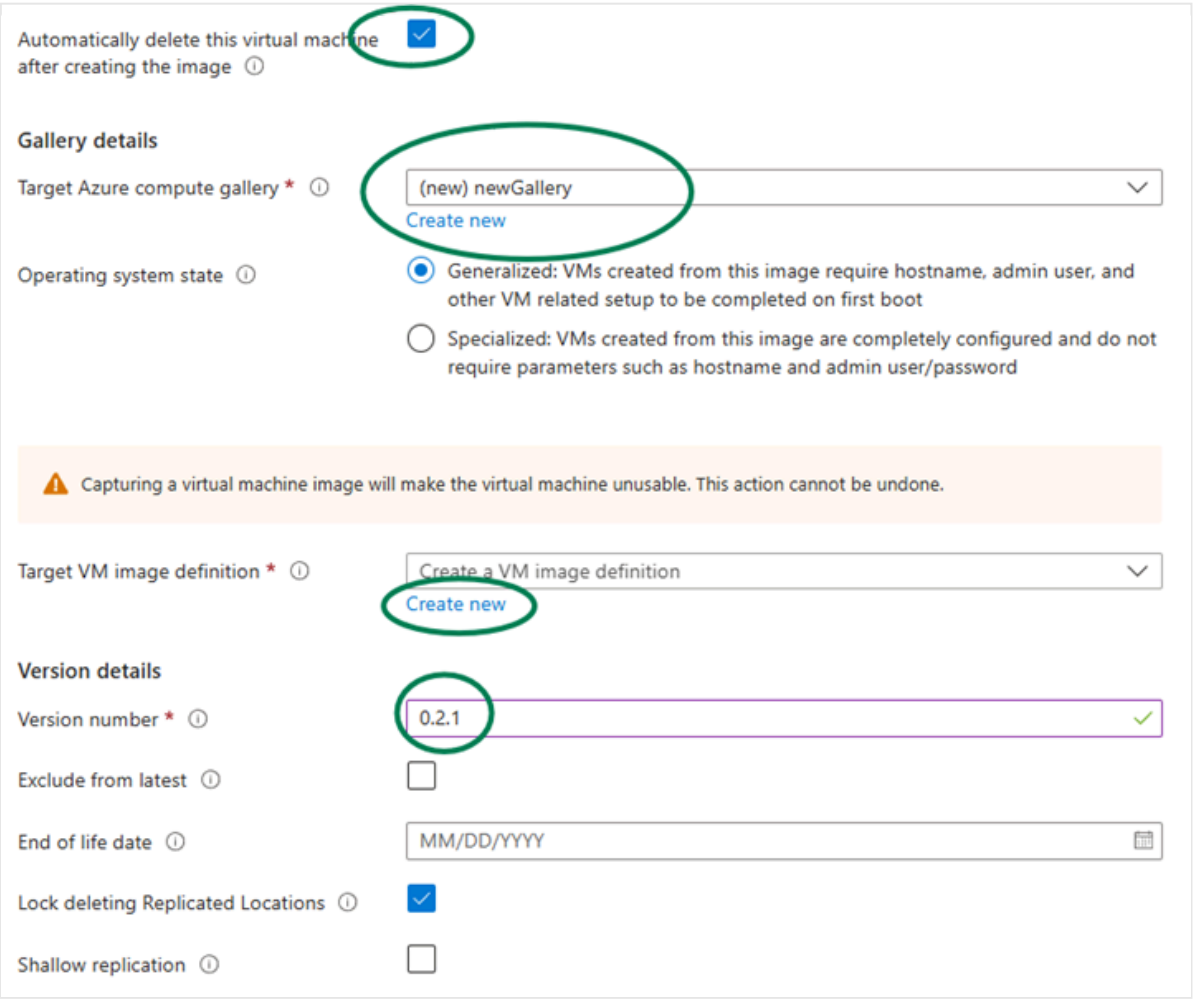
Available Regions:
Australia East, Canada Central, Central India, Central US, East US, East US 2, West US, West US 2, South Central US, Germany North, Germany West Central, Italy North, Japan East, Japan West, North Europe, South India, Southeast Asia, UK South, West Europe, UAE Central, UAE North, Switzerland North

To migrate your VMs to one of these available regions, follow these steps:

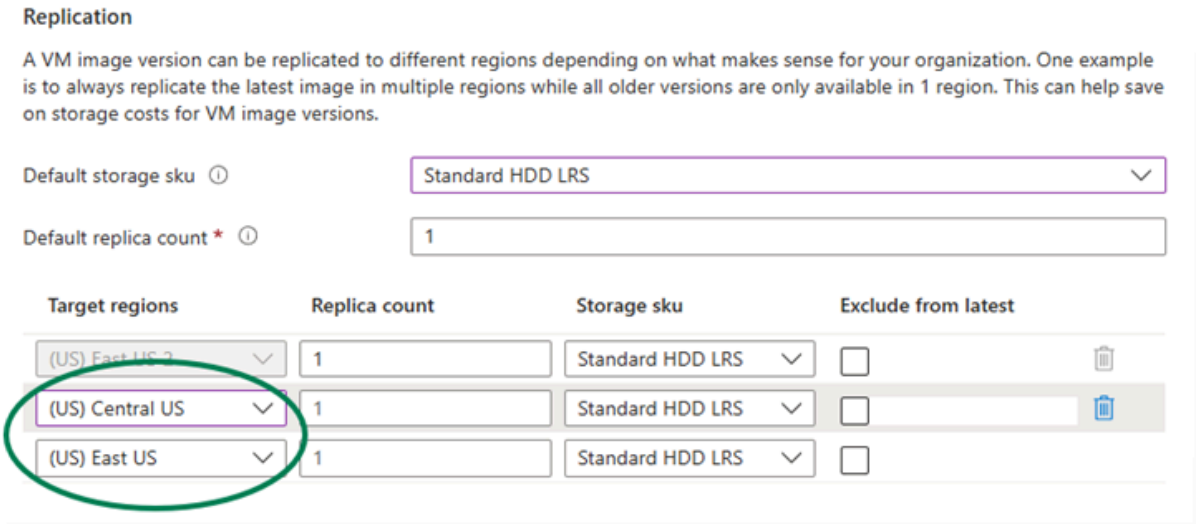
1. Identify the **target region** from the list above.
2. Prepare your VMs for migration by ensuring all data is backed up.
3. Verify that your subscription has sufficient quota for the DCdsv3-series VMs in the target region. Request a quota through the Azure portal if needed.
4. Navigate to VM you want to resize in the portal.
5. **Stop the VM** and wait for status of the VM to be Stopped (deallocated).
6. Find the **Capture** drop down in the overview tab of the VM (circled bellow). From the drop-down select **capture**, then image.



7. When you reach **the image creation page**:
    - Ensure you select **Automatically delete this virtual machine** after creating the image.
    - If you don't have a gallery, select **Create new** in the gallery option and name your gallery.
    - Fill in the name and other required options.
    - Create an image definition if none is available.
    - Read the descriptions for specialized vs generalized images and choose your option. (If not sure, generalized should work for most cases.)



- Continue to fill in the other options. In the replication section of the image capture option, add the region where you wish to relocate your VM. You'll need to select the target region from the dropdown menu.



8. Go to your gallery and select the image you captured. In the top left, select **Create a VM**.

- Fill in all options and ensure you select the new region and new size.
- If you don't see the size you want, ensure **No infrastructure redundancy required** is selected and that your requested region supports the desired size.

Migrate to new generation offerings

- To lift and shift into a VM based programming model, consider DCasv5/DCadsv5/ECasv5/ECadsv5 VMs, DCasv6/ECasv6 series (currently in preview) or DC/ECesv6 (currently in preview) confidential VMs (CVMs).
- If you already have or plan to transition to containerized workloads and want to lift and shift your containerized applications consider using Azure Confidential Container Instances (C-ACI) serverless infrastructure.
- If you need to orchestrate containerized workloads, consider using Virtual nodes on Azure Container Instances (C-VN2) for Azure Kubernetes Service (AKS).

Frequently Asked Questions

How does the DCsv2-series retirement affect me?

If you're running your workload on DCsv2-series, either by using virtual machines, Virtual Machine Scale Sets or by having app-enclave aware containers running on Azure Kubernetes Service, this retirement affects you.

What is the migration timeline?

On June 30, 2026, DCsv2-series virtual machines (VMs) will be retired. Before that date, please migrate your workloads to DCdsv3-series virtual machines. In case you prefer global availability and want to lift and shift your workloads consider using DCasv5/DCadsv5/ECasv5/ECadsv5 VMs, DCasv6/ECasv6 series (currently in preview) or DC/ECesv6 (currently in preview) confidential VMs (CVMs), or Azure Confidential Container Instances (C-ACI) serverless infrastructure.

Will DCsv2-series VMs still allow new customer sign-ups?

Starting from July 1, 2025, capacity restrictions will be applied to DCsv2-series virtual machines and no new subscription will be allowed.

Will Microsoft continue to support my current workload?

Yes, support continues for your workloads on DCsv2 virtual machines until the retirement date. You continue to receive SLA assurance, infrastructure updates, and maintenance.

Will other services built on top of the DCsv2 SKU still be available after the SKU retires?

No, all uses of the DCsv2 SKU retires simultaneously in June 2026, including those on Azure Kubernetes Service and Azure Virtual Machine Scale Sets.

Will DCsv2-series VMs provide any new features during the retirement period?

No, we aren't taking any feature requests or building new features for the DCsv2-series VMs. Instead, we focus on next-generation lift-and-shift offerings with more memory per vCPU, faster SSD storage, global availability, and a cloud-native approach with containerized workloads and serverless infrastructure.

Will DCsv2-series and DCsv3-series VMs be available in new regions?

No, we won't deploy DCsv2-series, DCsv3-series, and DCdsv3-series VMs in new Azure regions. Check availability in the existing DCsv3/DCdsv3 regions.

I'm using EPID Attestation (IAS). Can I continue using it on DCdsv3 VM?

No, you can't continue using Intel SGX Attestation Service Utilizing Intel EPID (IAS for short) on DCdsv3 VM. This feature isn't supported and won't be supported, as Intel announced IAS end of support on April 2, 2025. If you didn't transition away from IAS, please transition now.

How can I get a quota for the target VM size?

Follow the guide to request an increase in vCPU quota by VM family.

Can I resize to a VM size with no local temp disk?

You can't resize a VM size that has a local temp disk to a VM size with no local temp disk and vice versa. This means that you can resize from DCsv2-series VMs size to DCdsv3-series VMs, but you can't resize from DCsv2-series VMs size to DCsv3-series VMs.

I don't need a local temp disk, and I would like to resize to DCsv3-series virtual machines while keeping the same price as I was charged for DCsv2. How do I migrate from a VM size with a local temp disk to a VM size with no local temp disk?

If you're sure that you don't need a local temp disk, for a work-around, see How do I migrate from a VM size with local temp disk to a VM size with no local temp disk? The work-around can be used to resize a VM with no local temp disk to VM with a local temp disk. You create a snapshot of the VM with no local temp disk, then create a disk from the snapshot and lastly create VM from the disk with appropriate VM size that supports VMs with a local temp disk.

Does changing to a VM without a local temp disk break my custom scripts, custom images, or OS images that have scratch files or page files on a local temp disk?

If the custom OS image points to the local temp disk, the image might not work correctly with this diskless size.

Are my OS and data disks affected when resizing from DCsv2 to DCdsv3?

Deallocating the VM also releases any dynamic IP addresses assigned to the VM. The OS and data disks aren't affected.

What impact will migration from DCsv2 to DCdsv3 have on my dynamic IP addresses?

Deallocating the VM also releases any dynamic IP addresses assigned to the VM. If you need to retain the same IP addresses, consider using static IP addresses and set them in the network settings ahead of to migration (resize to DCdsv3).

What should I do if there are no DCdsv3-series VMs available in my current region?

If there are no DCdsv3-series VMs available in your current region, you can consider the following options:

- Check Availability in Nearby Regions: Look for DCdsv3-series VMs in nearby regions that might have the required capacity.
- Contact Azure Support: Reach out to Azure Support for assistance and to explore alternative solutions that meet your requirements.

How does migration affect my current billing?

During the migration and resize from DCsv2-series to DCdsv3-series VMs with a local temp disk, there is a price change. However, you'll receive a newer generation CPU, more RAM for the same number of cores, be able to attach more data disks, and have a larger local temp disk. For more information see Azure Virtual Machines pricing page for more information.

Are there any cost-saving options available during migration?

If you decide that you don't need a local temp disk, the price remains the same, and you'll still benefit from more RAM and the ability to attach more data disks. This option will require more effort after the resize to avoid using the local temp disk. For a workaround, please refer to how do I migrate from a VM size with local temp disk to a VM size with no local temp disk?

I'm on Reserved Instances (RIs) with DCsv2. How Do I Handle Migration?

If you have active Reserved Instances for DCsv2-series VMs, follow these steps:

1. Review Current Reservations

    - Check your active RIs in the Azure portal.
    - Identify which RIs expires or are affected by the VM retirement.

2. Migrate and Manage Your RIs
   Depending on your business needs, consider these options:

- Exchange Existing Reservations:
  - Swap current RIs for a new VM series without any penalties.
  - Refer to the RI Exchange Guide
- Trade-In for Savings Plan:
  - Convert your existing RIs into an Azure Savings Plan for compute.
  - This offers flexibility across VM families and regions.
  - Follow the Azure RI Trade-In Tutorial.
- Purchase New RIs:
  - Buy new reservations that align with your new VM series.
  - Consider shorter terms (1-year) for flexibility

How can I get transition help and support during migration?

If you have any questions, you can create a support request through the Azure portal for technical help.

What will happen after retirement date?

After June 30, 2026, any remaining DCsv2-series virtual machine subscriptions will stop working and will no longer incur billing charges. To avoid disruption, please migrate ahead of the retirement schedule.

Help and support

If you have questions, ask community experts in Microsoft Q&A. If you have a support plan and need technical help, create a support request:

1. In the Help + support page, select **Create a support request**. Follow the **New support request** page instructions. Use the following values:

    - For **Issue type**, select **Technical**.
    - For **Service**, select **My services**.
    - For **Service type**, select **Virtual Machine running Windows/Linux**.
    - For **Resource**, select your VM.
    - For **Problem type**, select **Assistance with resizing my VM**.
    - For **Problem subtype**, select the option that applies to you.

Follow instructions in the **Solutions** and **Details** tabs, as applicable, and then **Review + create**.

Next steps

- Learn more about DCdsv3-series VM
- Learn more about DCasv5/DCadsv5/ECasv5/ECadsv5 CVMs
- Learn more about DCasv6/ECasv6 CVMs (currently in preview)
- Learn more about DC/ECesv6 CVMs (currently in preview)
- Learn more about Azure Confidential Container Instances (C-ACI)
- Learn more about Virtual nodes on Azure Container Instances (C-VN2) for Azure Kubernetes Service (AKS).

Last updated on 06/30/2025

Ev3 and Esv3-series

07/23/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

The Ev3 and Esv3-series run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1 GHz (Skylake), or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processor in a hyper-threaded configuration, providing a better value proposition for most general purpose workloads, and bringing the Ev3 into alignment with the general purpose VMs of most other clouds. Memory has been expanded (from 7 GiB/vCPU to 8 GiB/vCPU) while disk and network limits have been adjusted on a per core basis to align with the move to hyperthreading. The Ev3 is the follow up to the high memory VM sizes of the D/Dv2 families.

Ev3-series

Ev3-series instances run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1 GHz (Skylake), or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processors, and feature Intel Turbo Boost Technology 2.0. Ev3-series instances are ideal for memory-intensive enterprise applications.

Data disk storage is billed separately from virtual machines. To use premium storage disks, use the ESv3 sizes. The pricing and billing meters for ESv3 sizes are the same as Ev3-series.

Ev3-series VM's feature Intel® Hyper-Threading Technology.

ACU: 160 - 190
Premium Storage: Not Supported
Premium Storage caching: Not Supported
Live Migration: Supported
Memory Preserving Updates: Supported
VM Generation Support: Generation 1
Accelerated Networking: Supported
Ephemeral OS Disks: Not Supported
Nested Virtualization: Supported

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max temp storage throughput: IOPS / Read MBps / Write MBps	Max NICs / Network bandwidth
Standard_E2_v3[1]	2	16	50	4	3000/46/23	2/1000
Standard_E4_v3	4	32	100	8	6000/93/46	2/2000

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max temp storage throughput: IOPS / Read MBps / Write MBps	Max NICs / Network bandwidth
Standard_E8_v3	8	64	200	16	12000/187/93	4/4000
Standard_E16_v3	16	128	400	32	24000/375/187	8/8000
Standard_E20_v3	20	160	500	32	30000/469/234	8/10000
Standard_E32_v3	32	256	800	32	48000/750/375	8/16000
Standard_E48_v3	48	384	1200	32	96000/1000/500	8/24000
Standard_E64_v3	64	432	1600	32	96000/1000/500	8/30000
Standard_E64i_v3 [2]	64	432	1600	32	96000/1000/500	8/30000

[1] Accelerated networking can only be applied to a single NIC. [2] Instance is isolated to hardware dedicated to a single customer.

Esv3-series

Esv3-series instances run on the 3rd Generation Intel® Xeon® Platinum 8370C (Ice Lake), Intel® Xeon® Platinum 8272CL (Cascade Lake), Intel® Xeon® 8171M 2.1 GHz (Skylake), or the Intel® Xeon® E5-2673 v4 2.3 GHz (Broadwell) processor, feature Intel Turbo Boost Technology 2.0 and use premium storage. Esv3-series instances are ideal for memory-intensive enterprise applications.

Esv3-series VM's feature Intel® Hyper-Threading Technology.

ACU: 160-190
Premium Storage: Supported
Premium Storage caching: Supported
Live Migration: Supported
Memory Preserving Updates: Supported
VM Generation Support: Generation 1 and 2
Accelerated Networking: Supported
Ephemeral OS Disks: Supported
Nested Virtualization: Supported

Expand table

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max cached and temp storage throughput: IOPS/MBps (cache size in GiB)	Max uncached disk throughput: IOPS/MBps	Burst uncached disk throughput: IOPS/MBps[3]	Max NICs/ Expected network bandwidth (Mbps)
Standard_E2s_v3[4]	2	16	32	4	4000/32 (50)	3200/48	4000/200	2/1000
Standard_E4s_v3 [1]	4	32	64	8	8000/64 (100)	6400/96	8000/200	2/2000
Standard_E8s_v3 [1]	8	64	128	16	16000/128 (200)	12800/192	16000/400	4/4000
Standard_E16s_v3 [1]	16	128	256	32	32000/256 (400)	25600/384	32000/800	8/8000
Standard_E20s_v3	20	160	320	32	40000/320 (400)	32000/480	40000/1000	8/10000
Standard_E32s_v3 [1]	32	256	512	32	64000/512 (800)	51200/768	64000/1600	8/16000
Standard_E48s_v3	48	384	768	32	96000/768 (1200)	76800/1152	80000/2000	8/24000
Standard_E64s_v3 [1]	64	432	864	32	128000/1024 (1600)	80000/1200	80000/2000	8/30000
Standard_E64is_v3 [2]	64	432	864	32	128000/1024 (1600)	80000/1200	80000/2000	8/30000

[1] Constrained core sizes available.

[2] Instance is isolated to hardware dedicated to a single customer.

[3] Esv3-series VMs can burst their disk performance and get up to their bursting max for up to 30 minutes at a time.

[4] Accelerated networking can only be applied to a single NIC.

Size table definitions

- Storage capacity is shown in units of GiB or 1024^3 bytes. When you compare disks measured in GB (1000^3 bytes) to disks measured in GiB (1024^3) remember that capacity numbers given in GiB may appear smaller. For example, 1023 GiB = 1098.4 GB.
- Disk throughput is measured in input/output operations per second (IOPS) and MBps where MBps = 10^6 bytes/sec.
- Data disks can operate in cached or uncached modes. For cached data disk operation, the host cache mode is set to **ReadOnly** or **ReadWrite**. For uncached data disk operation, the host cache

mode is set to **None**.

- To learn how to get the best storage performance for your VMs, see Virtual machine and disk performance.
- **Expected network bandwidth** is the maximum aggregated bandwidth allocated per VM type across all NICs, for all destinations. For more information, see Virtual machine network bandwidth.

  Upper limits aren't guaranteed. Limits offer guidance for selecting the right VM type for the intended application. Actual network performance will depend on several factors including network congestion, application loads, and network settings. For information on optimizing network throughput, see Optimize network throughput for Azure virtual machines. To achieve the expected network performance on Linux or Windows, you may need to select a specific version or optimize your VM. For more information, see Bandwidth/Throughput testing (NTTTCP).

Other sizes and information

- General purpose
- Memory optimized
- Storage optimized
- GPU optimized
- High performance compute
- Previous generations

Pricing Calculator : Pricing Calculator

For more information on disk types, see What disk types are available in Azure?

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Lsv2 sizes series

Summarize this article for me

The Lsv2-series features high throughput, low latency, directly mapped local NVMe storage running on the AMD EPYC™ 7551 processor with an all core boost of 2.55GHz and a max boost of 3.0GHz. The Lsv2-series VMs come in sizes from 8 to 80 vCPU in a simultaneous multi-threading configuration. There is 8 GiB of memory per vCPU, and one 1.92TB NVMe SSD M.2 device per 8 vCPUs, with up to 19.2TB (10x1.92TB) available on the L80s v2.

Host specifications

Expand table

Part	Quantity Count Units	Specs SKU ID, Performance Units, etc.
Processor	8 - 80 vCPUs	AMD EPYC 7551 (Naples) [x86-64]
Memory	64 - 640 GiB	
Local Storage	1 Temp Disk 1 - 10 NVMe Disks	80 - 800 GiB Temp Disks 1.92 TiB NVMe Disks
Remote Storage	16 - 32 Disks	8000 - 80000 IOPS 160 - 1400 MBps
Network	2 - 8 NICs	3200 - 16000+ Mbps
Accelerators	None	

For features supported by this series, see the Feature support section.

Sizes in series

Basics

vCPUs (Qty.) and Memory for each size

Expand table

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_L8s_v2	8	64

Size Name	vCPUs (Qty.)	Memory (GB)
Standard_L16s_v2	16	128
Standard_L32s_v2	32	256
Standard_L48s_v2	48	384
Standard_L64s_v2	64	512
Standard_L80s_v26	80	640

VM Basics resources

- Check vCPU quotas

Feature support

Expand table

Feature name	Support status
Premium Storage	Supported
Premium Storage caching	Not Supported
Live Migration	Not Supported
Memory Preserving Updates	Not Supported
Generation 2 VMs	Supported
Generation 1 VMs	Supported
Accelerated Networking	Supported
Ephemeral OS Disk	Supported
Nested Virtualization	Not Supported

Other size information

List of all available sizes: Sizes

Pricing Calculator: Pricing Calculator

Information on Disk Types: Disk Types

Next steps

Take advantage of the latest performance and features available for your workloads by changing the size of a virtual machine.

Utilize Microsoft's in-house designed ARM processors with Azure Cobalt VMs.

Learn how to Monitor Azure virtual machines.

Last updated on 03/10/2026

Migrate your NP-series virtual machines by May 31, 2027

> **Note**
>
> 1-year and 3-year purchases for the NP-series end April 2, 2026.

Microsoft Azure has announced the retirement of its NP-series virtual machines. The NP-series, introduced for FPGA-accelerated workloads, is powered by Intel Xeon 8171M (Skylake) CPUs and AMD Xilinx Alveo U250 FPGAs. These VMs are designed for accelerating workloads including machine learning inference, video transcoding, and database search & analytics.

How does the retirement of the NP-series virtual machines affect me?

After May 31, 2027, any remaining NP-series virtual machines (VMs) subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. NP-series will be no longer under SLA or have support included.

What action do I need to take before the retirement date?

To avoid service disruption, please resize or deallocate your NP-series VMs and change your VM sizes to a product that meets your workload requirements.

Suggested alternative Azure VM sizes

To ensure continuity and optimal performance, we recommend transitioning to one of the following Azure GPU VM families based on your workload needs:

- NDv2 VMs – designed for the needs of the most demanding GPU-accelerated AI, machine learning, simulation, and HPC workloads.
- NCads_H100_v5 VMs – real-world Azure Applied AI training and batch inference workloads.
- NCasT4_v3 – ideal for deploying AI services, such as real-time inferencing of user-generated requests, or for interactive graphics and visualization workloads.

Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Review the documentation for the series to understand the capabilities and limitations of the VMs in that series.
2. Request quota for the new target VM.
3. You can resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.
2. Under *Subscription*, select your subscription.
3. Under *Service*, click **My services.**
4. Under *Service type*, select **Virtual Machine running Windows/Linux.**
5. Under *Summary*, enter the summary of your request.
6. Under *Problem type*, select **Assistance with resizing my VM.**
7. Under *Problem subtype*, select the option that applies to you.

Last updated on 03/18/2026

Migrate your HC-series virtual machines by May 31, 2027

> **Note**
>
> 1-year and 3-year purchases for the HC-series end April 2, 2026.

Microsoft Azure has announced the retirement of its HC-series virtual machines effective May 31, 2027. Introduced in 2019, HC-series VMs are equipped with Intel Xeon Platinum 8168 processors, 8 GB of RAM per CPU core, and a 100 Gb/sec Mellanox EDR InfiniBand. Since launching this series, Microsoft introduced HBv5, HBv4, HX, and HBv3 series virtual machines. These newer offerings pack the latest technological improvements in compute, memory, and backend networking. This enables superior performance, scaling efficiency, and cost-effectiveness for a wide range of high-performance computing workloads.

Impacted sizes

The following HC-series VM sizes are being retired on May 31, 2027:

- Standard_HC44rs
- Standard_HC44-16rs
- Standard_HC44-32rs

How does the retirement of the HC-series virtual machines affect me?

After May 31, 2027, any remaining HC-series virtual machines (VMs) subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. HC-series will no longer be under SLA or have support included.

What action do I need to take before the retirement date?

To avoid service disruption, deallocate your HC-series VMs and migrate these workloads to newer HPC-optimized SKUs such as HBv5, HBv4, HX, or HBv3 series (or an alternative that

meets your workload requirements).

Suggested alternative Azure VM sizes

To ensure continuity and optimal performance, we recommend transitioning from the current HC-series VMs to newer VM series within Azure's HPC portfolio. HBv5-series offers higher performance and better price-performance than HBv4, making it well-suited for most HPC workloads. HX-series is optimized for high-memory HPC workloads with approximately twice the memory capacity of HBv4.

Expand table

Workload	Recommended VM to migrate to
Memory-bandwidth intensive HPC applications (for example, CFD, structural mechanics, energy research, weather, and climate, chemistry & materials)	HBv5, HBv4, HBv3
Memory capacity/latency bound (for example, silicon design/EDA)	HX, HBv5, HBv3
Compute bound (for example, financial calculations, rendering)	HX, HBv4, HBv3
Intel-specific requirements (for example, Intel MKL, ISV certification on Xeon)	Ddsv6/Edsv6, Ddsv5/Edsv5, Ddsv4/Edsv4

> **Note**
>
> These virtual machine recommendations are listed in order of cost performance benefit to customers from most to least.

Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Review the table above for Microsoft's recommendation. You can also file a support request if more assistance is needed.
2. Request quota for the new target VM.
3. Resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.
2. Under *Subscription*, select your subscription.
3. Under *Service*, select **My services.**
4. Under *Service type*, select **Virtual Machine running Windows/Linux.**
5. Under *Summary*, enter the summary of your request.
6. Under *Problem type*, select **Assistance with resizing my VM.**
7. Under *Problem subtype*, select the option that applies to you.

Last updated on 04/17/2026

Migrate your HBv2-series virtual machines by May 31, 2027

> **Note**
>
> 1-year and 3-year purchases for the HBv2-series end April 2, 2026.

Microsoft Azure has announced the retirement of its HBv2-series virtual machines effective May 31, 2027. Introduced in 2020, the HBv2-series is equipped with 120 AMD EPYC 7V12 processor cores, 4 GB of RAM per CPU core, up to 350 GB/s of memory bandwidth, and up to 4 teraFLOPS of FP64 compute. Since launching this series, Microsoft introduced HBv5, HBv4, HX-series, and HBv3 series virtual machines. These newer offerings pack the latest technological improvements in compute, memory, and backend networking, enabling superior performance, scaling efficiency, and cost-effectiveness for a wide range of high-performance computing workloads.

How does the retirement of the HBv2-series virtual machines affect me?

After May 31, 2027, any remaining HBv2-series virtual machines (VMs) subscriptions will be set to a deallocated state. They'll stop working and no longer incur billing charges. HBv2-series will be no longer under SLA or have support included.

What action do I need to take before the retirement date?

To avoid service disruption, resize or deallocate your HBv2-series VMs and migrate these workloads to newer HPC-optimized SKUs such as HBv5, HBv4, HX, and HBv3-series virtual machines (or an alternative that meets your workload requirements).

Suggested alternative Azure VM sizes

To ensure continuity and optimal performance, we recommend transitioning from the current HBv2-series VMs to newer VM series within Azure's HPC portfolio.

Expand table

Workload	Recommended VM to migrate to
Memory-bandwidth intensive HPC applications (e.g. CFD, structural mechanics, energy research, weather and climate, chemistry & materials)	HBv5, HBv4, HBv3
Memory capacity/latency bound (e.g. silicon design/EDA)	HX, HBv5, HBv3
Compute bound (e.g. financial calculations, rendering)	HX, HBv4, HBv3

> **Note**
>
> These virtual machine recommendations are listed in order of cost performance benefit to customers from most to least.

Check Azure Regions by Product page for region availability. Visit the Azure Virtual Machine pricing page for pricing information.

Steps to change VM size

1. Choose a series and size. Review the table above for Microsoft's recommendation. You can also file a support request if more assistance is needed.
2. Request quota for the new target VM.
3. Resize the virtual machine.

Help and support

If you have a support plan and you need technical help, create a support request.

1. Under *Issue type*, select **Technical**.
2. Under *Subscription*, select your subscription.
3. Under *Service*, select **My services.**
4. Under *Service type*, select **Virtual Machine running Windows/Linux.**
5. Under *Summary*, enter the summary of your request.
6. Under *Problem type*, select **Assistance with resizing my VM.**
7. Under *Problem subtype*, select the option that applies to you.

Last updated on 03/18/2026

Retired Azure VM size series

This article provides a list of all sizes that are retired or have been announced for retirement. For sizes that require it there are migration guides to help move to replacement sizes.

> ⚠ **Warning**
>
> Series with *Retirement Status* listed as *Retired* are **no longer available** and can't be provisioned.
>
> Virtual Machines (VMs) announced for retirement will have restrictions when deploying through new subscriptions. We recommend using the latest generation VMs to ensure best price-performance and to benefit from extended capacity availability over time for new deployments.

What are retired size series?

Retired virtual machine size series are running on older hardware which is no longer supported. The hardware is with newer generations of hardware.

Series with *Retirement Status* listed as *Announced* are still available, but will be retired on the *Planned Retirement Date*. It's recommended that you plan your migration to a replacement series well before the listed retirement date.

To learn more about size series retirement, previous-gen sizes, and the retirement process, see the size series retirement overview.

> ⓘ **Important**
>
> If you're currently using one of the size series listed as *Retired*, view the migration guide to switch to a replacement series as soon as possible.

Previous-gen size series aren't retired and still fully supported, but they have limitations similar to series that are announced for retirement. For a list of previous-gen sizes, see previous generation Azure VM sizes.

General purpose retired sizes

Expand table

Series name	Retirement Status	Retirement Announcement	Planned Retirement Date	Migration Guide
D-series	**Announced**	03/31/25	05/01/28	Migration Guide
Ds-series	**Announced**	03/31/25	05/01/28	Migration Guide
Dv2-series	**Announced**	03/31/25	05/01/28	Migration Guide
Dsv2-series	**Announced**	03/31/25	05/01/28	Migration Guide
Av2/Amv2-series	**Announced**	10/15/25	11/15/28	Migration Guide
B-series	**Announced**	10/15/25	11/15/28	Migration Guide

Compute optimized retired sizes

Expand table

Series name	Retirement Status	Retirement Announcement	Planned Retirement Date	Migration Guide
F-series	**Announced**	10/15/25	11/15/28	Migration Guide
Fs-series	**Announced**	10/15/25	11/15/28	Migration Guide
Fsv2-series	**Announced**	10/15/25	11/15/28	Migration Guide

Memory optimized retired sizes

Expand table

Series name	Retirement Status	Retirement Announcement	Planned Retirement Date	Migration Guide
G-series	**Announced**	10/15/25	11/15/28	Migration Guide

Series name	Retirement Status	Retirement Announcement	Planned Retirement Date	Migration Guide
Gs-series	**Announced**	10/15/25	11/15/28	Migration Guide
Standard_M192idms_v2	**Announced**	03/22/2024	03/31/27	Migration Guide
Standard_M192ids_v2	**Announced**	03/22/2024	03/31/27	Migration Guide
Standard_M192ims_v2	**Announced**	03/22/2024	03/31/27	Migration Guide
Standard_M192is_v2	**Announced**	03/22/2024	03/31/27	Migration Guide

Storage optimized retired sizes

Expand table

Series name	Retirement Status	Retirement Announcement	Planned Retirement Date	Migration Guide
Ls-series	**Announced**	03/31/25	05/01/28	Migration Guide
Lsv2-series	**Announced**	10/15/25	11/15/28	Migration Guide

GPU accelerated retired sizes

Expand table

Series name	Retirement Status	Retirement Announcement	Planned Retirement Date	Migration Guide
NCv3-NC24rs Series	**Retired**	-	30/9/25	NCv3-NC24rs-series Retirement
NCv3-Series	**Retired**	-	30/9/25	NCv3-series Retirement

FPGA accelerated retired sizes

Currently there are no retired FPGA accelerated series retired or announced for retirement.

HPC retired sizes

Currently there are no retired HPC series retired or announced for retirement.

ADH retired sizes

Currently there are no retired ADH series retired or announced for retirement.

Next steps

- For a list of older and capacity limited sizes, see Previous generation Azure VM sizes.
- For more information on VM sizes, see Sizes for virtual machines in Azure.

Last updated on 12/11/2025

Support for Generation 2 VMs on Azure

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Support for Generation 2 virtual machines (VMs) is now available on Azure. You can also upgrade existing Generation 1 virtual machines to Generation 2 with Trusted launch. Review the considerations on this page and the upgrade guidance before choosing or upgrading a generation.

Generation 2 VMs support key features that aren't supported in Generation 1 VMs. These features include increased memory, Intel Software Guard Extensions (Intel SGX), and virtualized persistent memory (vPMEM). Generation 2 VMs running on-premises, have some features that aren't supported in Azure yet. For more information, see the Features and capabilities section.

Generation 2 VMs use the new UEFI-based boot architecture rather than the BIOS-based architecture used by Generation 1 VMs. Compared to Generation 1 VMs, Generation 2 VMs might have improved boot and installation times. For an overview of Generation 2 VMs and some of the differences between Generation 1 and Generation 2, see Should I create a Generation 1 or 2 virtual machine in Hyper-V?.

Generation 2 VM sizes

Azure now offers Generation 2 support for the following selected VM series:

Expand table

Type	Generation 2 supported size families	Not supported size families
General purpose	B-family, D-family	A-family
Compute optimized	F-family, Fx-family	
Memory optimized	E-family, Eb-family, M-family	
Storage optimized	L-family	
GPU	NC-family, ND-family, NV-family	NC-series, NV-series
High Performance Compute	HBv2-series, HBv3-series, HBv4-series, HC-series, HX-series	

[1] Mv2-series, DC-series, NDv2-series, Msv2 and Mdsv2-series Medium Memory do not support Generation 1 VM images and only support a subset of Generation 2 images. Please see Mv2-

series documentation, ND A100 v4-series, NDv2-series, and Msv2 and Mdsv2 Medium Memory Series for details.

Generation 2 VM images in Azure Marketplace

Generation 2 VMs support the following Marketplace images:

- Windows Server 2025, 2022, 2019, 2016, 2012 R2, 2012
- Windows 11 Pro, Windows 11 Enterprise
- Windows 10 Pro, Windows 10 Enterprise
- SUSE Linux Enterprise Server 15 SP3, SP2
- SUSE Linux Enterprise Server 12 SP4
- Ubuntu Server 22.04 LTS, 20.04 LTS, 18.04 LTS, 16.04 LTS
- RHEL 9,5, 9.4, 9.3, 9.2, 9.1, 9.0, 8.10, 8.9, 8.8, 8.7, 8.6, 8.5, 8.4, 8.3, 8.2, 8.1, 8.0, 7.9, 7.8, 7.7, 7.6, 7.5, 7.4, 7.0
- Cent OS 8.4, 8.3, 8.2, 8.1, 8.0, 7.7, 7.6, 7.5, 7.4
- Oracle Linux 9.3, 9.2, 9.1, 9.0, 8.9, 8.8, 8.7, 8.6, 8.5, 8.4, 8.3, 8.2, 8.1, 7.9, 7.9, 7.8, 7.7

> **Note**
>
> Specific Virtual machine sizes like Mv2-Series, DC-series, ND A100 v4-series, NDv2-series, Msv2 and Mdsv2-series may only support a subset of these images - please look at the relevant virtual machine size documentation for complete details.

On-premises vs. Azure Generation 2 VMs

Azure doesn't currently support some of the features that on-premises Hyper-V supports for Generation 2 VMs.

Expand table

Generation 2 feature	On-premises Hyper-V	Azure
Secure boot	✔	With trusted launch
Shielded VM	✔	✘
vTPM	✔	With trusted launch
Virtualization-based security (VBS)	✔	✔
VHDX format	✔	✘

For more information, see Trusted launch.

Features and capabilities

Generation 1 vs. Generation 2 features

Expand table

Feature	Generation 1	Generation 2
Boot	PCAT	UEFI
Disk controllers	IDE	SCSI
VM sizes	All VM sizes	See available sizes

Generation 1 vs. Generation 2 capabilities

Expand table

Capability	Generation 1	Generation 2
OS disk > 2 TB	✗	✓
Custom disk/image/swap OS	✓	✓
Virtual machine scale set support	✓	✓
Azure Site Recovery	✓	✓
Backup/restore	✓	✓
Azure Compute Gallery	✓	✓
Azure disk encryption	✓	✓
Server-side encryption	✓	✓

Creating a Generation 2 VM

Azure Resource Manager Template

To create a simple Windows Generation 2 VM, see Create a Windows virtual machine from a Resource Manager template To create a simple Linux Generation 2 VM, see How to create a

Linux virtual machine with Azure Resource Manager templates

Marketplace image

In the Azure portal or Azure CLI, you can create Generation 2 VMs from a Marketplace image that supports UEFI boot.

Azure portal

Below are the steps to create a Generation 2 (Gen2) VM in Azure portal.

1. Sign in to the Azure portal.
2. Search for **Virtual Machines**
3. Under **Services**, select **Virtual machines**.
4. In the **Virtual machines** page, select **Add**, and then select **Virtual machine**.
5. Under **Project details**, make sure the correct subscription is selected.
6. Under **Resource group**, select **Create new** and type a name for your resource group or select an existing resource group from the dropdown.
7. Under **Instance details**, type a name for the virtual machine name and choose a region
8. Under **Image**, select a Generation 2 image from the **Marketplace images to get started**

   > **Tip**
   >
   > If you don't see the Generation 2 version of the image you want in the drop-down, select **See all images** and then change the **Image Type** filter to **Gen 2**.

9. Select a VM size that supports Generation 2. See a list of supported sizes.
10. Fill in the **Administrator account** information and then **Inbound port rules**
11. At the bottom of the page, select **Review + Create**
12. On the **Create a virtual machine** page, you can see the details about the VM you are about to deploy. Once validation shows as passed, select **Create**.

PowerShell

You can also use PowerShell to create a VM by directly referencing the Generation 1 or Generation 2 SKU.

For example, use the following PowerShell cmdlet to get a list of the SKUs in the `WindowsServer` offer.

```
PowerShell
```

```
Get-AzVMImageSku -Location westus2 -PublisherName MicrosoftWindowsServer -Offer
WindowsServer
```

If you're creating a VM with Windows Server 2019 as the OS, then you can select a Generation 2 (UEFI) image which looks like this:

PowerShell

```
2019-datacenter-gensecond
```

If you're creating a VM with Windows 10 as the OS, then you can select a Generation 2 (UEFI) image which looks like this:

PowerShell

```
20H2-PRO-G2
```

See the Features and capabilities section for a current list of supported Marketplace images.

Azure CLI

Alternatively, you can use the Azure CLI to see any available Generation 2 images, listed by **Publisher**.

Azure CLI

```
az vm image list --publisher Canonical --sku gen2 --output table --all
```

Managed image or managed disk

You can create a Generation 2 VM from a managed image or managed disk in the same way you would create a Generation 1 VM.

Virtual machine scale sets

You can also create Generation 2 VMs by using virtual machine scale sets. In the Azure CLI, use Azure scale sets to create Generation 2 VMs.

> **Note**

Alternatively, you can upgrade an existing Generation 1 VM to Generation 2 with Trusted launch. For more information, see Upgrade existing Gen1 VMs to Gen2-Trusted launch.

Frequently asked questions

- **Are Generation 2 VMs available in all Azure regions?**
  Yes. But not all generation 2 VM sizes are available in every region. The availability of the Generation 2 VM depends on the availability of the VM size.

- **Is there a price difference between Generation 1 and Generation 2 VMs?**
  No.

- **I have a .vhd file from my on-premises Generation 2 VM. Can I use that .vhd file to create a Generation 2 VM in Azure?** Yes, you can bring your Generation 2 .vhd file to Azure and use that to create a Generation 2 VM. Use the following steps to do so:

  1. Upload the .vhd to a storage account in the same region where you'd like to create your VM.

  2. Create a managed disk from the .vhd file. Set the Hyper-V Generation property to V2. The following PowerShell commands set Hyper-V Generation property when creating managed disk.

     PowerShell

     ```
     $sourceUri = 'https://xyzstorage.blob.core.windows.net/vhd/abcd.vhd'. #
     <Provide location to your uploaded .vhd file>
     $osDiskName = 'gen2Diskfrmgenvhd'  #<Provide a name for your disk>
     $diskconfig = New-AzDiskConfig -Location '<location>' -DiskSizeGB 127 -
     AccountType Standard_LRS -OsType Windows -HyperVGeneration "V2" -SourceUri
     $sourceUri -CreateOption 'Import'
     New-AzDisk -DiskName $osDiskName -ResourceGroupName '<Your Resource
     Group>' -Disk $diskconfig
     ```

  3. Once the disk is available, create a VM by attaching this disk. The VM created will be a Generation 2 VM. When the Generation 2 VM is created, you can optionally generalize the image of this VM. By generalizing the image, you can use it to create multiple VMs.

- **How do I increase the OS disk size?**

  OS disks larger than 2 TiB are new to Generation 2 VMs. By default, OS disks are smaller than 2 TiB for Generation 2 VMs. You can increase the disk size up to a recommended maximum of 4 TiB. Use the Azure CLI or the Azure portal to increase the OS disk size. For

information about how to expand disks programmatically, see **Resize a disk** for Windows or Linux.

To increase the OS disk size from the Azure portal:

1. In the Azure portal, go to the VM properties page.
2. To shut down and deallocate the VM, select the **Stop** button.
3. In the **Disks** section, select the OS disk you want to increase.
4. In the **Disks** section, select **Configuration**, and update the **Size** to the value you want.
5. Go back to the VM properties page and **Start** the VM.

You might see a warning for OS disks larger than 2 TiB. The warning doesn't apply to Generation 2 VMs. However, OS disk sizes larger than 4 TiB are not supported.

- **Do Generation 2 VMs support accelerated networking?**
  Yes. For more information, see Create a VM with accelerated networking.
- **Do Generation 2 VMs support Secure Boot or vTPM in Azure?** Both vTPM and Secure Boot are features of trusted launch for Generation 2 VMs. For more information, see Trusted launch.
- **Is VHDX supported on Generation 2?**
  No, Generation 2 VMs on Azure support only VHD.
- **Do Generation 2 VMs support Azure Ultra Disk Storage?**
  Yes.
- **Can I migrate a VM from Generation 1 to Generation 2?**
  Azure Virtual Machines supports upgrading existing Generation 1 VMs to Generation 2 by upgrading to the Trusted launch security type. See Upgrade existing Gen1 VMs to Gen2-Trusted launch for steps and prerequisites.
- **Why is my VM size not enabled in the size selector when I try to create a Generation 2 VM?**

  This may be solved by doing the following:

  1. Verify that the **VM Generation** property is set to **Gen 2**.
  2. Verify you are searching for a VM size which supports Generation 2 VMs.

Next steps

Learn more about the trusted launch with Generation 2 VMs.

Learn about Generation 2 virtual machines in Hyper-V.

Upgrade an existing Gen1 VM to Gen2-Trusted launch.

Last updated on 12/05/2025

Virtual machine isolation in Azure

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Compute offers virtual machine sizes that are isolated to a specific hardware type and dedicated to a single customer. Isolated sizes run on specific hardware generations. They're deprecated when the hardware generation is retired or a new generation becomes available.

Isolated virtual machine sizes are best suited for workloads that require a high degree of isolation from other customers' workloads. This requirement sometimes applies to compliance and regulatory standards. Utilizing an isolated size guarantees that your virtual machine is the only one running on that specific server instance.

Additionally, as the isolated size virtual machines are large, customers may choose to subdivide the resources of these virtual machines by using Azure support for nested virtual machines.

Isolated VM sizes

Expand table

Family	Isolated Sizes
E family	Standard_E192is_v6 Standard_E192ids_v6 Standard_E104i_v5 Standard_E104id_v5 Standard_E104is_v5 Standard_E104ids_v5 Standard_E112ias_v5 Standard_E112iads_v5 Standard_E80is_v4 Standard_E80ids_v4 Standard_E96ias_v4
Eb family	Standard_E112ibs_v5 Standard_E112ibds_v5
EC family	Standard_EC96ias_v5 Standard_EC96iads_v5
M family	Standard_M832is_16_v3 Standard_M832ids_16_v3 Standard_M192is_v2[1] Standard_M192ids_v2[1] Standard_M192ims_v2[1] Standard_M192idms_v2[1]

Family	Isolated Sizes
NC family	Standard_NC80adis_H100_v5
ND family	Standard_ND128isr_NDR_GB200_v6 Standard_ND128isr_NDR_GB300_v6 Standard_ND96isr_H100_v5 Standard_ND96isr_H200_v5 Standard_ND96isr_MI300X_v5

[1] These sizes are announced for retirement

> **Note**
>
> Isolated VM sizes have a limited lifespan that is tied to the lifetime of the hardware they run on.

Retired Isolated VM Sizes

Isolated VM sizes have a hardware limited lifespan. Azure issues reminders 12 months in advance of the official deprecation date of the sizes and provides an updated isolated offering for your consideration. The following sizes are retired or announced for retirement.

Expand table

Size	Retirement Announcement	Isolation Retirement Date	Migration Guide
Standard_DS15_v2	11/22/2019	05/15/2020	Migration Guide
Standard_D15_v2	11/22/2019	05/15/2020	Migration Guide
Standard_G5	02/22/2021	02/28/2022	Migration Guide
Standard_GS5	02/22/2021	02/28/2022	Migration Guide
Standard_E64i_v3	02/22/2021	02/28/2022	Migration Guide
Standard_E64is_v3	02/22/2021	02/28/2022	Migration Guide
Standard_M192is_v2	03/22/2024	03/31/2027	Migration Guide
Standard_M192ims_v2	03/22/2024	03/31/2027	Migration Guide
Standard_M192ids_v2	03/22/2024	03/31/2027	Migration Guide
Standard_M192idms_v2	03/22/2024	03/31/2027	Migration Guide

FAQ

Q: What specific scenarios benefit most from isolated VM sizes?

A: Isolated VM sizes run on dedicated physical servers for maximum security and consistency. These sizes are best for regulated workloads, highly sensitive data, and performance-critical systems.

Q: How are isolated VMs different from Azure Dedicated Hosts and Confidential VMs?

A: Isolated VMs run on a single physical server dedicated exclusively for one VM for strict compliance and isolation. Dedicated Hosts reserve an entire physical host for multiple VMs belonging to a single customer, giving full control on VM placement and maintenance scheduling. Confidential VMs use hardware-based encryption to protect data without requiring physical isolation.

Q: Is the size going to be retired or only its "isolation" feature?

A: If a VM sizes is published as isolated but doesn't have an "i" in its name, only the isolation feature is being retired (unless stated otherwise). VM sizes with an "i" in the name are deprecated entirely. We provide reminders 12 months in advance of the official deprecation of the isolated size.

Q: Is there downtime if my VM moves to nonisolated hardware?

A: For sizes where only isolation is retired (not the size), no action is needed and there's no downtime. If isolated is required, the announcement will include a recommended replacement size. Moving to that sizes requires resizing your VM.

Q: Is there any cost difference when moving from an isolated VM to a non-isolated VM?

A: Yes, isolated VM sizes carry a premium compared to non-isolated sizes. The premium is roughly 10% higher for isolated VMs

Q: What should I do if the isolated VM size I need is not available in my region?

A: Choose an alternative isolated SKU with similar specs or deploy to one of the supported regions.

Next steps

Customers can also choose to further subdivide the resources of these isolated virtual machines by using Azure support for nested virtual machines.

Last updated on 01/08/2026

Compute benchmark scores for Linux VMs

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

> **Note**
>
> This page is no longer being maintained and is planned for archival on 12/16/2024.
>
> Azure is no longer publishing CoreMark since the metric has limited ability to inform users of the expected performance of a virtual machine across various attributes. For the most accurate results on a specific virtual machine, Azure recommends users run their workload(s) on that virtual machine to verify performance.

The following CoreMark benchmark scores show compute performance for Azure's high-performance VM lineup running Ubuntu 18.04. Compute benchmark scores are also available for Windows VMs.

Azure (Coremark) TOC

Expand table

Type	Families
Compute optimized	Fsv2, FXMDVS
General purpose	B, DADSv5, DASv5, DDSv5, DDv5, DSv5, Dv5, Dasv4, Dav4, Ddsv4, Ddv4, Dsv4, Dv4, Dsv3, Dv3, DSv2, Dv2
High performance compute	HBv3, HBv2, HB, HC
Memory optimized	EADSv5, EASv5, EDSv5, EDv5, ESv5, Ev5, Easv4, Eav4, Edsv4, Edv4, Esv4, Ev4, DSv2, Esv3 Ev3, Dv2, Msv2, Ms
Storage optimized	Lsv2
Confidential Compute	DCS, DCSv3, DCDSv3, DCv2

Compute optimized

Fsv2 - Compute + Premium Storage

(10/10/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F2s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	2	1	4.0	35,925	603	1.68%	308
Standard_F2s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	4.0	35,659	344	0.96%	112
Standard_F4s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	4	1	8.0	65,819	851	1.29%	245
Standard_F4s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	8.0	65,683	459	0.70%	98
Standard_F8s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	8	1	16.0	136,027	1,272	0.94%	238
Standard_F8s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	16.0	135,829	912	0.67%	91
Standard_F16s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	16	1	32.0	271,369	1,955	0.72%	280

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F16s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	32.0	271,236	1,325	0.49%	105
Standard_F32s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	32	1	64.0	538,734	3,795	0.70%	245
Standard_F32s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	64.0	537,915	7,646	1.42%	91
Standard_F48s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	48	2	96.0	780,596	8,712	1.12%	182
Standard_F48s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	96.0	749,662	21,751	2.90%	56
Standard_F64s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	64	2	128.0	1,035,424	11,557	1.12%	175
Standard_F64s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	128.0	1,023,867	17,292	1.69%	63
Standard_F72s_v2	Intel(R) Xeon(R)	72	2	144.0	1,126,078	12,094	1.07%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8168 CPU @ 2.70GHz							
Standard_F72s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	72	2	144.0	1,120,116	15,662	1.40%	42

FXMDVS

(04/12/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_FX4mds	Intel(R) Xeon(R) Gold 6246R CPU @ 3.40GHz	4	1	82.5	77,794	711	0.91%	42
Standard_FX12mds	Intel(R) Xeon(R) Gold 6246R CPU @ 3.40GHz	12	1	247.8	228,808	827	0.36%	42
Standard_FX24mds	Intel(R) Xeon(R) Gold 6246R CPU @ 3.40GHz	24	1	495.8	475,203	2,142	0.45%	42
Standard_FX36mds	Intel(R) Xeon(R) Gold 6246R CPU @ 3.40GHz	36	2	744.1	678,947	10,647	1.57%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_FX48mds	Intel(R) Xeon(R) Gold 6246R CPU @ 3.40GHz	48	2	992.1	899,165	12,000	1.33%	42

General purpose

B - Burstable

(09/24/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_B1s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	1.0	18,768	429	2.29%	35
Standard_B1s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	1.0	19,725	1,258	6.38%	112
Standard_B1s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	1.0	18,287	2,316	12.66%	70
Standard_B1s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	1.0	21,992	371	1.69%	21
Standard_B1ls	Intel(R) Xeon(R)	1	1	0.5	19,162	605	3.16%	28

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E5-2673 v3 @ 2.40GHz							
Standard_B1ls	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	0.5	19,365	810	4.18%	126
Standard_B1ls	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	0.5	18,780	2,840	15.12%	63
Standard_B1ls	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	0.5	21,954	461	2.10%	21
Standard_B1ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	2.0	18,167	82	0.45%	14
Standard_B1ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	2.0	20,113	1,188	5.90%	126
Standard_B1ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	2.0	18,882	2,434	12.89%	77
Standard_B1ms	Intel(R) Xeon(R) Platinum 8272CL	1	1	2.0	22,182	57	0.26%	21

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_B2s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	4.0	36,074	431	1.20%	21
Standard_B2s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	4.0	37,634	1,445	3.84%	119
Standard_B2s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	4.0	37,824	5,913	15.63%	70
Standard_B2s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	4.0	44,147	397	0.90%	28
Standard_B2ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	8.0	36,338	333	0.92%	28
Standard_B2ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	8.0	38,026	1,749	4.60%	119
Standard_B2ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	8.0	38,060	5,978	15.71%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_B2ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	44,227	349	0.79%	21
Standard_B4ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	16.0	70,362	3,310	4.70%	28
Standard_B4ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	16.0	74,405	2,647	3.56%	112
Standard_B4ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	16.0	71,174	8,008	11.25%	77
Standard_B4ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	85,894	1,445	1.68%	21
Standard_B8ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	32.0	144,753	3,091	2.14%	35
Standard_B8ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	32.0	145,027	1,460	1.01%	98
Standard_B8ms	Intel(R) Xeon(R)	8	1	32.0	138,848	11,991	8.64%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8171M CPU @ 2.60GHz							
Standard_B8ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	170,427	2,402	1.41%	14
Standard_B12ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	12	1	48.0	214,731	4,672	2.18%	28
Standard_B12ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	12	1	48.0	219,383	2,213	1.01%	112
Standard_B12ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	12	1	48.0	202,912	17,172	8.46%	77
Standard_B12ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	12	1	48.0	256,373	1,768	0.69%	21
Standard_B16ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	64.0	276,669	8,921	3.22%	35
Standard_B16ms	Intel(R) Xeon(R) CPU E5-2673 v4	16	1	64.0	294,079	3,437	1.17%	112

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_B16ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	64.0	270,563	27,650	10.22%	70
Standard_B16ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	340,582	6,378	1.87%	21
Standard_B20ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	2	80.0	347,156	5,967	1.72%	14
Standard_B20ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	80.0	363,209	3,837	1.06%	112
Standard_B20ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	80.0	342,212	21,775	6.36%	91
Standard_B20ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	80.0	416,103	3,337	0.80%	21

DADSv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2ads_v5	AMD EPYC 7763 64-Core Processor	2	1	7.8	38,919	41	0.10%	35
Standard_D4ads_v5	AMD EPYC 7763 64-Core Processor	4	1	15.6	72,644	172	0.24%	35
Standard_D8ads_v5	AMD EPYC 7763 64-Core Processor	8	1	31.4	153,951	552	0.36%	35
Standard_D16ads_v5	AMD EPYC 7763 64-Core Processor	16	1	62.8	306,800	941	0.31%	35
Standard_D32ads_v5	AMD EPYC 7763 64-Core Processor	32	1	125.8	600,925	7,721	1.28%	35
Standard_D48ads_v5	AMD EPYC 7763 64-Core Processor	48	1	188.7	893,740	12,161	1.36%	35
Standard_D64ads_v5	AMD EPYC 7763 64-Core Processor	64	1	251.7	1,195,169	16,506	1.38%	35
Standard_D96ads_v5	AMD EPYC 7763 64-Core Processor	96	2	377.9	1,831,129	20,839	1.14%	35

DASv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2as_v5	AMD EPYC 7763 64-Core Processor	2	1	7.8	38,869	118	0.30%	35
Standard_D4as_v5	AMD EPYC 7763 64-Core Processor	4	1	15.6	72,928	460	0.63%	35
Standard_D8as_v5	AMD EPYC 7763 64-Core Processor	8	1	31.4	153,842	462	0.30%	35
Standard_D16as_v5	AMD EPYC 7763 64-Core Processor	16	1	62.8	304,560	3,347	1.10%	35
Standard_D32as_v5	AMD EPYC 7763 64-Core Processor	32	1	125.8	599,269	8,844	1.48%	35
Standard_D48as_v5	AMD EPYC 7763 64-Core Processor	48	1	188.7	896,034	12,918	1.44%	35
Standard_D64as_v5	AMD EPYC 7763 64-Core Processor	64	1	251.7	1,195,829	16,444	1.38%	35
Standard_D96as_v5	AMD EPYC 7763 64-	96	2	377.9	1,833,797	20,117	1.10%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							

DDSv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	7.8	34,926	11	0.03%	56
Standard_D4ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	15.6	68,673	207	0.30%	56
Standard_D8ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	31.4	136,764	489	0.36%	49
Standard_D16ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	62.8	273,303	1,122	0.41%	56
Standard_D32ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	125.8	545,658	2,409	0.44%	49
Standard_D48ds_v5	Intel(R) Xeon(R) Platinum	48	1	188.7	813,359	5,923	0.73%	49

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8370C CPU @ 2.80GHz							
Standard_D64ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	251.9	1,061,667	10,151	0.96%	35
Standard_D96ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	377.9	1,577,187	17,287	1.10%	56

DDv5

(03/11/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	7.8	34,923	16	0.05%	70
Standard_D4d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	15.6	68,696	234	0.34%	56
Standard_D8d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	31.4	136,791	496	0.36%	42
Standard_D16d_v5	Intel(R) Xeon(R)	16	1	62.8	273,463	1,085	0.40%	49

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8370C CPU @ 2.80GHz							
Standard_D32d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	125.8	544,718	2,672	0.49%	49
Standard_D48d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	1	188.7	812,195	7,150	0.88%	56
Standard_D64d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	251.9	1,061,317	11,637	1.10%	42
Standard_D96d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	377.9	1,579,691	19,962	1.26%	49

DSv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	7.8	32,093	2,789	8.69%	63

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D4s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	15.6	67,114	3,816	5.69%	42
Standard_D8s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	31.4	129,302	9,365	7.24%	49
Standard_D16s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	62.8	250,482	19,982	7.98%	49
Standard_D32s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	125.8	500,612	33,756	6.74%	42
Standard_D48s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	1	188.7	725,001	39,805	5.49%	35
Standard_D64s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	251.9	965,147	62,581	6.48%	42
Standard_D96s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	377.9	1,422,950	79,151	5.56%	35

Dv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	7.8	29,597	565	1.91%	49
Standard_D4_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	15.6	66,338	3,932	5.93%	56
Standard_D8_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	31.4	121,070	8,218	6.79%	245
Standard_D16_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	62.8	247,579	18,327	7.40%	42
Standard_D32_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	125.8	475,867	11,659	2.45%	35
Standard_D48_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	1	188.7	733,540	43,528	5.93%	35
Standard_D64_v5	Intel(R) Xeon(R) Platinum 8370C	64	2	251.9	924,146	9,983	1.08%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.80GHz							
Standard_D96_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	377.9	1,378,842	14,644	1.06%	35

Dasv4

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2as_v4	AMD EPYC 7452 32-Core Processor	2	1	8.0	38,230	313	0.82%	77
Standard_D4as_v4	AMD EPYC 7452 32-Core Processor	4	1	16.0	71,707	790	1.10%	98
Standard_D8as_v4	AMD EPYC 7452 32-Core Processor	8	1	32.0	151,474	1,599	1.06%	56
Standard_D16as_v4	AMD EPYC 7452 32-Core Processor	16	2	64.0	292,074	9,782	3.35%	91
Standard_D32as_v4	AMD EPYC 7452 32-Core Processor	32	4	128.0	575,465	14,574	2.53%	98

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D48as_v4	AMD EPYC 7452 32-Core Processor	48	6	192.0	843,316	21,739	2.58%	105
Standard_D64as_v4	AMD EPYC 7452 32-Core Processor	64	8	256.0	1,120,467	30,714	2.74%	7

Dav4

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2a_v4	AMD EPYC 7452 32-Core Processor	2	1	8.0	38,061	1,219	3.20%	119
Standard_D4a_v4	AMD EPYC 7452 32-Core Processor	4	1	16.0	70,996	2,280	3.21%	140
Standard_D8a_v4	AMD EPYC 7452 32-Core Processor	8	1	32.0	151,379	1,946	1.29%	147
Standard_D16a_v4	AMD EPYC 7452 32-Core Processor	16	2	64.0	293,654	6,263	2.13%	161
Standard_D32a_v4	AMD EPYC 7452 32-	32	4	128.0	571,897	16,009	2.80%	147

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_D48a_v4	AMD EPYC 7452 32-Core Processor	48	6	192.0	837,332	22,923	2.74%	161
Standard_D64a_v4	AMD EPYC 7452 32-Core Processor	64	8	256.0	1,126,793	29,148	2.59%	21
Standard_D96a_v4	AMD EPYC 7452 32-Core Processor	96	12	384.0	1,590,434	31,887	2.00%	14

DDSv4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	35,557	740	2.08%	189
Standard_D4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	65,958	788	1.19%	189
Standard_D8ds_v4	Intel(R) Xeon(R) Platinum 8272CL	8	1	32.0	135,907	1,108	0.81%	189

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_D16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	271,137	1,374	0.51%	189
Standard_D32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	540,212	4,954	0.92%	189
Standard_D48ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	756,538	15,048	1.99%	154
Standard_D48ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	775,291	11,776	1.52%	35
Standard_D64ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	1,025,017	14,482	1.41%	182

DDv4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2d_v4	Intel(R) Xeon(R) Platinum	2	1	8.0	35,640	371	1.04%	189

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8272CL CPU @ 2.60GHz							
Standard_D4d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	65,767	590	0.90%	189
Standard_D8d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	135,756	901	0.66%	189
Standard_D16d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	271,216	1,630	0.60%	189
Standard_D32d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	540,052	4,426	0.82%	189
Standard_D48d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	744,246	20,864	2.80%	147
Standard_D48d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	769,702	10,800	1.40%	42
Standard_D64d_v4	Intel(R) Xeon(R) Platinum 8272CL	64	2	256.0	1,021,118	15,896	1.56%	189

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							

Dsv4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	35,794	578	1.61%	175
Standard_D4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	65,910	668	1.01%	182
Standard_D8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	135,542	835	0.62%	182
Standard_D16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	271,053	1,429	0.53%	182
Standard_D32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	539,271	8,236	1.53%	182
Standard_D48s_v4	Intel(R) Xeon(R) Platinum	48	1	192.0	743,539	18,204	2.45%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8272CL CPU @ 2.60GHz							
Standard_D48s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	772,877	9,466	1.22%	35
Standard_D64s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	1,024,799	15,499	1.51%	182

Dv4

(09/21/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	35,651	228	0.64%	182
Standard_D4_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	65,896	647	0.98%	175
Standard_D8_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	135,803	963	0.71%	182
Standard_D16_v4	Intel(R) Xeon(R)	16	1	64.0	271,203	1,345	0.50%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8272CL CPU @ 2.60GHz							
Standard_D32_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	539,406	5,745	1.07%	182
Standard_D48_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	745,137	17,607	2.36%	154
Standard_D48_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	771,269	7,936	1.03%	28
Standard_D64_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	1,020,266	19,814	1.94%	182

DSv3 - General Compute + Premium Storage

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	8.0	26,323	1,194	4.54%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	8.0	27,724	2,091	7.54%	84
Standard_D2s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	8.0	26,916	1,424	5.29%	91
Standard_D2s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	31,342	18	0.06%	21
Standard_D4s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	16.0	50,716	1,058	2.09%	35
Standard_D4s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	16.0	52,445	3,515	6.70%	77
Standard_D4s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	16.0	50,029	3,332	6.66%	91
Standard_D4s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	57,991	277	0.48%	35
Standard_D8s_v3	Intel(R) Xeon(R)	8	1	32.0	99,350	179	0.18%	7

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E5-2673 v3 @ 2.40GHz							
Standard_D8s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	32.0	101,053	7,572	7.49%	126
Standard_D8s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	32.0	102,485	3,143	3.07%	56
Standard_D8s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	120,003	546	0.46%	35
Standard_D16s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	1	64.0	199,434	1,886	0.95%	35
Standard_D16s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	64.0	196,242	5,336	2.72%	77
Standard_D16s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	64.0	200,086	3,164	1.58%	98
Standard_D16s_v3	Intel(R) Xeon(R) Platinum 8272CL	16	1	64.0	239,837	1,457	0.61%	28

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_D32s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	32	2	128.0	389,200	3,438	0.88%	35
Standard_D32s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	1	128.0	390,983	1,845	0.47%	91
Standard_D32s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	128.0	397,239	2,996	0.75%	77
Standard_D32s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	478,885	1,968	0.41%	35
Standard_D48s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	48	2	192.0	571,460	4,725	0.83%	112
Standard_D48s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	48	1	192.0	581,930	3,357	0.58%	63
Standard_D48s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	48	2	192.0	578,181	5,736	0.99%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D48s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	698,208	8,889	1.27%	28
Standard_D64s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	64	2	256.0	761,965	4,971	0.65%	49
Standard_D64s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	64	2	256.0	767,645	10,497	1.37%	154
Standard_D64s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	909,271	7,791	0.86%	28

Dv3 - General Compute

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	8.0	25,478	977	3.84%	91
Standard_D2_v3	Intel(R) Xeon(R) CPU E5-2673 v4	2	1	8.0	26,601	1,381	5.19%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_D2_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	8.0	27,116	1,688	6.23%	70
Standard_D2_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	31,352	22	0.07%	7
Standard_D4_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	16.0	49,708	2,668	5.37%	77
Standard_D4_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	16.0	51,650	3,799	7.35%	91
Standard_D4_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	16.0	50,025	2,210	4.42%	70
Standard_D8_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	32.0	100,422	1,323	1.32%	98
Standard_D8_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	32.0	97,861	5,903	6.03%	91

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D8_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	32.0	100,826	1,597	1.58%	42
Standard_D8_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	120,063	559	0.47%	7
Standard_D16_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	1	64.0	198,362	2,228	1.12%	84
Standard_D16_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	64.0	193,498	2,080	1.07%	77
Standard_D16_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	64.0	199,683	1,862	0.93%	63
Standard_D16_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	239,145	1,767	0.74%	7
Standard_D32_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	32	2	128.0	387,724	3,770	0.97%	63
Standard_D32_v3	Intel(R) Xeon(R)	32	1	128.0	388,466	9,936	2.56%	112

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E5-2673 v4 @ 2.30GHz							
Standard_D32_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	128.0	397,605	2,423	0.61%	49
Standard_D32_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	479,158	1,356	0.28%	14
Standard_D48_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	48	2	192.0	569,331	6,445	1.13%	140
Standard_D48_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	48	1	192.0	578,441	6,615	1.14%	70
Standard_D48_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	48	2	192.0	576,459	7,500	1.30%	21
Standard_D48_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	693,147	8,980	1.30%	7
Standard_D64_v3	Intel(R) Xeon(R) CPU E5-2673 v4	64	2	256.0	762,667	5,489	0.72%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_D64_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	64	2	256.0	766,739	7,516	0.98%	189
Standard_D64_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	916,549	8,944	0.98%	7

DSv2 - General Purpose + Premium Storage

(09/30/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	3.5	18,952	876	4.62%	49
Standard_DS1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	3.5	20,066	937	4.67%	70
Standard_DS1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	3.5	19,721	1,686	8.55%	98
Standard_DS1_v2	Intel(R) Xeon(R) Platinum	1	1	3.5	22,210	25	0.11%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8272CL CPU @ 2.60GHz							
Standard_DS2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	7.0	37,635	977	2.60%	70
Standard_DS2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	7.0	39,784	2,446	6.15%	84
Standard_DS2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	7.0	37,761	1,322	3.50%	70
Standard_DS2_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	7.0	44,728	39	0.09%	7
Standard_DS3_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	14.0	73,128	1,720	2.35%	70
Standard_DS3_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	14.0	75,583	4,279	5.66%	70
Standard_DS3_v2	Intel(R) Xeon(R) Platinum 8171M	4	1	14.0	72,459	1,230	1.70%	91

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_DS4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	28.0	146,804	1,270	0.86%	49
Standard_DS4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	28.0	147,510	3,214	2.18%	98
Standard_DS4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	28.0	144,002	1,731	1.20%	70
Standard_DS4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	28.0	116,687	59,050	50.61%	14
Standard_DS5_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	56.0	284,713	4,702	1.65%	63
Standard_DS5_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	56.0	297,550	1,502	0.50%	70
Standard_DS5_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	56.0	287,238	4,723	1.64%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS5_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	56.0	345,144	2,027	0.59%	14

Dv2 - General Compute

(09/30/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	3.5	19,112	851	4.45%	105
Standard_D1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	3.5	21,866	1,802	8.24%	49
Standard_D1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	3.5	19,641	808	4.11%	63
Standard_D2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	7.0	36,684	1,542	4.20%	112
Standard_D2_v2	Intel(R) Xeon(R) CPU E5-2673 v4	2	1	7.0	40,452	2,922	7.22%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_D2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	7.0	37,109	1,525	4.11%	63
Standard_D3_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	14.0	72,681	1,424	1.96%	84
Standard_D3_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	14.0	76,631	4,948	6.46%	70
Standard_D3_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	14.0	73,768	2,279	3.09%	56
Standard_D4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	28.0	141,204	19,309	13.67%	70
Standard_D4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	28.0	149,027	1,783	1.20%	98
Standard_D4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	28.0	143,576	1,305	0.91%	49

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D5_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	56.0	281,644	9,594	3.41%	105
Standard_D5_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	56.0	296,879	2,069	0.70%	28
Standard_D5_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	56.0	284,996	5,417	1.90%	77

High performance compute

HBv3

(04/02/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_HB120rs_v3	AMD EPYC 7V13 64-Core Processor	120	4	440.9	2,725,819	62,125	2.28%	7
Standard_HB120rs_v3	AMD EPYC 7V73X 64-Core Processor	120	4	440.9	2,736,592	88,258	3.23%	49
Standard_HB120-16rs_v3	AMD EPYC 7V73X	16	4	440.9	441,414	16,866	3.82%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	64-Core Processor							
Standard_HB120-32rs_v3	AMD EPYC 7V73X 64-Core Processor	32	4	440.9	821,802	31,550	3.84%	35
Standard_HB120-32rs_v3	AMD EPYC 7V13 64-Core Processor	32	4	440.9	812,130	29,738	3.66%	7
Standard_HB120-64rs_v3	AMD EPYC 7V73X 64-Core Processor	64	4	440.9	1,602,181	77,107	4.81%	42
Standard_HB120-96rs_v3	AMD EPYC 7V73X 64-Core Processor	96	4	440.9	2,396,063	79,104	3.30%	35
Standard_HB120-96rs_v3	AMD EPYC 7V13 64-Core Processor	96	4	440.9	2,345,787	74,660	3.18%	7

HBrsv2

(04/02/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_HB120rs_v2	AMD EPYC 7V12 64-Core Processor	120	30	425.1	2,583,980	68,594	2.65%	133

HBS - memory bandwidth (AMD EPYC)

(10/14/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_HB60rs	AMD EPYC 7551 32-Core Processor	60	15	228.0	986,593	28,102	2.85%	21

HCS - dense computation (Intel Xeon Platinum 8168)

(10/14/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_HC44rs	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	44	2	352.0	995,006	25,995	2.61%	21

Memory optimized

EADSv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2ads_v5	AMD EPYC 7763 64-Core Processor	2	1	15.6	38,922	47	0.12%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E4ads_v5	AMD EPYC 7763 64-Core Processor	4	1	31.4	72,638	140	0.19%	35
Standard_E4-2ads_v5	AMD EPYC 7763 64-Core Processor	2	1	31.4	38,924	52	0.13%	35
Standard_E8ads_v5	AMD EPYC 7763 64-Core Processor	8	1	62.8	153,765	572	0.37%	35
Standard_E8-2ads_v5	AMD EPYC 7763 64-Core Processor	2	1	62.8	38,916	48	0.12%	35
Standard_E8-4ads_v5	AMD EPYC 7763 64-Core Processor	4	1	62.8	73,154	920	1.26%	35
Standard_E16ads_v5	AMD EPYC 7763 64-Core Processor	16	1	125.8	303,780	4,336	1.43%	35
Standard_E16-4ads_v5	AMD EPYC 7763 64-Core Processor	4	1	125.8	72,871	365	0.50%	35
Standard_E16-8ads_v5	AMD EPYC 7763 64-Core Processor	8	1	125.8	153,767	603	0.39%	35
Standard_E20ads_v5	AMD EPYC 7763 64-	20	2	157.4	375,535	4,506	1.20%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_E32ads_v5	AMD EPYC 7763 64-Core Processor	32	1	251.7	599,632	7,739	1.29%	35
Standard_E32-8ads_v5	AMD EPYC 7763 64-Core Processor	8	1	251.7	153,859	642	0.42%	35
Standard_E32-16ads_v5	AMD EPYC 7763 64-Core Processor	16	1	251.7	305,170	3,524	1.15%	35
Standard_E48ads_v5	AMD EPYC 7763 64-Core Processor	48	1	377.7	892,509	12,866	1.44%	35
Standard_E64ads_v5	AMD EPYC 7763 64-Core Processor	64	1	503.7	1,195,479	15,280	1.28%	35
Standard_E64-16ads_v5	AMD EPYC 7763 64-Core Processor	16	1	503.7	307,095	958	0.31%	35
Standard_E64-32ads_v5	AMD EPYC 7763 64-Core Processor	32	1	503.7	598,891	7,892	1.32%	35
Standard_E96ads_v5	AMD EPYC 7763 64-Core Processor	96	2	661.4	1,832,942	15,387	0.84%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E96-24ads_v5	AMD EPYC 7763 64-Core Processor	24	1	661.0	450,787	5,791	1.28%	35
Standard_E96-48ads_v5	AMD EPYC 7763 64-Core Processor	48	1	661.0	894,161	10,898	1.22%	35
Standard_E96iads_v5	AMD EPYC 7763 64-Core Processor	96	2	661.4	1,837,643	18,009	0.98%	35
Standard_E112iads_v5	AMD EPYC 7763 64-Core Processor	112	2	661.4	2,125,685	28,945	1.36%	35

EASv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2as_v5	AMD EPYC 7763 64-Core Processor	2	1	15.6	38,919	45	0.11%	35
Standard_E4as_v5	AMD EPYC 7763 64-Core Processor	4	1	31.4	72,704	161	0.22%	35
Standard_E4-2as_v5	AMD EPYC 7763 64-	2	1	31.4	37,346	3,168	8.48%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_E8as_v5	AMD EPYC 7763 64-Core Processor	8	1	62.8	153,881	485	0.31%	35
Standard_E8-2as_v5	AMD EPYC 7763 64-Core Processor	2	1	62.8	38,929	47	0.12%	35
Standard_E8-4as_v5	AMD EPYC 7763 64-Core Processor	4	1	62.8	72,735	153	0.21%	35
Standard_E16as_v5	AMD EPYC 7763 64-Core Processor	16	1	125.8	305,729	2,717	0.89%	35
Standard_E16-4as_v5	AMD EPYC 7763 64-Core Processor	4	1	125.8	72,736	225	0.31%	35
Standard_E16-8as_v5	AMD EPYC 7763 64-Core Processor	8	1	125.8	154,340	805	0.52%	35
Standard_E20as_v5	AMD EPYC 7763 64-Core Processor	20	2	157.4	376,978	4,605	1.22%	35
Standard_E32as_v5	AMD EPYC 7763 64-Core Processor	32	1	251.7	595,261	7,046	1.18%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E32-8as_v5	AMD EPYC 7763 64-Core Processor	8	1	251.7	153,867	462	0.30%	35
Standard_E32-16as_v5	AMD EPYC 7763 64-Core Processor	16	1	251.7	306,591	4,517	1.47%	35
Standard_E48as_v5	AMD EPYC 7763 64-Core Processor	48	1	377.7	892,935	11,974	1.34%	35
Standard_E64as_v5	AMD EPYC 7763 64-Core Processor	64	1	503.7	1,186,352	19,335	1.63%	35
Standard_E64-16as_v5	AMD EPYC 7763 64-Core Processor	16	1	503.7	306,793	869	0.28%	35
Standard_E64-32as_v5	AMD EPYC 7763 64-Core Processor	32	1	503.7	600,716	7,948	1.32%	35
Standard_E96as_v5	AMD EPYC 7763 64-Core Processor	96	2	661.4	1,829,274	23,977	1.31%	35
Standard_E96-24as_v5	AMD EPYC 7763 64-Core Processor	24	1	661.0	448,295	6,035	1.35%	35
Standard_E96-48as_v5	AMD EPYC 7763 64-	48	1	661.0	899,801	11,515	1.28%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_E96ias_v5	AMD EPYC 7763 64-Core Processor	96	2	661.4	1,836,489	27,407	1.49%	35
Standard_E112ias_v5	AMD EPYC 7763 64-Core Processor	112	2	661.4	2,115,432	31,094	1.47%	35

EDSv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	15.6	34,923	12	0.04%	35
Standard_E4ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	31.4	68,727	286	0.42%	63
Standard_E4-2ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	31.4	34,926	10	0.03%	49
Standard_E8ds_v5	Intel(R) Xeon(R) Platinum 8370C	8	1	62.8	136,905	460	0.34%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.80GHz							
Standard_E8-2ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	62.8	34,902	56	0.16%	42
Standard_E8-4ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	62.8	68,738	252	0.37%	42
Standard_E16ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	125.8	271,926	3,097	1.14%	35
Standard_E16-4ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	125.8	68,703	217	0.32%	42
Standard_E16-8ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	125.8	136,880	469	0.34%	49
Standard_E20ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	20	1	157.2	341,049	1,350	0.40%	35
Standard_E32ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	251.7	545,162	2,334	0.43%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E32-8ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	251.7	136,995	454	0.33%	49
Standard_E32-16ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	251.7	273,226	1,050	0.38%	56
Standard_E48ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	377.9	807,259	9,663	1.20%	35
Standard_E64ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	503.9	1,060,197	13,810	1.30%	42
Standard_E64-16ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	2	503.9	273,485	1,014	0.37%	35
Standard_E64-32ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	2	503.9	536,329	9,334	1.74%	42
Standard_E96ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	661.4	1,580,476	17,308	1.10%	35
Standard_E96-24ds_v5	Intel(R) Xeon(R)	24	2	661.4	397,038	5,317	1.34%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8370C CPU @ 2.80GHz							
Standard_E96-48ds_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	661.4	800,107	8,216	1.03%	35
Standard_E104ids_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	104	2	661.4	1,708,585	21,877	1.28%	42

EDv5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	15.6	34,927	9	0.03%	49
Standard_E4d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	31.4	68,699	254	0.37%	42
Standard_E8d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	62.8	136,974	509	0.37%	49

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E16d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	125.8	273,081	1,049	0.38%	49
Standard_E20d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	20	1	157.2	341,711	1,560	0.46%	42
Standard_E32d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	251.7	545,310	2,471	0.45%	56
Standard_E48d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	377.9	801,326	9,677	1.21%	42
Standard_E64d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	503.9	1,062,425	13,190	1.24%	49
Standard_E96d_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	661.4	1,584,556	18,706	1.18%	49
Standard_E104id_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	104	2	661.4	1,709,169	17,132	1.00%	35

ESv5

(03/11/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	15.6	31,454	2,465	7.84%	49
Standard_E4s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	31.4	65,672	4,558	6.94%	49
Standard_E4-2s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	31.4	32,937	2,633	7.99%	49
Standard_E8s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	62.8	120,429	6,932	5.76%	49
Standard_E8-2s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	62.8	30,473	1,714	5.62%	35
Standard_E8-4s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	62.8	64,115	4,800	7.49%	42
Standard_E16s_v5	Intel(R) Xeon(R) Platinum 8370C	16	1	125.8	254,478	19,263	7.57%	56

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.80GHz							
Standard_E16-4s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	125.8	65,592	5,002	7.63%	49
Standard_E16-8s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	125.8	121,123	7,984	6.59%	35
Standard_E20s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	20	1	157.2	323,191	23,552	7.29%	35
Standard_E32s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	251.7	495,411	28,367	5.73%	35
Standard_E32-8s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	251.7	123,436	9,678	7.84%	42
Standard_E32-16s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	251.7	232,861	1,867	0.80%	35
Standard_E48s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	377.9	699,578	15,956	2.28%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E64s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	503.9	969,853	49,428	5.10%	35
Standard_E64-16s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	2	503.9	244,325	13,378	5.48%	35
Standard_E64-32s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	2	503.9	499,778	35,333	7.07%	35
Standard_E96s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	661.4	1,417,598	86,297	6.09%	35
Standard_E96-24s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	24	2	661.4	368,448	27,903	7.57%	35
Standard_E96-48s_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	661.4	711,917	43,548	6.12%	35
Standard_E104is_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	104	2	661.4	1,718,000	19,930	1.16%	35

Ev5

(03/14/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	15.6	31,147	2,390	7.67%	49
Standard_E4_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	31.4	63,068	5,021	7.96%	49
Standard_E8_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	62.8	126,346	9,905	7.84%	49
Standard_E16_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	125.8	252,327	18,906	7.49%	49
Standard_E20_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	20	1	157.2	307,839	23,401	7.60%	49
Standard_E32_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	251.7	528,288	18,156	3.44%	42
Standard_E48_v5	Intel(R) Xeon(R) Platinum 8370C	48	2	377.9	704,530	25,690	3.65%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.80GHz							
Standard_E64_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	64	2	503.9	925,660	15,870	1.71%	35
Standard_E96_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	96	2	661.4	1,369,408	18,928	1.38%	35
Standard_E104i_v5	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	104	2	661.4	1,507,969	12,223	0.81%	35

Easv4

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2as_v4	AMD EPYC 7452 32-Core Processor	2	1	16.0	38,155	432	1.13%	259
Standard_E4as_v4	AMD EPYC 7452 32-Core Processor	4	1	32.0	71,500	1,260	1.76%	259
Standard_E8as_v4	AMD EPYC 7452 32-	8	1	64.0	151,201	3,171	2.10%	259

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_E16as_v4	AMD EPYC 7452 32-Core Processor	16	2	128.0	293,186	6,371	2.17%	280
Standard_E20as_v4	AMD EPYC 7452 32-Core Processor	20	3	160.0	363,292	8,744	2.41%	280
Standard_E32as_v4	AMD EPYC 7452 32-Core Processor	32	4	256.0	570,199	14,140	2.48%	294
Standard_E48as_v4	AMD EPYC 7452 32-Core Processor	48	6	384.0	841,547	21,858	2.60%	301
Standard_E64as_v4	AMD EPYC 7452 32-Core Processor	64	8	512.0	1,108,151	28,783	2.60%	28
Standard_E96as_v4	AMD EPYC 7452 32-Core Processor	96	12	672.0	1,578,349	27,080	1.72%	42

Eav4

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2a_v4	AMD EPYC	2	1	16.0	38,035	900	2.37%	196

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	7452 32-Core Processor							
Standard_E4a_v4	AMD EPYC 7452 32-Core Processor	4	1	32.0	71,345	949	1.33%	196
Standard_E8a_v4	AMD EPYC 7452 32-Core Processor	8	1	64.0	151,511	1,940	1.28%	189
Standard_E16a_v4	AMD EPYC 7452 32-Core Processor	16	2	128.0	292,668	7,220	2.47%	196
Standard_E20a_v4	AMD EPYC 7452 32-Core Processor	20	3	160.0	361,845	9,332	2.58%	196
Standard_E32a_v4	AMD EPYC 7452 32-Core Processor	32	4	256.0	570,615	14,419	2.53%	196
Standard_E48a_v4	AMD EPYC 7452 32-Core Processor	48	6	384.0	841,419	26,181	3.11%	196
Standard_E64a_v4	AMD EPYC 7452 32-Core Processor	64	8	512.0	1,116,891	28,545	2.56%	14
Standard_E96a_v4	AMD EPYC 7452 32-Core Processor	96	12	672.0	1,592,228	32,515	2.04%	14

EDSv4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	35,975	673	1.87%	189
Standard_E4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	65,819	523	0.79%	182
Standard_E4-2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	32.0	35,623	312	0.88%	189
Standard_E8ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	135,576	723	0.53%	189
Standard_E8-2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	64.0	35,831	529	1.48%	189
Standard_E8-4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	64.0	66,007	778	1.18%	189
Standard_E16ds_v4	Intel(R) Xeon(R) Platinum	16	1	128.0	271,138	1,608	0.59%	189

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8272CL CPU @ 2.60GHz							
Standard_E16-4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	128.0	66,461	1,250	1.88%	189
Standard_E16-8ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	128.0	135,960	1,073	0.79%	189
Standard_E20ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	339,000	1,496	0.44%	189
Standard_E32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	538,949	6,669	1.24%	168
Standard_E32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	256.0	527,258	11,587	2.20%	21
Standard_E32-8ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	256.0	136,425	1,166	0.85%	147
Standard_E32-8ds_v4	Intel(R) Xeon(R) Platinum 8272CL	8	2	256.0	136,743	2,124	1.55%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E32-16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	256.0	271,264	1,345	0.50%	154
Standard_E32-16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	2	256.0	269,101	4,332	1.61%	28
Standard_E48ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	775,434	10,356	1.34%	189
Standard_E64ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	504.0	1,027,169	12,890	1.25%	182
Standard_E64-16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	2	504.0	266,956	4,181	1.57%	189
Standard_E64-32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	504.0	521,183	6,655	1.28%	182

EDv4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	35,901	490	1.36%	189
Standard_E4d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	65,859	786	1.19%	189
Standard_E8d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	135,986	1,194	0.88%	189
Standard_E16d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	271,196	1,448	0.53%	182
Standard_E20d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	338,449	2,948	0.87%	189
Standard_E32d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	539,487	5,562	1.03%	161
Standard_E32d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	256.0	529,841	13,344	2.52%	28

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E48d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	772,684	11,745	1.52%	189
Standard_E64d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	504.0	1,025,563	13,902	1.36%	189

Esv4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	35,750	491	1.37%	175
Standard_E4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	65,919	696	1.06%	182
Standard_E4-2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	32.0	35,875	598	1.67%	182
Standard_E8s_v4	Intel(R) Xeon(R) Platinum 8272CL	8	1	64.0	136,061	1,199	0.88%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E8-2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	64.0	35,762	404	1.13%	182
Standard_E8-4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	64.0	65,865	530	0.81%	182
Standard_E16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	270,956	1,486	0.55%	182
Standard_E16-4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	128.0	66,290	1,010	1.52%	182
Standard_E16-8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	128.0	135,990	1,139	0.84%	182
Standard_E20s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	339,149	1,689	0.50%	182
Standard_E32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	540,883	3,428	0.63%	154

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	256.0	533,105	10,700	2.01%	28
Standard_E32-8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	256.0	136,061	1,055	0.78%	140
Standard_E32-8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	2	256.0	136,911	3,347	2.44%	35
Standard_E32-16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	256.0	271,570	1,334	0.49%	147
Standard_E32-16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	2	256.0	264,434	5,182	1.96%	35
Standard_E48s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	774,071	9,926	1.28%	182
Standard_E64s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	504.0	1,024,690	13,752	1.34%	182
Standard_E64-16s_v4	Intel(R) Xeon(R)	16	2	504.0	266,980	3,951	1.48%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8272CL CPU @ 2.60GHz							
Standard_E64-32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	504.0	521,342	7,052	1.35%	182

Ev4

(09/22/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	35,781	422	1.18%	182
Standard_E4_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	65,742	489	0.74%	175
Standard_E8_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	135,668	847	0.62%	182
Standard_E16_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	270,553	2,565	0.95%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E20_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	338,166	4,445	1.31%	175
Standard_E32_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	541,040	2,289	0.42%	154
Standard_E32_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	256.0	520,279	6,500	1.25%	28
Standard_E48_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	774,116	10,210	1.32%	175
Standard_E64_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	504.0	1,025,603	13,715	1.34%	182

Esv3 - Memory Optimized + Premium Storage

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2s_v3	Intel(R) Xeon(R) CPU E5-2673 v4	2	1	16.0	27,008	2,136	7.91%	105

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_E2s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	16.0	26,729	987	3.69%	98
Standard_E2s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	31,341	14	0.04%	21
Standard_E4s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	32.0	51,647	3,354	6.49%	105
Standard_E4s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	32.0	49,459	2,426	4.91%	98
Standard_E4s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	57,986	283	0.49%	35
Standard_E4-2s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	32.0	26,468	1,997	7.54%	105
Standard_E4-2s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	32.0	26,457	817	3.09%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E4-2s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	32.0	31,350	15	0.05%	42
Standard_E8s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	64.0	101,750	7,494	7.36%	126
Standard_E8s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	64.0	101,526	1,976	1.95%	77
Standard_E8s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	120,073	826	0.69%	35
Standard_E8-2s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	64.0	26,318	1,665	6.33%	140
Standard_E8-2s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	64.0	27,219	1,167	4.29%	56
Standard_E8-2s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	64.0	31,322	38	0.12%	42
Standard_E8-4s_v3	Intel(R) Xeon(R)	4	1	64.0	52,719	3,706	7.03%	119

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E5-2673 v4 @ 2.30GHz							
Standard_E8-4s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	64.0	49,518	2,675	5.40%	77
Standard_E8-4s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	64.0	57,859	378	0.65%	42
Standard_E16s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	128.0	195,755	5,337	2.73%	126
Standard_E16s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	128.0	199,681	2,067	1.04%	77
Standard_E16s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	240,250	984	0.41%	28
Standard_E16-4s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	128.0	54,920	3,833	6.98%	140
Standard_E16-4s_v3	Intel(R) Xeon(R) Platinum 8171M	4	1	128.0	50,869	3,400	6.68%	77

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E16-4s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	128.0	58,019	307	0.53%	21
Standard_E16-8s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	128.0	103,358	6,272	6.07%	133
Standard_E16-8s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	128.0	102,945	2,982	2.90%	63
Standard_E16-8s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	128.0	120,189	963	0.80%	35
Standard_E20s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	160.0	245,271	1,479	0.60%	112
Standard_E20s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	160.0	249,245	2,057	0.83%	98
Standard_E20s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	300,197	1,881	0.63%	28

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E32s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	2	256.0	379,015	6,761	1.78%	126
Standard_E32s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	256.0	397,086	2,450	0.62%	84
Standard_E32s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	477,740	4,873	1.02%	28
Standard_E32-8s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	2	256.0	105,799	5,853	5.53%	98
Standard_E32-8s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	256.0	103,535	2,602	2.51%	105
Standard_E32-8s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	256.0	120,109	593	0.49%	28
Standard_E32-16s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	2	256.0	199,765	7,562	3.79%	119
Standard_E32-16s_v3	Intel(R) Xeon(R)	16	1	256.0	202,033	3,104	1.54%	77

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8171M CPU @ 2.60GHz							
Standard_E32-16s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	256.0	240,249	1,126	0.47%	42
Standard_E48s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	48	2	384.0	570,727	4,290	0.75%	98
Standard_E48s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	48	2	384.0	578,285	5,864	1.01%	105
Standard_E48s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	687,857	6,709	0.98%	35
Standard_E64s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	64	2	432.0	763,127	5,541	0.73%	42
Standard_E64s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	64	2	432.0	770,574	7,440	0.97%	161
Standard_E64s_v3	Intel(R) Xeon(R) Platinum 8272CL	64	2	432.0	913,390	9,287	1.02%	21

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E64-16s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	2	432.0	234,395	2,364	1.01%	42
Standard_E64-16s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	2	432.0	211,781	7,841	3.70%	161
Standard_E64-16s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	2	432.0	232,571	3,046	1.31%	28
Standard_E64-32s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	2	432.0	411,194	4,489	1.09%	56
Standard_E64-32s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	2	432.0	395,227	5,245	1.33%	161
Standard_E64-32s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	432.0	464,744	6,855	1.48%	21

Ev3 - Memory Optimized

(09/23/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	16.0	27,615	2,226	8.06%	133
Standard_E2_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	16.0	27,471	1,897	6.90%	84
Standard_E4_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	32.0	51,755	3,604	6.96%	175
Standard_E4_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	32.0	48,159	1,286	2.67%	63
Standard_E8_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	64.0	100,794	5,678	5.63%	154
Standard_E8_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	64.0	100,458	1,381	1.38%	84
Standard_E16_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	128.0	196,170	4,590	2.34%	189

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E16_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	128.0	199,986	1,534	0.77%	42
Standard_E16_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	240,246	680	0.28%	7
Standard_E20_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	160.0	243,816	2,644	1.08%	196
Standard_E20_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	160.0	245,193	6,282	2.56%	35
Standard_E20_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	300,653	666	0.22%	7
Standard_E32_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	2	256.0	377,052	7,434	1.97%	147
Standard_E32_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	256.0	393,569	5,278	1.34%	70
Standard_E32_v3	Intel(R) Xeon(R)	32	1	256.0	479,255	2,237	0.47%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8272CL CPU @ 2.60GHz							
Standard_E48_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	48	2	384.0	569,428	5,166	0.91%	147
Standard_E48_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	48	2	384.0	577,766	6,460	1.12%	77
Standard_E48_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	687,699	7,983	1.16%	14
Standard_E64_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	64	2	432.0	761,587	5,928	0.78%	63
Standard_E64_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	64	2	432.0	765,301	11,204	1.46%	161
Standard_E64_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	432.0	913,332	12,240	1.34%	14

DSv2 - Memory Optimized + Premium Storage

(09/30/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS11_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	14.0	37,489	1,314	3.50%	49
Standard_DS11_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	14.0	40,138	3,286	8.19%	70
Standard_DS11_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	14.0	37,889	1,096	2.89%	84
Standard_DS11_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	14.0	44,107	406	0.92%	28
Standard_DS11-1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	14.0	19,615	846	4.31%	42
Standard_DS11-1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	14.0	20,835	1,937	9.30%	42
Standard_DS11-1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	14.0	18,765	767	4.09%	126

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS11-1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	14.0	23,145	1,425	6.16%	21
Standard_DS12_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	28.0	73,222	1,506	2.06%	42
Standard_DS12_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	28.0	76,452	5,858	7.66%	77
Standard_DS12_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	28.0	73,140	3,323	4.54%	84
Standard_DS12_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	28.0	86,916	1,044	1.20%	28
Standard_DS12-1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	28.0	19,387	899	4.64%	56
Standard_DS12-1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	28.0	22,048	1,729	7.84%	77
Standard_DS12-1_v2	Intel(R) Xeon(R)	1	1	28.0	19,102	943	4.94%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8171M CPU @ 2.60GHz							
Standard_DS12-1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	28.0	22,205	32	0.14%	28
Standard_DS12-2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	28.0	38,766	1,060	2.74%	56
Standard_DS12-2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	28.0	41,447	3,954	9.54%	91
Standard_DS12-2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	28.0	37,639	1,579	4.19%	70
Standard_DS12-2_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	28.0	44,226	810	1.83%	14
Standard_DS13_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	56.0	147,234	1,202	0.82%	56
Standard_DS13_v2	Intel(R) Xeon(R) CPU E5-2673 v4	8	1	56.0	147,218	1,636	1.11%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_DS13_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	56.0	144,612	1,582	1.09%	84
Standard_DS13_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	56.0	172,617	755	0.44%	21
Standard_DS13-2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	56.0	38,688	816	2.11%	56
Standard_DS13-2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	56.0	39,898	2,493	6.25%	84
Standard_DS13-2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	56.0	38,861	2,788	7.17%	91
Standard_DS13-4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	56.0	73,105	1,727	2.36%	70
Standard_DS13-4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	56.0	77,848	4,307	5.53%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS13-4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	56.0	74,935	4,384	5.85%	63
Standard_DS13-4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	56.0	86,402	529	0.61%	14
Standard_DS14_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	112.0	284,796	4,889	1.72%	49
Standard_DS14_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	112.0	298,261	1,351	0.45%	63
Standard_DS14_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	112.0	286,993	3,418	1.19%	98
Standard_DS14_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	112.0	345,521	1,549	0.45%	21
Standard_DS14-4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	2	112.0	71,287	2,692	3.78%	42
Standard_DS14-4_v2	Intel(R) Xeon(R)	4	1	112.0	81,710	6,156	7.53%	56

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E5-2673 v4 @ 2.30GHz							
Standard_DS14-4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	112.0	75,101	2,765	3.68%	112
Standard_DS14-4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	112.0	86,598	517	0.60%	21
Standard_DS14-8_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	2	112.0	143,882	4,104	2.85%	49
Standard_DS14-8_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	112.0	150,389	4,548	3.02%	70
Standard_DS14-8_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	112.0	145,955	2,224	1.52%	105
Standard_DS14-8_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	112.0	171,910	754	0.44%	7
Standard_DS15_v2	Intel(R) Xeon(R) CPU E5-2673 v3	20	2	140.0	355,354	6,693	1.88%	49

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.40GHz							
Standard_DS15_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	140.0	421,202	56,292	13.36%	154

Dv2 - Memory Optimized Compute

(09/30/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D11_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	14.0	37,655	1,925	5.11%	70
Standard_D11_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	14.0	39,953	3,019	7.56%	70
Standard_D11_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	14.0	37,979	1,830	4.82%	56
Standard_D12_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	28.0	71,521	4,218	5.90%	77
Standard_D12_v2	Intel(R) Xeon(R) CPU E5-	4	1	28.0	77,439	5,095	6.58%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	2673 v4 @ 2.30GHz							
Standard_D12_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	28.0	73,464	1,263	1.72%	56
Standard_D12_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	28.0	87,100	434	0.50%	7
Standard_D13_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	56.0	141,552	16,057	11.34%	70
Standard_D13_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	56.0	148,352	2,754	1.86%	70
Standard_D13_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	56.0	141,946	3,604	2.54%	42
Standard_D13_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	56.0	167,641	1,680	1.00%	7
Standard_D14_v2	Intel(R) Xeon(R) CPU E5-2673 v3	16	2	112.0	283,793	5,207	1.83%	77

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.40GHz							
Standard_D14_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	112.0	297,163	1,817	0.61%	63
Standard_D14_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	112.0	286,817	4,109	1.43%	49
Standard_D15_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	2	140.0	354,279	6,021	1.70%	35
Standard_D15_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	140.0	418,475	58,044	13.87%	154

Msv2 High Memory

(03/17/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_M208s_v2	Intel(R) Xeon(R) Platinum 8180M CPU @ 2.50GHz	208	4	2,850.0	3,020,762	55,134	1.83%	84
Standard_M208ms_v2	Intel(R) Xeon(R) Platinum	208	4	5,700.0	3,009,120	58,843	1.96%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8180M CPU @ 2.50GHz							
Standard_M208ms_v2	Intel(R) Xeon(R) Platinum 8180M CPU @ 2.50GHz	208	8	5,700.0	3,093,184	33,253	1.08%	42
Standard_M416s_v2	Intel(R) Xeon(R) Platinum 8180M CPU @ 2.50GHz	416	8	5,700.0	5,959,252	93,933	1.58%	84
Standard_M416ms_v2	Intel(R) Xeon(R) Platinum 8180M CPU @ 2.50GHz	416	8	11,400.0	5,910,261	101,190	1.71%	84
Standard_M416-208s_v2	Intel(R) Xeon(R) Platinum 8180M CPU @ 2.50GHz	208	8	5,700.0	3,085,232	36,568	1.19%	70
Standard_M416-208ms_v2	Intel(R) Xeon(R) Platinum 8180M CPU @ 2.50GHz	208	8	11,400.0	3,064,892	40,531	1.32%	77

Msv2 Medium Memory

(03/16/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_M32ms_v2	Intel(R) Xeon(R)	32	2	861.2	501,859	6,988	1.39%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8280M CPU @ 2.70GHz							
Standard_M32dms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	32	2	861.2	507,318	12,103	2.39%	35
Standard_M64s_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	1,007.9	977,226	15,256	1.56%	35
Standard_M64ds_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	1,007.9	980,928	16,781	1.71%	35
Standard_M64ms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	4	1,763.9	980,511	16,384	1.67%	42
Standard_M64dms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	4	1,763.9	979,367	13,927	1.42%	35
Standard_M128s_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	2,015.9	1,905,457	53,830	2.83%	35
Standard_M128ds_v2	Intel(R) Xeon(R) Platinum 8280M	128	4	2,015.9	1,925,932	29,177	1.51%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.70GHz							
Standard_M128ms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	3,831.1	1,907,485	30,641	1.61%	42
Standard_M128dms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	3,831.1	1,907,006	25,958	1.36%	35
Standard_M192ms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	192	4	4,031.9	2,794,826	44,549	1.59%	84
Standard_M192is_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	192	4	2,015.9	2,805,023	39,961	1.42%	35
Standard_M192ims_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	192	4	4,031.9	2,797,557	39,161	1.40%	35
Standard_M192ids_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	192	4	2,015.9	2,818,523	48,260	1.71%	35
Standard_M192idms_v2	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	192	4	4,031.9	2,813,406	47,652	1.69%	35

M-series Medium Memory

(09/29/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_M64	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	64	2	1,024.0	821,678	11,118	1.35%	77
Standard_M64	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	1,024.0	754,180	5,686	0.75%	7
Standard_M64m	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	64	2	1,792.0	817,400	9,397	1.15%	56
Standard_M64m	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	1,792.0	754,929	10,566	1.40%	28
Standard_M128	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	128	4	2,048.0	1,643,769	17,798	1.08%	70
Standard_M128	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	2,048.0	1,475,843	16,197	1.10%	14
Standard_M128m	Intel(R) Xeon(R) CPU E7-	128	4	3,892.0	1,631,313	18,162	1.11%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8890 v3 @ 2.50GHz							
Standard_M128m	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	3,892.0	1,471,781	17,646	1.20%	14
Standard_M8ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	8	1	218.8	110,370	366	0.33%	84
Standard_M8-2ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	2	1	218.8	28,041	35	0.12%	84
Standard_M8-4ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	4	1	218.8	56,231	348	0.62%	77
Standard_M8-4ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	4	1	218.8	48,391	280	0.58%	7
Standard_M16ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	16	1	437.5	220,331	473	0.21%	70
Standard_M16ms	Intel(R) Xeon(R) Platinum 8280M	16	1	437.5	200,561	877	0.44%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.70GHz							
Standard_M16-4ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	4	1	437.5	56,218	304	0.54%	70
Standard_M16-4ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	4	1	437.5	48,401	205	0.42%	14
Standard_M16-8ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	8	1	437.5	110,426	383	0.35%	56
Standard_M16-8ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	8	1	437.5	100,377	442	0.44%	28
Standard_M32ls	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	32	1	256.0	433,173	4,290	0.99%	70
Standard_M32ls	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	32	1	256.0	400,287	1,610	0.40%	14
Standard_M32ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	32	1	875.0	434,866	2,560	0.59%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_M32ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	32	1	875.0	400,583	1,596	0.40%	14
Standard_M32-8ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	8	1	875.0	110,408	346	0.31%	56
Standard_M32-8ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	8	1	875.0	100,521	643	0.64%	28
Standard_M32-16ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	16	1	875.0	220,495	569	0.26%	70
Standard_M32-16ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	16	1	875.0	200,494	723	0.36%	14
Standard_M32ts	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	32	1	192.0	435,785	2,164	0.50%	77
Standard_M32ts	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	32	1	192.0	399,771	1,291	0.32%	7
Standard_M64s	Intel(R) Xeon(R)	64	2	1,024.0	822,245	7,256	0.88%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E7-8890 v3 @ 2.50GHz							
Standard_M64s	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	1,024.0	750,702	9,866	1.31%	14
Standard_M64ls	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	64	2	512.0	820,406	13,530	1.65%	77
Standard_M64ls	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	512.0	759,753	7,639	1.01%	7
Standard_M64ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	64	2	1,792.0	823,449	7,565	0.92%	70
Standard_M64ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	2	1,792.0	755,541	9,337	1.24%	14
Standard_M64-16ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	16	2	1,792.0	210,947	2,766	1.31%	77
Standard_M64-16ms	Intel(R) Xeon(R) Platinum 8280M	16	2	1,792.0	191,761	3,503	1.83%	7

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.70GHz							
Standard_M64-32ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	32	2	1,792.0	416,909	3,769	0.90%	63
Standard_M64-32ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	32	3	1,792.0	417,771	4,820	1.15%	7
Standard_M64-32ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	32	2	1,792.0	382,773	3,547	0.93%	14
Standard_M128s	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	128	4	2,048.0	1,641,176	16,698	1.02%	70
Standard_M128s	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	2,048.0	1,477,253	15,626	1.06%	14
Standard_M128ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	128	4	3,892.0	1,645,242	17,679	1.07%	63
Standard_M128ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	128	4	3,892.0	1,479,298	16,978	1.15%	21

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_M128-32ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	32	4	3,892.0	414,541	4,739	1.14%	63
Standard_M128-32ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	32	4	3,892.0	380,611	5,653	1.49%	21
Standard_M128-64ms	Intel(R) Xeon(R) CPU E7-8890 v3 @ 2.50GHz	64	4	3,892.0	817,496	8,437	1.03%	56
Standard_M128-64ms	Intel(R) Xeon(R) Platinum 8280M CPU @ 2.70GHz	64	4	3,892.0	749,800	8,506	1.13%	28

Storage optimized

Lsv2 - Storage Optimized

(10/13/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_L8s_v2	AMD EPYC 7551 32-Core Processor	8	1	64.0	102,237	1,320	1.29%	77
Standard_L16s_v2	AMD EPYC 7551 32-	16	2	128.0	198,472	3,734	1.88%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_L32s_v2	AMD EPYC 7551 32-Core Processor	32	4	256.0	390,015	10,126	2.60%	77
Standard_L48s_v2	AMD EPYC 7551 32-Core Processor	48	6	384.0	583,388	15,479	2.65%	77
Standard_L64s_v2	AMD EPYC 7551 32-Core Processor	64	8	512.0	774,827	19,205	2.48%	77
Standard_L80s_v2	AMD EPYC 7551 32-Core Processor	80	10	640.0	966,682	22,811	2.36%	77

Confidential Compute

DCSv3

(04/05/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DC1s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	1	1	7.8	29,127	72	0.25%	56
Standard_DC2s_v3	Intel(R) Xeon(R) Platinum	2	1	15.6	55,785	534	0.96%	56

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8370C CPU @ 2.80GHz							
Standard_DC4s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	31.4	111,636	827	0.74%	63
Standard_DC8s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	62.8	220,443	2,083	0.94%	56
Standard_DC16s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	125.8	439,985	4,224	0.96%	56
Standard_DC24s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	24	1	188.7	659,602	5,809	0.88%	42
Standard_DC32s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	251.7	850,747	8,658	1.02%	63
Standard_DC48s_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	377.9	1,221,232	25,082	2.05%	63

DCDSv3

(04/05/2022 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DC1ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	1	1	7.8	29,130	66	0.23%	42
Standard_DC2ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	2	1	15.6	55,911	207	0.37%	56
Standard_DC4ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	4	1	31.4	111,549	745	0.67%	56
Standard_DC8ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	8	1	62.8	220,223	2,067	0.94%	42
Standard_DC16ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	16	1	125.8	440,580	3,915	0.89%	63
Standard_DC24ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	24	1	188.7	656,455	9,830	1.50%	56
Standard_DC32ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	32	1	251.7	848,418	18,613	2.19%	35

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DC48ds_v3	Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz	48	2	377.9	1,228,361	21,388	1.74%	35

DCsv2

(10/08/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DC1s_v2	Intel(R) Xeon(R) E-2288G CPU @ 3.70GHz	1	1	4.0	34,418	162	0.47%	77
Standard_DC2s_v2	Intel(R) Xeon(R) E-2288G CPU @ 3.70GHz	2	1	8.0	68,562	758	1.11%	77
Standard_DC4s_v2	Intel(R) Xeon(R) E-2288G CPU @ 3.70GHz	4	1	16.0	133,836	1,964	1.47%	77

DCv2

(10/13/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DC8_v2	Intel(R) Xeon(R) E-2288G CPU @ 3.70GHz	8	1	32.0	252,047	3,051	1.21%	77

DCS

(10/01/2020 PBIID:7668456)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DC2s	Intel(R) Xeon(R) E-2176G CPU @ 3.70GHz	2	1	8.0	63,426	391	0.62%	28
Standard_DC4s	Intel(R) Xeon(R) E-2176G CPU @ 3.70GHz	4	1	16.0	124,015	410	0.33%	21

About CoreMark

CoreMark is a benchmark that tests the functionality of a microcontroller (MCU) or central processing unit (CPU). CoreMark is not system dependent, so it functions the same regardless of the platform (e.g. big or little endian, high-end or low-end processor).

Linux numbers were computed by running CoreMark on Ubuntu 18.04. CoreMark was configured with the number of threads set to the number of virtual CPUs, and concurrency set to `PThreads`. The target number of iterations was adjusted based on expected performance to provide a runtime of at least 20 seconds (typically much longer). The final score represents the number of iterations completed divided by the number of seconds it took to run the test. Each test was run at least seven times on each VM. Test run dates shown above. Tests run on multiple VMs across Azure public regions the VM was supported in on the date run.

Running Coremark on Azure VMs

Download:

CoreMark is an open source tool that can be downloaded from GitHub.

Building and Running:

To build and run the benchmark, type:

```
> make
```

Full results are available in the files `run1.log` and `run2.log`. `run1.log` contains CoreMark results. These are the benchmark results with performance parameters. `run2.log` contains benchmark results with validation parameters.

Run Time:

By default, the benchmark will run between 10-100 seconds. To override, use `ITERATIONS=N`

```
% make ITERATIONS=10
```

above flag will run the benchmark for 10 iterations. **Results are only valid for reporting if the benchmark ran for at least 10 seconds!**

Parallel Execution:

Use `XCFLAGS=-DMULTITHREAD=N` where N is number of threads to run in parallel. Several implementations are available to execute in multiple contexts.

```
% make XCFLAGS="-DMULTITHREAD=4 -DUSE_PTHREAD"
```

The above will compile the benchmark for execution on 4 cores.

Recommendations for best results

- The benchmark needs to run for at least 10 seconds, probably longer on larger systems.
- All source files must be compiled with same flags.
- Do not change source files other than `core_portme*` (use `make check` to validate)
- Multiple runs are suggested for best results.

Coverage

Older end-of-life series may not be shown. N series not shown as they are GPU centric and Coremark doesn't measure GPU performance. Newer series may not have been benchmarked yet. Previous versions of this document cited benchmark runs from Ubuntu 16.04 which resulted in slightly lower performance than the current benchmarks running on Ubuntu 18.04.

GPU Series

Performance of GPU based VM series is best understood by using GPU appropriate benchmarks and running at the scale required for your workloads. Azure ranks among the best there:

- Top 10 Supercomputer: November 2021 | TOP500 (Azure powered #10: Voyager-EUS2)
- Machine Learning: MLCommons Training: v1.1 Results | MLCommons (2 highest at scale and largest in the cloud)

Next steps

- For storage capacities, disk details, and additional considerations for choosing among VM sizes, see Sizes for virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Compute benchmark scores for Windows VMs

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

> **Note**
>
> This page is no longer being maintained and is planned for archival on 12/16/2024.
>
> Azure is no longer publishing CoreMark since the metric has limited ability to inform users of the expected performance of a virtual machine across various attributes. For the most accurate results on a specific virtual machine, Azure recommends users run their workload(s) on that virtual machine to verify performance.

The following CoreMark benchmark scores show compute performance for select Azure VMs running Windows Server 2019. Compute benchmark scores are also available for Linux VMs.

Expand table

Type	Families
Compute optimized	Fsv2, Fs, F, GS, G
General purpose	B, Dasv4, Dav4, Ddsv4, Ddv4, Dsv4, Dv4, Dsv3, Dv3, DSv2, Dv2, Av2
High performance compute	HB, HC
Memory optimized	Easv4, Eav4, Edsv4, Edv4, Esv4, Ev4, Esv3, Ev3, DSv2, Dv2
Storage optimized	Lsv2, Ls

Compute optimized

Fsv2 - Compute + Premium Storage

(03/29/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F2s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70 GHz	2	1	4.0	34,903	1,101	3.15%	112
Standard_F2s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	2	1	4.0	34,738	1,331	3.83%	224
Standard_F4s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70 GHz	4	1	8.0	66,828	1,524	2.28%	168
Standard_F4s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	4	1	8.0	66,903	1,047	1.57%	182
Standard_F8s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70 GHz	8	1	16.0	131,477	2,180	1.66%	140
Standard_F8s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	8	1	16.0	132,533	1,732	1.31%	210
Standard_F16s_v2	Intel(R) Xeon(R)	16	1	32.0	260,760	3,629	1.39%	112

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8168 CPU @ 2.70 GHz							
Standard_F16s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	16	1	32.0	265,158	2,185	0.82%	182
Standard_F32s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70 GHz	32	1	64.0	525,608	6,270	1.19%	98
Standard_F32s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	32	1	64.0	530,137	6,085	1.15%	140
Standard_F48s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70 GHz	48	2	96.0	769,768	7,567	0.98%	112
Standard_F48s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	48	1	96.0	742,828	17,316	2.33%	112
Standard_F64s_v2	Intel(R) Xeon(R) Platinum 8168	64	2	128.0	1,030,552	8,106	0.79%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.70 GHz							
Standard_F64s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	64	2	128.0	1,028,052	9,373	0.91%	168
Standard_F72s_v2	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70 GHz	72	2	144.0	N/A	-	-	-
Standard_F72s_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	72	2	144.0	N/A	-	-	-

Fs - Compute Optimized + Premium Storage

(04/28/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F1s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	1	1	2.0	16,445	825	5.02%	42
Standard_F1s	Intel(R) Xeon(R) CPU E5-2673 v4	1	1	2.0	17,614	2,873	16.31%	210

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30 GHz							
Standard_F1s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	1	1	2.0	16,053	1,802	11.22%	70
Standard_F1s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	1	1	2.0	20,007	1,684	8.42%	28
Standard_F2s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	2	1	4.0	33,451	3,424	10.24%	70
Standard_F2s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	2	1	4.0	33,626	2,990	8.89%	154
Standard_F2s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	2	1	4.0	34,386	3,851	11.20%	98
Standard_F2s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	2	1	4.0	36,826	344	0.94%	28

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F4s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	4	1	8.0	67,351	4,407	6.54%	42
Standard_F4s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	4	1	8.0	67,009	4,637	6.92%	196
Standard_F4s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	4	1	8.0	63,668	3,375	5.30%	84
Standard_F4s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	4	1	8.0	79,153	15,034	18.99%	28
Standard_F8s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	8	1	16.0	128,232	1,272	0.99%	42
Standard_F8s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	8	1	16.0	127,871	5,109	4.00%	154
Standard_F8s	Intel(R) Xeon(R) Platinum 8171M CPU @	8	1	16.0	122,811	5,481	4.46%	126

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	2.60 GHz							
Standard_F8s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	8	1	16.0	154,842	10,354	6.69%	28
Standard_F16s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	16	2	32.0	260,883	15,853	6.08%	42
Standard_F16s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	16	1	32.0	255,762	4,966	1.94%	182
Standard_F16s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	16	1	32.0	248,884	11,035	4.43%	70
Standard_F16s	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	16	1	32.0	310,303	21,942	7.07%	28

F - Compute Optimized

(04/28/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F1	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	1	1	2.0	17,356	1,151	6.63%	112
Standard_F1	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	1	1	2.0	16,508	1,740	10.54%	154
Standard_F1	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	1	1	2.0	16,076	2,065	12.84%	70
Standard_F1	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	1	1	2.0	20,074	1,612	8.03%	14
Standard_F2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	2	1	4.0	32,770	1,915	5.84%	126
Standard_F2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	2	1	4.0	33,081	2,242	6.78%	126
Standard_F2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	2	1	4.0	33,310	2,532	7.60%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	2	1	4.0	40,746	2,027	4.98%	14
Standard_F4	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	4	1	8.0	65,694	3,512	5.35%	126
Standard_F4	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	4	1	8.0	65,054	3,457	5.31%	154
Standard_F4	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	4	1	8.0	61,607	3,662	5.94%	56
Standard_F4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	4	1	8.0	76,884	1,763	2.29%	14
Standard_F8	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	8	1	16.0	130,415	5,353	4.10%	98
Standard_F8	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	8	1	16.0	126,139	2,917	2.31%	126

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_F8	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	8	1	16.0	122,443	4,391	3.59%	98
Standard_F8	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	8	1	16.0	144,696	2,172	1.50%	14
Standard_F16	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	16	2	32.0	253,473	8,597	3.39%	140
Standard_F16	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	16	1	32.0	257,457	7,596	2.95%	126
Standard_F16	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	16	1	32.0	244,559	8,036	3.29%	70
Standard_F16	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	16	1	32.0	283,565	8,683	3.06%	14

GS - Compute Optimized + Premium Storage

(05/27/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_GS1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	2	1	28.0	35,593	2,888	8.11%	252
Standard_GS2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	4	1	56.0	72,188	5,949	8.24%	252
Standard_GS3	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	8	1	112.0	132,665	6,910	5.21%	238
Standard_GS4	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	16	1	224.0	261,542	3,722	1.42%	252
Standard_GS4-4	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	4	1	224.0	79,352	4,935	6.22%	224
Standard_GS4-8	Intel(R) Xeon(R) CPU E5-2698B	8	1	224.0	137,774	6,887	5.00%	238

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	v3 @ 2.00 GHz							
Standard_GS5	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	32	2	448.0	507,026	6,895	1.36%	252
Standard_GS5-8	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	8	2	448.0	157,541	3,151	2.00%	238
Standard_GS5-16	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	16	2	448.0	278,656	5,235	1.88%	224

G - Compute Optimized

(05/27/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_G1	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	2	1	28.0	36,386	4,100	11.27%	252

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_G2	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	4	1	56.0	72,484	5,563	7.67%	252
Standard_G3	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	8	1	112.0	136,618	5,714	4.18%	252
Standard_G4	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	16	1	224.0	261,708	3,426	1.31%	238
Standard_G5	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00 GHz	32	2	448.0	507,423	7,261	1.43%	252

General purpose

B - Burstable

(04/12/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_B1ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	1	1	2.0	18,093	679	3.75%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_B1ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	1	1	2.0	18,197	1,341	7.37%	168
Standard_B1ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	1	1	2.0	17,975	920	5.12%	112
Standard_B1ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	1	1	2.0	20,176	1,568	7.77%	28
Standard_B2s	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	2	1	4.0	35,546	660	1.86%	42
Standard_B2s	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	2	1	4.0	36,569	2,172	5.94%	154
Standard_B2s	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	2	1	4.0	36,136	924	2.56%	140
Standard_B2s	Intel(R) Xeon(R) Platinum 8272CL CPU @	2	1	4.0	42,546	834	1.96%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	2.60 GHz							
Standard_B2hms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	2	1	8.0	36,949	1,494	4.04%	28
Standard_B2hms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	2	1	8.0	36,512	2,537	6.95%	70
Standard_B2hms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	2	1	8.0	36,389	990	2.72%	56
Standard_B2ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	2	1	8.0	35,758	1,028	2.88%	42
Standard_B2ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	2	1	8.0	36,028	1,605	4.45%	182
Standard_B2ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	2	1	8.0	36,122	2,128	5.89%	112
Standard_B2ms	Intel(R) Xeon(R) Platinum	2	1	8.0	42,525	672	1.58%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8272CL CPU @ 2.60 GHz							
Standard_B4hms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	4	1	16.0	71,028	879	1.24%	28
Standard_B4hms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	4	1	16.0	73,126	2,954	4.04%	56
Standard_B4hms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	4	1	16.0	68,451	1,571	2.29%	56
Standard_B4hms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	4	1	16.0	83,525	563	0.67%	14
Standard_B4ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	4	1	16.0	70,831	1,135	1.60%	28
Standard_B4ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	4	1	16.0	70,987	2,287	3.22%	168

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_B4ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	4	1	16.0	68,796	1,897	2.76%	84
Standard_B4ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	4	1	16.0	81,712	4,042	4.95%	70
Standard_B8ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	8	1	32.0	141,620	2,256	1.59%	42
Standard_B8ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	8	1	32.0	139,090	3,229	2.32%	182
Standard_B8ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	8	1	32.0	135,510	2,653	1.96%	112
Standard_B8ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	8	1	32.0	164,510	2,254	1.37%	14
Standard_B12ms	Intel(R) Xeon(R) CPU E5-2673 v3	12	1	48.0	206,957	5,240	2.53%	56

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.40 GHz							
Standard_B12ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	12	1	48.0	211,461	4,115	1.95%	154
Standard_B12ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	12	1	48.0	200,729	3,475	1.73%	140
Standard_B16ms	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40 GHz	16	2	64.0	273,257	3,862	1.41%	42
Standard_B16ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	16	1	64.0	282,187	5,030	1.78%	154
Standard_B16ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	16	1	64.0	265,834	5,545	2.09%	112
Standard_B16ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	16	1	64.0	331,694	3,537	1.07%	28
Standard_B20ms	Intel(R) Xeon(R)	20	2	80.0	334,369	8,555	2.56%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU E5-2673 v3 @ 2.40 GHz							
Standard_B20ms	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30 GHz	20	1	80.0	345,686	6,702	1.94%	154
Standard_B20ms	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60 GHz	20	1	80.0	328,900	7,625	2.32%	126
Standard_B20ms	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60 GHz	20	1	80.0	409,515	4,792	1.17%	14

Dasv4

(03/25/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2as_v4	AMD EPYC 7452 32-Core Processor	2	1	8.0	37,986	1,199	3.16%	238
Standard_D4as_v4	AMD EPYC 7452 32-Core Processor	4	1	16.0	75,828	1,343	1.77%	196

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D8as_v4	AMD EPYC 7452 32-Core Processor	8	1	32.0	150,134	2,511	1.67%	210
Standard_D16as_v4	AMD EPYC 7452 32-Core Processor	16	2	64.0	286,789	5,984	2.09%	224
Standard_D32as_v4	AMD EPYC 7452 32-Core Processor	32	4	128.0	566,270	8,484	1.50%	140
Standard_D48as_v4	AMD EPYC 7452 32-Core Processor	48	6	192.0	829,547	15,679	1.89%	126
Standard_D64as_v4	AMD EPYC 7452 32-Core Processor	64	8	256.0	1,088,030	16,708	1.54%	28
Standard_D96as_v4	AMD EPYC 7452 32-Core Processor	96	12	384.0	N/A	-	-	-

Dav4

(03/25/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2a_v4	AMD EPYC 7452 32-	2	1	8.0	38,028	995	2.62%	238

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_D4a_v4	AMD EPYC 7452 32-Core Processor	4	1	16.0	75,058	1,874	2.50%	238
Standard_D8a_v4	AMD EPYC 7452 32-Core Processor	8	1	32.0	149,706	2,520	1.68%	168
Standard_D16a_v4	AMD EPYC 7452 32-Core Processor	16	2	64.0	287,479	4,907	1.71%	238
Standard_D32a_v4	AMD EPYC 7452 32-Core Processor	32	4	128.0	567,019	11,019	1.94%	210
Standard_D48a_v4	AMD EPYC 7452 32-Core Processor	48	6	192.0	835,617	13,097	1.57%	140
Standard_D64a_v4	AMD EPYC 7452 32-Core Processor	64	8	256.0	1,099,165	21,962	2.00%	252
Standard_D96a_v4	AMD EPYC 7452 32-Core Processor	96	12	384.0	N/A	-	-	-

DDSv4

(03/26/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	34,621	1,588	4.59%	336
Standard_D4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	66,583	2,327	3.49%	336
Standard_D8ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	131,888	3,913	2.97%	336
Standard_D16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	262,436	9,177	3.50%	336
Standard_D32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	531,747	5,956	1.12%	322
Standard_D48ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	750,843	15,060	2.01%	420
Standard_D48ds_v4	Intel(R) Xeon(R) Platinum 8272CL	48	2	192.0	753,948	31,559	4.19%	252

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							

DDv4

(03/26/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	34,704	1,455	4.19%	336
Standard_D4d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	66,629	2,005	3.01%	336
Standard_D8d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	131,953	3,911	2.96%	336
Standard_D16d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	263,568	7,317	2.78%	336
Standard_D32d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	527,571	11,076	2.10%	336

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D48d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	742,908	19,323	2.60%	378
Standard_D48d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	759,921	18,783	2.47%	280

Dsv4

(03/24/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	31,643	3,054	9.65%	406
Standard_D4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	60,878	4,594	7.55%	392
Standard_D8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	119,076	7,683	6.45%	406
Standard_D16s_v4	Intel(R) Xeon(R) Platinum 8272CL	16	1	64.0	242,206	16,772	6.92%	406

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_D32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	483,021	28,105	5.82%	392
Standard_D48s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	694,366	33,144	4.77%	280
Standard_D48s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	705,192	24,651	3.50%	126
Standard_D64s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	1,023,014	17,746	1.73%	364

Dv4

(03/25/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	31,469	2,948	9.37%	406

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D4_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	61,806	4,467	7.23%	406
Standard_D8_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	120,421	8,407	6.98%	392
Standard_D16_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	245,522	17,151	6.99%	812
Standard_D32_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	128.0	487,165	28,119	5.77%	378
Standard_D48_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	1	192.0	688,018	24,945	3.63%	252
Standard_D48_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	192.0	696,691	30,283	4.35%	112
Standard_D64_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	256.0	1,018,300	23,085	2.27%	392

DSv3 - General Compute + Premium Storage

(04/05/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	8.0	23,534	724	3.08%	42
Standard_D2s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	8.0	24,742	2,045	8.27%	112
Standard_D2s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	8.0	24,822	3,702	14.91%	126
Standard_D2s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	8.0	30,392	1,514	4.98%	28
Standard_D4s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	16.0	44,404	537	1.21%	28
Standard_D4s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	16.0	45,725	4,388	9.60%	154

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D4s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	16.0	46,590	3,963	8.51%	112
Standard_D4s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	16.0	50,797	306	0.60%	28
Standard_D8s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	32.0	89,102	849	0.95%	56
Standard_D8s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	32.0	89,422	6,441	7.20%	154
Standard_D8s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	32.0	85,673	2,704	3.16%	112
Standard_D8s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	32.0	101,753	1,013	1.00%	14
Standard_D16s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	1	64.0	179,390	1,403	0.78%	42

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D16s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	64.0	173,313	14,382	8.30%	98
Standard_D16s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	64.0	171,750	1,261	0.73%	70
Standard_D16s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	64.0	204,568	2,434	1.19%	14
Standard_D32s_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	32	2	128.0	358,426	6,880	1.92%	56
Standard_D32s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	1	128.0	364,032	20,351	5.59%	84
Standard_D32s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	128.0	346,172	2,859	0.83%	84

Dv3 - General Compute

(04/05/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	8.0	23,795	1,893	7.96%	70
Standard_D2_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	8.0	24,582	2,036	8.28%	154
Standard_D2_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	8.0	24,376	1,915	7.86%	84
Standard_D4_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	16.0	45,883	3,929	8.56%	70
Standard_D4_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	16.0	46,836	5,296	11.31%	140
Standard_D4_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	16.0	46,281	4,133	8.93%	112
Standard_D8_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	32.0	88,815	1,091	1.23%	126

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D8_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	32.0	89,625	6,366	7.10%	112
Standard_D8_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	32.0	87,549	3,215	3.67%	98
Standard_D32_v3	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	32	2	128.0	353,069	3,792	1.07%	70
Standard_D32_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	32	1	128.0	358,984	19,517	5.44%	126
Standard_D32_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	128.0	356,479	16,176	4.54%	126

DSv2 - General Purpose + Premium Storage

(05/24/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS1_v2	Intel(R) Xeon(R) CPU E5-2673 v3	1	1	3.5	17,338	2,482	14.32%	98

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.40GHz							
Standard_DS1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	3.5	18,579	2,267	12.20%	112
Standard_DS1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	3.5	18,617	2,963	15.92%	168
Standard_DS1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	3.5	20,095	2,368	11.79%	154
Standard_DS2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	7.0	33,367	1,923	5.76%	98
Standard_DS2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	7.0	34,329	4,015	11.70%	182
Standard_DS2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	7.0	34,084	3,226	9.46%	154
Standard_DS2_v2	Intel(R) Xeon(R) Platinum 8272CL	2	1	7.0	38,782	4,385	11.31%	112

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_DS3_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	14.0	67,964	4,394	6.47%	112
Standard_DS3_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	14.0	67,396	5,434	8.06%	168
Standard_DS3_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	14.0	63,593	4,477	7.04%	112
Standard_DS3_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	14.0	78,599	8,145	10.36%	140
Standard_DS4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	28.0	129,452	4,586	3.54%	98
Standard_DS4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	28.0	128,545	6,193	4.82%	140
Standard_DS4_v2	Intel(R) Xeon(R) Platinum 8171M	8	1	28.0	125,892	6,614	5.25%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_DS4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	28.0	150,995	10,036	6.65%	98
Standard_DS5_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	56.0	250,112	1,907	0.76%	84
Standard_DS5_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	56.0	263,493	11,774	4.47%	182
Standard_DS5_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	56.0	248,170	10,260	4.13%	126
Standard_DS5_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	56.0	304,566	25,421	8.35%	98

Dv2 - General Compute

(05/24/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	3.5	17,093	1,249	7.30%	224
Standard_D1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	3.5	17,229	1,954	11.34%	182
Standard_D1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	3.5	16,474	1,558	9.46%	112
Standard_D1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	3.5	21,631	306	1.41%	14
Standard_D2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	7.0	33,613	1,961	5.83%	196
Standard_D2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	7.0	34,301	2,859	8.34%	168
Standard_D2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	7.0	34,041	2,386	7.01%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D2_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	7.0	37,309	911	2.44%	14
Standard_D3_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	14.0	65,186	3,256	4.99%	196
Standard_D3_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	14.0	66,161	3,850	5.82%	168
Standard_D3_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	14.0	65,731	4,742	7.21%	126
Standard_D3_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	14.0	78,383	5,224	6.67%	28
Standard_D4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	28.0	129,637	5,291	4.08%	238
Standard_D4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	28.0	126,313	2,045	1.62%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	28.0	120,903	1,877	1.55%	84
Standard_D4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	28.0	145,529	1,683	1.16%	14
Standard_D5_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	56.0	253,120	9,301	3.67%	238
Standard_D5_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	56.0	258,915	10,136	3.91%	126
Standard_D5_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	56.0	242,876	4,157	1.71%	112
Standard_D5_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	56.0	284,831	20,223	7.10%	28

Av2 - General Compute

(04/12/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_A1_v2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	1	1	2.0	6,854	551	8.04%	28
Standard_A1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	2.0	6,798	724	10.65%	140
Standard_A1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	2.0	6,476	1,151	17.77%	84
Standard_A1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	2.0	6,195	397	6.41%	56
Standard_A1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	2.0	5,924	674	11.38%	28
Standard_A2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	4.0	13,666	1,079	7.89%	168
Standard_A2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	4.0	14,276	756	5.29%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_A2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	4.0	14,223	1,010	7.10%	70
Standard_A2_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	4.0	14,805	166	1.12%	14
Standard_A2m_v2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	2	1	16.0	15,159	72	0.47%	14
Standard_A2m_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	16.0	14,315	1,255	8.76%	126
Standard_A2m_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	16.0	14,149	1,121	7.92%	98
Standard_A2m_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	16.0	13,791	1,201	8.71%	98
Standard_A2m_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	11,336	898	7.93%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_A4_v2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	4	1	8.0	28,551	1,839	6.44%	42
Standard_A4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	8.0	27,710	2,001	7.22%	98
Standard_A4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	8.0	28,371	2,769	9.76%	98
Standard_A4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	8.0	26,780	2,141	7.99%	84
Standard_A4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	8.0	26,476	1,608	6.07%	14
Standard_A4m_v2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	4	1	32.0	28,710	1,032	3.59%	14
Standard_A4m_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	32.0	28,128	2,283	8.12%	112

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_A4m_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	32.0	28,220	2,997	10.62%	126
Standard_A4m_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	32.0	27,720	1,839	6.63%	98
Standard_A8_v2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	8	1	16.0	52,295	1,356	2.59%	28
Standard_A8_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	16.0	56,004	4,045	7.22%	140
Standard_A8_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	16.0	53,367	3,252	6.09%	84
Standard_A8_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	16.0	51,525	3,170	6.15%	84
Standard_A8_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	16.0	49,802	1,009	2.03%	14

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_A8m_v2	Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz	8	2	64.0	51,496	1,643	3.19%	14
Standard_A8m_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	64.0	54,477	3,024	5.55%	126
Standard_A8m_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	64.0	55,195	4,426	8.02%	126
Standard_A8m_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	64.0	52,768	3,528	6.69%	70
Standard_A8m_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	49,188	1,869	3.80%	14

High performance compute

HBS - memory bandwidth (AMD EPYC)

(04/29/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_HB60rs	AMD EPYC 7551 32-Core Processor	60	15	228.0	1,023,210	20,154	1.97%	42

HCS - dense computation (Intel Xeon Platinum 8168)

(04/28/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_HC44rs	Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz	44	2	352.0	963,560	17,319	1.80%	84

Memory optimized

Easv4

(03/26/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2as_v4	AMD EPYC 7452 32-Core Processor	2	1	16.0	38,070	1,150	3.02%	210
Standard_E4as_v4	AMD EPYC 7452 32-Core Processor	4	1	32.0	75,733	1,444	1.91%	196

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E4-2as_v4	AMD EPYC 7452 32-Core Processor	2	1	32.0	38,105	943	2.47%	168
Standard_E8as_v4	AMD EPYC 7452 32-Core Processor	8	1	64.0	149,522	2,333	1.56%	210
Standard_E8-2as_v4	AMD EPYC 7452 32-Core Processor	2	1	64.0	38,103	1,078	2.83%	168
Standard_E8-4as_v4	AMD EPYC 7452 32-Core Processor	4	1	64.0	76,060	1,132	1.49%	168
Standard_E16as_v4	AMD EPYC 7452 32-Core Processor	16	2	128.0	288,136	4,720	1.64%	210
Standard_E16-4as_v4	AMD EPYC 7452 32-Core Processor	4	2	128.0	73,038	2,310	3.16%	196
Standard_E16-8as_v4	AMD EPYC 7452 32-Core Processor	8	2	128.0	144,266	2,782	1.93%	168
Standard_E20as_v4	AMD EPYC 7452 32-Core Processor	20	3	160.0	346,277	7,387	2.13%	14
Standard_E20as_v4	AMD EPYC	20	5	160.0	351,213	7,002	1.99%	196

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	7452 32-Core Processor							
Standard_E32as_v4	AMD EPYC 7452 32-Core Processor	32	4	256.0	561,950	7,679	1.37%	42
Standard_E32-8as_v4	AMD EPYC 7452 32-Core Processor	8	4	256.0	143,569	3,393	2.36%	182
Standard_E32-16as_v4	AMD EPYC 7452 32-Core Processor	16	4	256.0	283,614	5,018	1.77%	182
Standard_E48as_v4	AMD EPYC 7452 32-Core Processor	48	6	384.0	832,627	19,565	2.35%	210
Standard_E64as_v4	AMD EPYC 7452 32-Core Processor	64	8	512.0	1,097,588	26,100	2.38%	280
Standard_E64-16as_v4	AMD EPYC 7452 32-Core Processor	16	8	512.0	284,934	5,065	1.78%	154
Standard_E64-32as_v4	AMD EPYC 7452 32-Core Processor	32	8	512.0	561,951	9,691	1.72%	140
Standard_E96as_v4	AMD EPYC 7452 32-	96	12	672.0	N/A	-	-	-

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Core Processor							
Standard_E96-24as_v4	AMD EPYC 7452 32-Core Processor	24	11	672.0	423,442	8,504	2.01%	182
Standard_E96-48as_v4	AMD EPYC 7452 32-Core Processor	48	11	672.0	839,993	14,218	1.69%	70

Eav4

(03/27/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2a_v4	AMD EPYC 7452 32-Core Processor	2	1	16.0	38,008	995	2.62%	210
Standard_E4a_v4	AMD EPYC 7452 32-Core Processor	4	1	32.0	75,410	1,431	1.90%	196
Standard_E8a_v4	AMD EPYC 7452 32-Core Processor	8	1	64.0	148,810	2,630	1.77%	210
Standard_E16a_v4	AMD EPYC 7452 32-Core Processor	16	2	128.0	286,811	4,877	1.70%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E20a_v4	AMD EPYC 7452 32-Core Processor	20	3	160.0	351,049	6,268	1.79%	210
Standard_E32a_v4	AMD EPYC 7452 32-Core Processor	32	4	256.0	565,363	10,941	1.94%	126
Standard_E48a_v4	AMD EPYC 7452 32-Core Processor	48	6	384.0	837,493	15,803	1.89%	126
Standard_E64a_v4	AMD EPYC 7452 32-Core Processor	64	8	512.0	1,097,111	30,290	2.76%	336
Standard_E96a_v4	AMD EPYC 7452 32-Core Processor	96	12	672.0	N/A	-	-	-

EDSv4

(03/27/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	34,923	1,107	3.17%	336
Standard_E4ds_v4	Intel(R) Xeon(R)	4	1	32.0	66,921	1,294	1.93%	322

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	Platinum 8272CL CPU @ 2.60GHz							
Standard_E4-2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	32.0	34,909	811	2.32%	294
Standard_E8ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	132,164	2,102	1.59%	154
Standard_E8-2ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	64.0	35,031	965	2.76%	252
Standard_E8-4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	64.0	67,144	1,200	1.79%	182
Standard_E16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	265,181	2,634	0.99%	336
Standard_E16-4ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	128.0	67,155	1,596	2.38%	336
Standard_E16-8ds_v4	Intel(R) Xeon(R) Platinum	8	1	128.0	132,939	1,471	1.11%	336

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	8272CL CPU @ 2.60GHz							
Standard_E20ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	331,456	2,766	0.83%	336
Standard_E32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	531,560	5,700	1.07%	196
Standard_E32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	256.0	512,931	5,110	1.00%	14
Standard_E32-8ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	256.0	132,929	1,671	1.26%	182
Standard_E32-16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	256.0	265,471	2,268	0.85%	154
Standard_E48ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	768,428	6,891	0.90%	224
Standard_E64ds_v4	Intel(R) Xeon(R) Platinum 8272CL	64	2	504.0	1,005,554	78,398	7.80%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E64-16ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	2	504.0	260,677	3,340	1.28%	154
Standard_E64-32ds_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	504.0	514,504	4,082	0.79%	98

Edsv4 Isolated Extended

(04/05/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E80ids_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	80	2	504.0	N/A	-	-	-

EDv4

(03/26/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2d_v4	Intel(R) Xeon(R) Platinum 8272CL	2	1	16.0	34,916	1,063	3.04%	322

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E4d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	66,889	1,283	1.92%	336
Standard_E8d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	132,382	2,020	1.53%	322
Standard_E16d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	265,094	2,803	1.06%	336
Standard_E20d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	331,516	2,568	0.77%	336
Standard_E32d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	530,364	9,914	1.87%	336
Standard_E48d_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	761,410	21,640	2.84%	336
Standard_E64d_v4	Intel(R) Xeon(R) Platinum 8272CL	64	2	504.0	1,030,708	9,500	0.92%	322

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							

EIASv4

(04/05/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E96ias_v4	AMD EPYC 7452 32-Core Processor	96	12	672.0	N/A	-	-	-

Esv4

(03/25/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	31,390	2,786	8.88%	336
Standard_E4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	59,677	3,904	6.54%	336
Standard_E4-2s_v4	Intel(R) Xeon(R) Platinum 8272CL	2	1	32.0	31,443	2,480	7.89%	364

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	117,898	7,464	6.33%	406
Standard_E8-2s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	64.0	30,989	2,864	9.24%	406
Standard_E8-4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	64.0	59,589	4,762	7.99%	406
Standard_E16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	236,972	13,376	5.64%	406
Standard_E16-4s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	128.0	60,316	4,792	7.94%	406
Standard_E16-8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	128.0	117,057	6,569	5.61%	392
Standard_E20s_v4	Intel(R) Xeon(R) Platinum 8272CL	20	1	160.0	294,231	15,477	5.26%	406

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	481,943	22,707	4.71%	266
Standard_E32-8s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	256.0	116,774	6,791	5.82%	224
Standard_E32-16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	256.0	235,620	11,909	5.05%	266
Standard_E32-16s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	2	256.0	222,478	3,411	1.53%	14
Standard_E48s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	693,841	23,265	3.35%	182
Standard_E64s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	504.0	922,196	7,708	0.84%	182
Standard_E64-16s_v4	Intel(R) Xeon(R) Platinum 8272CL	16	2	504.0	224,499	3,955	1.76%	168

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E64-32s_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	2	504.0	441,521	30,939	7.01%	168

Esv4 Isolated Extended

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E80is_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	80	2	504.0	N/A	-	-	-

Ev4

(03/25/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	30,825	2,765	8.97%	406
Standard_E4_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	60,495	4,419	7.30%	406

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E8_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	64.0	119,562	8,628	7.22%	406
Standard_E16_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	237,126	13,328	5.62%	392
Standard_E20_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	299,681	17,288	5.77%	406
Standard_E32_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	486,051	28,085	5.78%	378
Standard_E48_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	48	2	384.0	686,812	20,561	2.99%	378
Standard_E64_v4	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	64	2	504.0	919,491	15,261	1.66%	378

Esv3 - Memory Optimized + Premium Storage

(04/05/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	16.0	23,704	2,155	9.09%	168
Standard_E2s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	16.0	21,917	1,521	6.94%	112
Standard_E2s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	28,549	3,105	10.88%	42
Standard_E4s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	32.0	46,370	4,256	9.18%	140
Standard_E4s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	32.0	47,178	3,791	8.04%	98
Standard_E4s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	53,636	4,231	7.89%	84
Standard_E16s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	128.0	175,905	7,275	4.14%	196

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E16s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	128.0	176,579	9,650	5.47%	112
Standard_E16s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	206,776	19,901	9.62%	28
Standard_E20s_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	160.0	219,370	7,086	3.23%	224
Standard_E20s_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	160.0	224,353	11,954	5.33%	98
Standard_E20s_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	280,572	13,326	4.75%	28

Ev3 - Memory Optimized

(04/05/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_E2_v3	Intel(R) Xeon(R) CPU E5-2673 v4	2	1	16.0	23,304	2,074	8.90%	182

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_E2_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	16.0	24,513	2,428	9.90%	112
Standard_E2_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	16.0	26,171	153	0.58%	14
Standard_E4_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	32.0	46,224	3,713	8.03%	238
Standard_E4_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	32.0	49,200	3,457	7.03%	42
Standard_E4_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	32.0	53,476	4,219	7.89%	42
Standard_E8_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	64.0	90,915	7,711	8.48%	224
Standard_E8_v3	Intel(R) Xeon(R) Platinum 8171M	8	1	64.0	89,968	5,738	6.38%	84

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_E16_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	128.0	174,677	7,198	4.12%	210
Standard_E16_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	128.0	180,002	14,028	7.79%	98
Standard_E16_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	128.0	217,439	13,826	6.36%	28
Standard_E20_v3	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	160.0	221,787	10,447	4.71%	238
Standard_E20_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	160.0	234,854	10,704	4.56%	70
Standard_E20_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	160.0	293,226	3,480	1.19%	14
Standard_E32_v3	Intel(R) Xeon(R) CPU E5-2673 v4	32	2	256.0	349,134	13,895	3.98%	210

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_E32_v3	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	32	1	256.0	352,509	14,689	4.17%	84
Standard_E32_v3	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	32	1	256.0	413,946	2,239	0.54%	14

DSv2 - Memory Optimized + Premium Storage

(05/24/2021 PBIID:9198755 OS: MicrosoftWindowsServer-WindowsServer-2019-Datacenter-latest)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS11_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	14.0	33,150	2,097	6.33%	112
Standard_DS11_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	14.0	34,294	3,107	9.06%	182
Standard_DS11_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	14.0	33,447	2,690	8.04%	98

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS11_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	14.0	37,222	3,740	10.05%	126
Standard_DS11-1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	14.0	16,994	1,127	6.63%	98
Standard_DS11-1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	14.0	17,538	2,029	11.57%	112
Standard_DS11-1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	14.0	17,255	1,647	9.54%	140
Standard_DS11-1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	14.0	19,463	3,017	15.50%	140
Standard_DS12_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	28.0	65,209	2,830	4.34%	98
Standard_DS12_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	28.0	70,755	5,315	7.51%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS12_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	28.0	62,898	3,666	5.83%	126
Standard_DS12_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	28.0	76,876	7,628	9.92%	140
Standard_DS12-1_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	1	1	28.0	17,253	1,094	6.34%	56
Standard_DS12-1_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	1	1	28.0	17,748	2,342	13.20%	182
Standard_DS12-1_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	1	1	28.0	16,866	1,506	8.93%	154
Standard_DS12-1_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	1	1	28.0	18,925	2,461	13.00%	140
Standard_DS12-2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	28.0	35,288	2,477	7.02%	98

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS12-2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	28.0	35,716	2,923	8.19%	140
Standard_DS12-2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	28.0	33,884	3,285	9.70%	112
Standard_DS12-2_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	28.0	39,473	3,712	9.40%	140
Standard_DS13_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	56.0	136,322	6,996	5.13%	70
Standard_DS13_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	56.0	127,110	4,330	3.41%	154
Standard_DS13_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	56.0	126,037	6,952	5.52%	126
Standard_DS13_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	56.0	158,319	11,403	7.20%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS13-2_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	56.0	34,673	1,705	4.92%	56
Standard_DS13-2_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	56.0	34,869	4,134	11.86%	182
Standard_DS13-2_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	56.0	33,986	2,411	7.09%	112
Standard_DS13-2_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	56.0	39,104	3,607	9.22%	140
Standard_DS13-4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	56.0	67,217	3,774	5.61%	84
Standard_DS13-4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	56.0	69,377	6,103	8.80%	210
Standard_DS13-4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	56.0	66,193	6,252	9.45%	70

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS13-4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	56.0	76,528	6,662	8.71%	126
Standard_DS14_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	112.0	255,718	11,055	4.32%	98
Standard_DS14_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	16	1	112.0	263,225	11,664	4.43%	182
Standard_DS14_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	112.0	257,737	11,726	4.55%	84
Standard_DS14_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	112.0	296,962	22,427	7.55%	126
Standard_DS14-4_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	2	112.0	67,170	4,554	6.78%	70
Standard_DS14-4_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	112.0	66,748	4,294	6.43%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS14-4_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	4	1	112.0	67,116	7,254	10.81%	140
Standard_DS14-4_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	112.0	75,100	4,719	6.28%	154
Standard_DS14-8_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	2	112.0	125,558	1,961	1.56%	84
Standard_DS14-8_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	112.0	128,904	4,055	3.15%	84
Standard_DS14-8_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	112.0	127,309	8,301	6.52%	182
Standard_DS14-8_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	112.0	148,717	13,028	8.76%	140
Standard_DS15_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	2	140.0	314,358	3,189	1.01%	56

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_DS15_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	140.0	324,517	14,805	4.56%	168
Standard_DS15_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	140.0	312,229	16,417	5.26%	126
Standard_DS15_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	140.0	369,680	36,859	9.97%	126
Standard_DS15i_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	2	140.0	330,307	18,294	5.54%	28
Standard_DS15i_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	140.0	330,397	17,469	5.29%	294
Standard_DS15i_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	20	1	140.0	371,086	33,246	8.96%	140

Dv2 - Memory Optimized

(05/24/2021 PBIID:9198755 OS: MicrosoftWindowsServer-WindowsServer-2019-Datacenter-latest)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_D11_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	2	1	14.0	33,085	1,835	5.55%	196
Standard_D11_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	2	1	14.0	33,998	3,523	10.36%	168
Standard_D11_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	2	1	14.0	32,964	2,690	8.16%	140
Standard_D11_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	2	1	14.0	39,330	3,092	7.86%	28
Standard_D12_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	4	1	28.0	65,877	3,537	5.37%	196
Standard_D12_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	4	1	28.0	66,128	4,438	6.71%	210
Standard_D12_v2	Intel(R) Xeon(R) Platinum 8171M	4	1	28.0	63,397	4,021	6.34%	98

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							
Standard_D12_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	4	1	28.0	80,559	3,760	4.67%	14
Standard_D13_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	8	1	56.0	133,275	7,394	5.55%	182
Standard_D13_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	8	1	56.0	127,557	3,854	3.02%	210
Standard_D13_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	8	1	56.0	120,793	1,444	1.20%	70
Standard_D13_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	8	1	56.0	151,473	7,667	5.06%	42
Standard_D14_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	16	2	112.0	253,422	9,024	3.56%	224
Standard_D14_v2	Intel(R) Xeon(R) CPU E5-2673 v4	16	1	112.0	261,437	12,701	4.86%	140

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	@ 2.30GHz							
Standard_D14_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	16	1	112.0	246,994	6,166	2.50%	98
Standard_D14_v2	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz	16	1	112.0	315,200	15,164	4.81%	28
Standard_D15_v2	Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz	20	2	140.0	315,711	8,560	2.71%	252
Standard_D15_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	140.0	321,475	12,246	3.81%	168
Standard_D15_v2	Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz	20	1	140.0	314,692	11,528	3.66%	56
Standard_D15i_v2	Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz	20	1	140.0	323,411	10,041	3.10%	462
Standard_D15i_v2	Intel(R) Xeon(R) Platinum 8272CL	20	1	140.0	370,436	5,018	1.35%	28

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
	CPU @ 2.60GHz							

Storage optimized

Lsv2 - Storage Optimized

(04/29/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_L8s_v2	AMD EPYC 7551 32-Core Processor	8	1	64.0	101,433	844	0.83%	448
Standard_L16s_v2	AMD EPYC 7551 32-Core Processor	16	2	128.0	200,664	2,866	1.43%	448
Standard_L32s_v2	AMD EPYC 7551 32-Core Processor	32	4	256.0	396,781	7,237	1.82%	462
Standard_L48s_v2	AMD EPYC 7551 32-Core Processor	48	6	384.0	584,059	13,101	2.24%	112
Standard_L64s_v2	AMD EPYC 7551 32-Core Processor	64	8	512.0	768,442	19,518	2.54%	70

Ls - Storage Optimized + Premium Storage

(05/25/2021 PBIID:9198755)

Expand table

VM Size	CPU	vCPUs	NUMA Nodes	Memory(GiB)	Avg Score	StdDev	StdDev%	#Runs
Standard_L4s	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	4	1	32.0	69,902	6,012	8.60%	294
Standard_L8s	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	8	1	64.0	133,859	7,026	5.25%	280
Standard_L16s	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	16	1	128.0	259,512	7,269	2.80%	252
Standard_L32s	Intel(R) Xeon(R) CPU E5-2698B v3 @ 2.00GHz	32	2	256.0	506,359	7,472	1.48%	252

About CoreMark

CoreMark is a benchmark that tests the functionality of a microcontroller (MCU) or central processing unit (CPU). CoreMark isn't system dependent, so it functions the same regardless of the platform (for example, big or little endian, high-end or low-end processor).

Windows numbers were computed by running CoreMark on Windows Server 2019. CoreMark was configured with the number of threads set to the number of virtual CPUs, and concurrency set to `PThreads`. The target number of iterations was adjusted based on expected performance to provide a runtime of at least 20 seconds (typically much longer). The final score represents the number of iterations completed divided by the number of seconds it took to run the test. Each test was run at least seven times on each VM. Test run

dates shown above. Tests run on multiple VMs across Azure public regions the VM was supported in on the date run.

Windows numbers were computed by running CoreMark on Windows Server 2019. CoreMark was configured with the number of threads set to the number of virtual CPUs, and concurrency set to `PThreads`. The target number of iterations was adjusted based on expected performance to provide a runtime of at least 20 seconds (typically much longer). The final score represents the number of iterations completed divided by the number of seconds it took to run the test. Each test was run at least seven times on each VM. Test run dates shown above. Tests run on multiple VMs across Azure public regions the VM was supported in on the date run. (Coremark doesn't properly support more than 64 vCPUs on Windows, therefore SKUs with > 64 vCPUs have been marked as N/A.)

Running Coremark on Azure VMs

Download:

CoreMark is an open source tool that can be downloaded from GitHub.

Building and Running:

To build and run the benchmark, type:

```
> make
```

Full results are available in the files `run1.log` and `run2.log`. `run1.log` contains CoreMark results with performance parameters. `run2.log` contains benchmark results with validation parameters.

Run Time:

By default, the benchmark will run between 10-100 seconds. To override, use `ITERATIONS=N`

```
% make ITERATIONS=10
```

above flag will run the benchmark for 10 iterations. **Results are only valid for reporting if the benchmark ran for at least 10 seconds!**

Parallel Execution:

Use `XCFLAGS=-DMULTITHREAD=N` where N is number of threads to run in parallel. Several implementations are available to execute in multiple contexts.

```
% make XCFLAGS="-DMULTITHREAD=4 -DUSE_PTHREAD"
```

The above will compile the benchmark for execution on 4 cores.

Recommendations for best results

- The benchmark needs to run for at least 10 seconds, probably longer on larger systems.
- All source files must be compiled with same flags.
- Don't change source files other than `core_portme*` (use `make check` to validate)
- Multiple runs are suggested for best results.

GPU Series

Performance of GPU based VM series is best understood by using GPU appropriate benchmarks and running at the scale required for your workloads. Azure ranks among the best there:

- Top 10 Supercomputer: November 2021 | TOP500 (Azure powered #10: Voyager-EUS2)
- Machine Learning: MLCommons Training: v1.1 Results | MLCommons (2 highest at scale and largest in the cloud)

Next steps

- For storage capacities, disk details, and other considerations for choosing among VM sizes, see Sizes for virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

States and billing status of Azure Virtual Machines

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Virtual Machines (VM) instances go through different states. There are *provisioning* and *power* states. This article describes these states and highlights when customers are billed for instance usage.

Get states using Instance View

The instance view API provides VM running-state information. For more information, see Virtual Machines - Instance View.

Azure Resources Explorer provides a simple UI for viewing the VM running state: Resource Explorer.

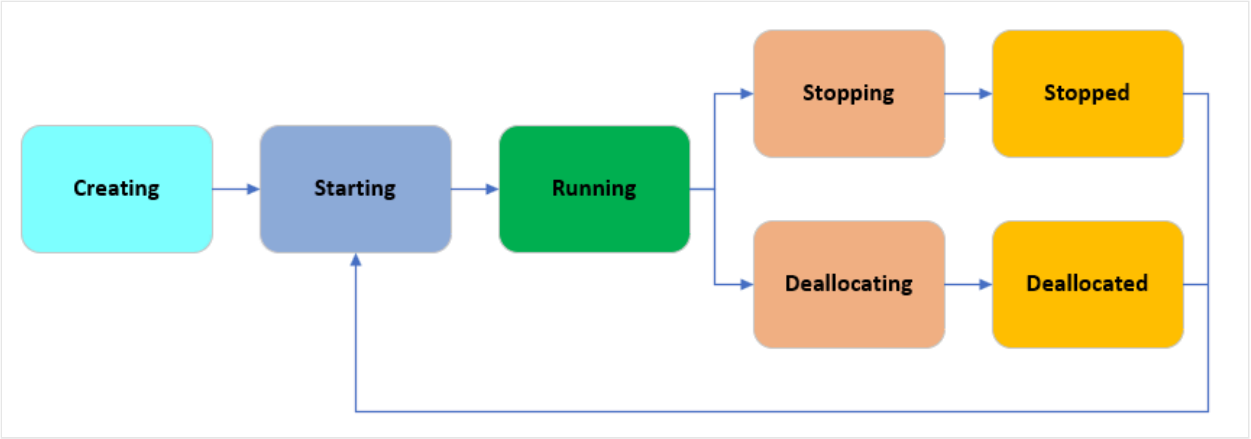
The VM provisioning state is available, in slightly different forms, from within the VM properties `provisioningState` and the InstanceView. In the VM InstanceView, there's an element within the `status` array in the form of `ProvisioningState/<state>[/<errorCode>]`.

To retrieve the power state of all the VMs in your subscription, use the Virtual Machines - List All API with parameter `statusOnly` set to `true`.

> **Note**
>
> Virtual Machines - List All API with parameter `statusOnly` set to `true` retrieves the power states of all VMs in a subscription. However, in some rare situations, the power state may not available due to intermittent issues in the retrieval process. In such situations, we recommend retrying using the same API or using Azure Resource Health to check the power state of your VMs.

Power states and billing

The power state represents the last known state of the VM.



The following table provides a description of each instance state and indicates whether that state is billed for instance usage.

Expand table

Power state	Description	Billing
Creating	Virtual machine is allocating resources.	Not Billed*
Starting	Virtual machine is powering up.	Billed
Running	Virtual machine is fully up. This state is the standard working state.	Billed
Stopping	This state is transitional between running and stopped.	Billed
Stopped	The virtual machine is allocated on a host but not running. Also called *PoweredOff* state or *Stopped (Allocated)*. This state can be result of invoking the `PowerOff` API operation or invoking shutdown from within the guest OS. The *Stopped* state might also be observed briefly during VM creation or while starting a VM from *Stopped (Deallocated)* state.	Billed
Deallocating	This state is transitional between *Running* and *Deallocated*.	Not billed*
Deallocated	The virtual machine has released the lease on the underlying hardware. If the machine is powered off it is shown as *Stopped (Deallocated)*. If it has entered hibernation it is shown as *Hibernated (Deallocated)*	Not billed*

* Some Azure resources, such as Disks and Networking continue to incur charges.

Example of PowerState in JSON:

JSON

```
{
  "code": "PowerState/running",
  "level": "Info",
```

```
    "displayStatus": "VM running"
  }
```

Provisioning states

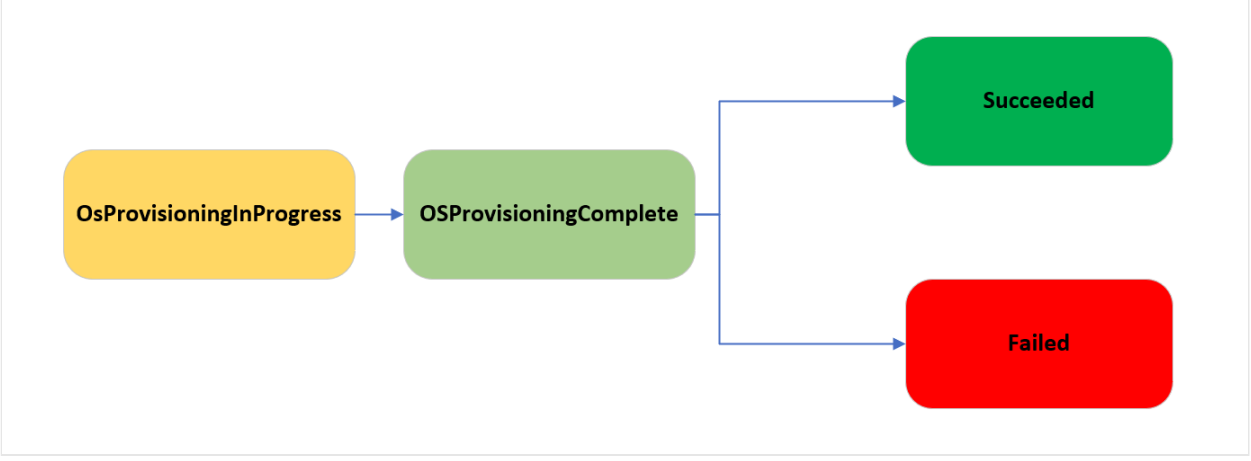
The provisioning state is the status of a user-initiated, control-plane operation on the VM. These states are separate from the power state of a VM.

Expand table

Provisioning state	Description
Creating	Virtual machine is being created.
Updating	Virtual machine is updating to the latest model. Some non-model changes to a virtual machine such as start and restart fall under the updating state.
Failed	Last operation on the virtual machine resource was unsuccessful.
Succeeded	Last operation on the virtual machine resource was successful.
Deleting	Virtual machine is being deleted.
Migrating	Seen when migrating from Azure Service Manager to Azure Resource Manager.

OS Provisioning states

OS Provisioning states only apply to virtual machines created with a generalized OS image. Specialized images and disks attached as OS disk don't display these states. The OS provisioning state isn't shown separately. It's a substate of the Provisioning State in the VM InstanceView. For example, `ProvisioningState/creating/osProvisioningComplete`.



Expand table

OS Provisioning state	Description
OSProvisioningInProgress	The VM is running and the initialization (setup) of the Guest OS is in progress.
OSProvisioningComplete	This state is a short-lived state. The virtual machine quickly transitions from this state to *Success*. If extensions are still being installed, you continue to see this state until installation is complete.
Succeeded	The user-initiated actions have completed.
Failed	Represents a failed operation. For more information and possible solutions, see the error code.

Troubleshooting VM states

To troubleshoot specific VM state issues, see Troubleshoot Windows VM deployments and Troubleshoot Linux VM deployments.

To troubleshoot hibernation, see Troubleshoot VM hibernation.

For other troubleshooting help visit Azure Virtual Machines troubleshooting documentation.

Next steps

- Review the Azure Cost Management and Billing documentation
- Use the Azure Pricing calculator to plan your deployments.
- Learn more about monitoring your VM, see Monitor virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Check vCPU quotas

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

The vCPU quotas for virtual machines and scale sets are arranged in two tiers for each subscription, in each region. The first tier is the Total Regional vCPUs, and the second tier is the various VM size family cores such as the D-series vCPUs. Anytime a new VM is deployed the vCPUs for the VM must not exceed the vCPU quota for the VM size family or the total regional vCPU quota. If you exceed either of those quotas, the VM deployment won't be allowed. There is also a quota for the overall number of virtual machines in the region. The details on each of these quotas can be seen in the **Usage + quotas** section of the **Subscription** page in the Azure portal, or you can query for the values using the Azure CLI.

> **Note**
>
> Quota is calculated based on the total number of cores in use both allocated and deallocated. If you need additional cores, request a quota increase or delete VMs that are no longer needed.

Quota and capacity

Azure checks quota and capacity separately. Quota is your subscription's permission to deploy a certain number of vCPUs or VMs. Capacity is the actual compute resources available in the selected region or availability zone.

Even with sufficient quota, deployments can still fail if Azure doesn't have enough capacity for your requested VM size in the target region or zone. To improve deployment success, try a different VM size, zone, or region. For guaranteed capacity, see On-demand capacity reservation.

Check usage

CLI

You can check your quota usage using az vm list-usage.

Azure CLI

```
az vm list-usage --location "Central US" -o table
```

The output should look something like this:

```
Name                                 CurrentValue    Limit
-------------------------------  --------------  -------
Availability Sets                               0     2000
Total Regional vCPUs                           29      100
Virtual Machines                                7    10000
Virtual Machine Scale Sets                      0     2000
Standard DSv3 Family vCPUs                      8      100
Standard DSv2 Family vCPUs                      3      100
Standard Dv3 Family vCPUs                       2      100
Standard D Family vCPUs                         8      100
Standard Dv2 Family vCPUs                       8      100
Basic A Family vCPUs                            0      100
Standard A0-A7 Family vCPUs                     0      100
Standard A8-A11 Family vCPUs                    0      100
Standard DS Family vCPUs                        0      100
Standard G Family vCPUs                         0      100
Standard GS Family vCPUs                        0      100
Standard F Family vCPUs                         0      100
Standard FS Family vCPUs                        0      100
Standard Storage managed disks                  5    10000
Premium Storage managed disks                   5    10000
...
```

Reserved VM Instances

Reserved VM Instances, scoped to a single subscription without VM size flexibility, will add a new aspect to the vCPU quotas. These values describe the number of instances of the stated size that must be deployable in the subscription. They work as a placeholder in the quota system to ensure that quota is reserved to ensure Azure reservations are deployable in the subscription. For example, if a specific subscription has 10 Standard_D1 reservations the usages limit for Standard_D1 reservations will be 10. Azure will ensure that there are always at least 10 vCPUs available in the Total Regional vCPUs quota to be used for Standard_D1 instances and there are at least 10 vCPUs available in the Standard D Family vCPU quota to be used for Standard_D1 instances.

If a quota increase is required to either purchase a Single Subscription RI, you can request a quota increase on your subscription.

Related content

For more information about billing and quotas, see Azure subscription and service limits, quotas, and constraints.

Last updated on 04/10/2026

Constrained vCPU sizes for database workloads

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

> **Tip**
>
> Try the **Virtual Machine selector tool** to find other sizes that best fit your workload.

Some database workloads require high memory, storage, and I/O bandwidth, but don't benefit from a high number of cores. Products designed for these workloads are often licensed on a per-CPU-core basis. This licensing system means that a VM size with otherwise ideal specifications but an excessive vCPU count can cause a substantial increase in licensing costs. To help manage your costs, Azure offers predefined VM sizes with a lowered vCPU count to help to reduce the cost of software licensing, while maintaining the same memory, storage, and I/O bandwidth.

The original size's available vCPU count can be reduced to either one half or one quarter of the original size specification. These new VM sizes have a suffix that specifies the number of available vCPUs to make them easier for you to identify. Learn more about VM size naming conventions.

Example workload

The licensing fees charged for SQL Server are based on the available vCPU count. Third party products should count the available vCPUs and report it as the amount to be used and licensed. This constraint results in a 50% to 75% increase in the ratio of the VM specs to available (billable) vCPUs while maintaining the benefits of the original VM size's specifications.

Specification comparison

The Standard_E32s_v5 VM size comes with 32 vCPUs, 256 GiB of RAM, 32 disks, and 80,000 IOPs of I/O bandwidth. The pre-defined Standard_E32-16s_v5 and Standard_E32-8s_v5 VM sizes comes with 16 and 8 active vCPUs respectively, while maintaining the memory, storage, and I/O bandwidth specifications of the Standard_E32s_v5.

Expand table

Size type	Size name	Disabled vCPUs	Active vCPUs	Memory, Storage, I/O
Base (Original)	Standard_E32s_v5	0	**32**	256 GiB, 32 Disks, 80,000 IOPs
Constrained	Standard_E32-**16**s_v5	16	**16**	256 GiB, 32 Disks, 80,000 IOPs
Constrained	Standard_E32-**8**s_v5	24	**8**	256 GiB, 32 Disks, 80,000 IOPs

> **Note**
>
> Disabled vCPUs are not available for use by the VM. If your VM has multiple workloads assigned to it then it may require more vCPUs. If possible, relocate secondary tasks to another virtual machine to avoid increased licensing costs.

Pricing

At this time, the VM pricing, remains the same as the original size. For more information, see Azure VM sizes for more cost-effective database workloads.

List of available sizes with constrained vCPUs

M-family sizes

Expand table

Size name	Active vCPUs	Base size
Standard_M8-2ms	2	M8ms
Standard_M8-4ms	4	M8ms
Standard_M16-4ms	4	M16ms
Standard_M16-8ms	8	M16ms
Standard_M32-8ms	8	M32ms
Standard_M32-16ms	16	M32ms

Size name	Active vCPUs	Base size
Standard_M64-32ms	32	M64ms
Standard_M64-16ms	16	M64ms
Standard_M128-64ms	64	M128ms
Standard_M128-32ms	32	M128ms
Standard_M416-208s_v2	208	M416s_v2
Standard_M416-208ms_v2	208	M416ms_v2
Standard_M64-32bds_1_v3	32	M64bds_1_v3
Standard_M96-48bds_2_v3	48	M96bds_2_v3
Standard_M128-64bds_3_v3	64	M128bds_3_v3
Standard_M176-88bds_4_v3	88	M176bds_4_v3
Standard_M128-64bds_v3	64	M128bds_v3
Standard_M176-88bds_v3	88	M176bds_v3
Standard_M128-64bs_v3	64	M128bs_v3
Standard_M176-88bs_v3	88	M176bs_v3

Other standard sizes

- Compute optimized
- Memory optimized
- Storage optimized
- GPU
- High performance compute

Next steps

Learn more about how Azure compute units (ACU) can help you compare compute performance across Azure SKUs.

Last updated on 10/16/2025

VM vCore Customization Feature: Disable Simultaneous Multi-Threading (SMT/HT) and Configurable Constrained Cores (Preview)

VM vCore Customization is a new Azure VM feature that gives you greater control over the CPU resources of a virtual machine. It consists of two related capabilities:

Disable Simultaneous Multi-Threading (Threads Per Core = 1): Allows you to run a VM with only one thread per physical CPU core, effectively turning off Simultaneous Multithreading (SMT). Disabling gives your VM full use of each physical core, which can improve performance for certain workloads (like some HPC or latency-sensitive applications) that benefit from exclusive core access.

Configurable Constrained Cores (Customize vCPUs): Allows you to **choose a custom number of vCPUs** for a new VM, lower than the default count for that VM size. It lets you allocate only the CPU cores you need, for example, to reduce licensing costs for software that is licensed per core (such as databases or analytics servers) while still getting the full memory and I/O of a larger VM.

Benefits: With these features, you can optimize VMs for both performance and cost:

Performance: Disabling hyperthreading can provide more consistent and sometimes higher single-thread performance by eliminating contention between threads on the same core.

Cost Optimization: Reducing the vCPU count of a VM can lower costs for software that charges per CPU. You could run a memory-intensive SQL Server on a VM with fewer vCPUs active, cutting SQL licensing fees, without paying for unused CPU capacity.

There's *no extra charge* to use these CPU configuration options. The base VM price remains the same as if you deployed the full-size VM with default settings. However, customers get reduced licensing costs for software billed per vCPU.

VM Customization Settings Configuration

You can configure the "Threads per core" and "vCPUs available" settings using the Azure portal, Azure Resource Manager templates (ARM), or command-line tools.

Azure portal

In the Azure portal, the VM creation workflow has a UI for these options.

Start creating a VM as usual (for example, click **Create a resource** > **Virtual Machine** and fill out the Basics tab).

1. In the Size section of the Basics tab, select a VM size that you want to use. Under the size selection, click the "Customize cores" button. This opens more fields for VM Customization.

2. To disable SMT, set Threads per core to 1. (Leave it at 2 if you want to keep hyperthreading enabled.)

3. To the vCPU count, set vCPUs available to your desired number of vCPUs. The portal provides valid values for the chosen VM size.

4. Continue with the rest of the VM creation (set up disks, networking, etc.) and create the VM.

Once the VM is deployed, it has the specified number of vCPUs. If you set Threads per core to 1, the VM's OS sees half the usual number of processors (since hyperthreading is off). If you reduced vCPUs, it would see that lower count.

Azure CLI

To disable SMT and configure cores during instance launch:

To disable SMT/HT, use the Azure CLI command and specify a value of 1 for vCPUsPerCore for the--cpu-options parameter. To configure cores, specify the number of CPU cores for vCPUsAvailable. In this example, to specify the default CPU core count for a Standard_D8s_v6 instance, specify a value of 8.

azure

```
Az vm create --resource-group ccctest-rg-01 --name ccctestvm01 --image Ubuntu2204 --size Standard_D8s_v6 --location eastus2euap --admin-username azureuser --generate-ssh-keys --public-ip-address """ --v-cpus-available 4 --v-cpus-per-core 1
```

PowerShell

To disable SMT and configure cores during instance launch

Use PowerShell and specify the properties on the underlying configuration object. To disable SMT/HT, specify a value of 1 for vCPUsPerCore for the --cpu-options parameter. To configure cores, specify the number of CPU cores for vCPUsAvailable.

PowerShell

```
$vmConfig = New-AzVMConfig -VMName "MyVM" -VMSize "Standard_D8s_v6"

$vmConfig.HardwareProfile.VmSizeProperties = New-Object
Microsoft.Azure.Management.Compute.Models.VMSizeProperties

$vmConfig.HardwareProfile.VmSizeProperties.VCPUsAvailable = 4

$vmConfig.HardwareProfile.VmSizeProperties.VCPUsPerCore = 1
```

Then proceed to set OS, network, etc., and use New-AzVM to create the VM. This approach uses the Azure PowerShell SDK objects directly to inject the values.

ARM Template (Azure Resource Manager)

For automation or scenarios where you need to deploy via infrastructure-as-code, you can use Azure Resource Manager templates to specify these CPU options. This can be used to disable SMT and deploy customized VMs via CLI or PowerShell as well (by deploying a template).

In your ARM template's resource definition for the virtual machine, the CPU options are specified under the hardwareProfile property of the VM. Specifically, you use vmSizeProperties inside hardwareProfile to set the values:

vCPUsPerCore - Set this to 1 to disable hyperthreading (for example, 1 thread per core). Omit this property or set to null/2 to use the default hyperthreading (2 threads per core).

vCPUsAvailable - Set this to the number of vCPUs you want active. If this property isn't set, the VM uses the default number of vCPUs for that size.

Here are brief examples of ARM template snippets for different scenarios:

Disable SMT (SMT/HT Off)

This snippet shows the setting to turn off SMT on a VM (the VM uses 1 thread per core)

JSON

```
"properties": {

"hardwareProfile": {

"vmSize": " Standard_D8s_v6",

"vmSizeProperties": {
```

```
"vCPUsPerCore": 1

}

},

...

}
```

In this case, if Standard_D8s_v6 normally has 8 vCPUs (4 cores * 2 threads), setting vCPUsPerCore: 1 means the VM has 4 vCPUs (one per core).

Constrain vCPU Count (Customize Cores)

This snippet shows a VM configured to use a specific number of vCPUs (fewer than the default)

JSON

```
"properties": {

"hardwareProfile": {

"vmSize": " Standard_D8s_v6",

"vmSizeProperties": {

"vCPUsAvailable": 2

}

},

...

}
```

Here, we requested two cores. On Standard_D8s_v6 (which is hyperthreaded by default), 2 *physical* cores are allocated, and since SMT is still on by default (2 threads per core), the VM has 4 logical vCPUs.

Disable SMT and Customize vCPUs

You can combine both settings as shown:

JSON

```
"properties": {

"hardwareProfile": {

"vmSize": " Standard_D8s_v6",

"vmSizeProperties": {

"vCPUsPerCore": 1,

"vCPUsAvailable": 2

}

},

...

}
```

In this example, vCPUsPerCore: 1 disables SMT, and vCPUsAvailable: 2 then requests 2 vCPUs. With SMT off, those 2 correspond one-to-one with two physical cores (no threading). The VM has two logical processors in the OS.

Make sure to use an API version **2021-07-01 or later** for the Microsoft.Compute/virtualMachines resource in your template, as that's when these properties were introduced.

Identify Supported vCores for Configuration

To determine which vCPUs can be constrained in a specific region, you can use the Azure CLI or the Azure portal.

Using Azure CLI

Run the following command to retrieve the list of compute SKUs for your subscription and region:

Azure

```
az vm list-skus --location {location} --resource-type virtualMachines --query "
[name=='VM_NAME_HERE']
```

- This command exports shows the supported vCPU configurations. The "vCPUsConstraintsAllowed" field outlines supported vCores.

Using Azure Portal

If you request a vCPU configuration that isn't supported, the portal will display an error message and provide a list of supported vCPU options for that VM size.

Considerations

Most Azure VM families support these features, but there are some important rules and considerations to understand when using the feature:

- You can only disable hyperthreading on VM sizes that use hyperthreading by default (for example, VMs with 2 threads per core).
- You can only *reduce* the number of vCPUs, not increase it beyond the VM's default. The vCPUsAvailable value specified must be less than or equal to the default vCPU count of the chosen VM size.
- On VM sizes that are hyperthreaded (default 2 threads/core), any custom vCPU count must be an even number.
- You can disable hyperthreading and constrain vCPUs at the same time on the same VM. In this case, both above rules apply.
- CPU options can only be specified at VM creation time or during a resize operation. You cannot dynamically adjust the core count or SMT setting on an allocated VM, this update requires the VM to be deallocated.
- If you move to a new VM size in the same family that also supports the feature, your settings are carried over by default.
- If you resize to a VM size that doesn't support the setting, the operation is blocked or error.
- Anytime you resize a VM (either within the same series or to a different series), a VM reboot occurs. Plan for downtime during the resize operation.
- In preview, only first-party Azure Marketplace images (Windows Server, Ubuntu, Red Hat, SUSE, etc.) and custom images are supported. The specialized Marketplace offerings such as SQL Server on Virtual Machines are not supported.

Last updated on 11/03/2025

Azure VM sizes with no local temporary disk

> **Tip**
>
> Try the Virtual Machine selector tool to find other sizes that best fit your workload.

This article provides answers to frequently asked questions (FAQ) about Azure VM sizes that don't have a local temporary disk (that is, no local temp disk).

What does no local temp disk mean?

Traditionally, we have had VM sizes (for example, Standard_D2s_v3, Standard_E48_v3) that include a small local disk (that is, a D: Drive). With the VM series such as Dasv5 and Easv5 that small local disk no longer exists. However, you can still attach Standard HDD, Premium SSD or Ultra Disk to use as remote storage.

What if I still want a local temp disk?

If your workload requires a local temporary disk, we still offer sizes such as the Dadsv5.

> **Note**
>
> Local temporary disk isn't persistent; to ensure your data is persistent, please use Standard HDD, Premium SSD or Ultra Disk options.

Can I resize a VM size that has a local temp disk to a VM size with no local temp disk?

Resize between disk and diskless sizes is supported for Linux VMs. For Windows, the only combinations allowed for resizing are:

1. VM (with local temp disk) -> VM (with local temp disk); and
2. VM (with no local temp disk) -> VM (with no local temp disk).

If interested in a work around for Windows, please see next question.

> **Note**

If an image depends on the resource disk, or a pagefile or swapfile exists on the local temp disk, the diskless images won't work—instead, use the 'with disk' alternative.

How do I migrate my Windows VM from a size with local temp disk to a VM size with no local temp disk?

You can migrate by following these steps:

1. Connect to your Virtual Machine that has a local temporary disk (for example, a D: Drive) as a local admin.

2. Follow the guidelines on the "Temporarily move pagefile.sys to C drive" section of Use the D: drive as a data drive on a Windows VM to move the page file from the local temporary disk (D: drive) to the C: drive.

   > ⓘ **Note**
   >
   > Follow the guidelines on the "Temporarily move pagefile.sys to C drive" section of Use the D: drive as a data drive on a Windows VM to move page file from the local temporary disk (D: drive) to C: drive. **Deviation from the steps outlined will lead to the error message - "Unable to resize the VM since changing from resource disk to non-resource disk VM size and vice-versa is not allowed.**

3. Take a snapshot of the VM by following the steps outlined in Create a snapshot using the portal or Azure CLI.

4. Use snapshot to create a new diskless VM (such as, Dv5, Dsv5, Dasv5, Ev5, Esv5, Easv5 series) by following the steps outlined in Create a virtual machine from a snapshot with CLI.

Are VM sizes with no local temp disk supported by both Linux and Windows Operating Systems (OS)?

Yes.

Will this break my custom scripts, custom images or OS images that have scratch files or page files on a local temp disk?

If the custom OS image points to the local temp disk, the image might not work correctly with this diskless size.

Next steps

In this document, you learned more about the most frequent questions related to Azure VMs with no local temp disk. For more information about these VM sizes, see the following articles:

- Specifications for Dv5 and Dsv5-series
- Specifications for Dasv5 and Dadsv5-series
- Specifications for Ev5 and Esv5-series
- Specifications for Easv5 and Eadsv5-series

Store and share resources in Azure Compute Gallery

08/06/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Compute Gallery helps you build structure and organization around your Azure resources, like images and applications. Compute Gallery provides:

- Global replication.[1]
- Versioning and grouping of resources for easier management.
- Highly available resources with zone-redundant storage (ZRS) accounts, in regions that support availability zones. ZRS offers resilience against zonal failures.
- Premium storage support with locally redundant storage (LRS).
- Sharing to the community, across subscriptions, and between Active Directory tenants.
- Scaling your deployments with resource replicas in each region.

With a gallery, you can share your resources with everyone. Or you can limit sharing to particular users, service principals, or Active Directory groups within your organization. You can replicate resources to multiple regions, for quicker scaling of your deployments.

[1] The Azure Compute Gallery service isn't a global resource. For disaster recovery scenarios, the best practice is to have at least two galleries in separate regions.

Images

For more information about storing images in Azure Compute Gallery, see Store and share images in Azure Compute Gallery.

VM apps

Although you can create an image of a virtual machine (VM) with apps preinstalled, you would need to update your image each time an application changes. Separating your application installation from your VM images means there's no need to publish a new image for every line of code change.

For more information about storing applications in Azure Compute Gallery, see VM applications.

Regional support

All public regions can be target regions, but certain regions require that customers go through a request process to gain access. To request the addition of a subscription to the allowlist for a region such as Australia Central or Australia Central 2, submit an access request.

Limits

The following limits apply when you deploy resources by using Azure Compute Gallery:

- You can have a maximum of 100 galleries per subscription, per region.
- You can have a maximum of 1,000 image definitions per subscription, per region.
- You can have a maximum of 10,000 image versions per subscription, per region.
- You can have a maximum 100 replicas per image version. However, 50 replicas should be sufficient for most use cases.
- The image size should be less than 2 TB, but you can use shallow replication to support larger image sizes (up to 32 TB).
- Resource movement isn't supported for Azure Compute Gallery resources.

For more information and for examples of how to check your current usage, see Check resource usage against limits.

Scaling

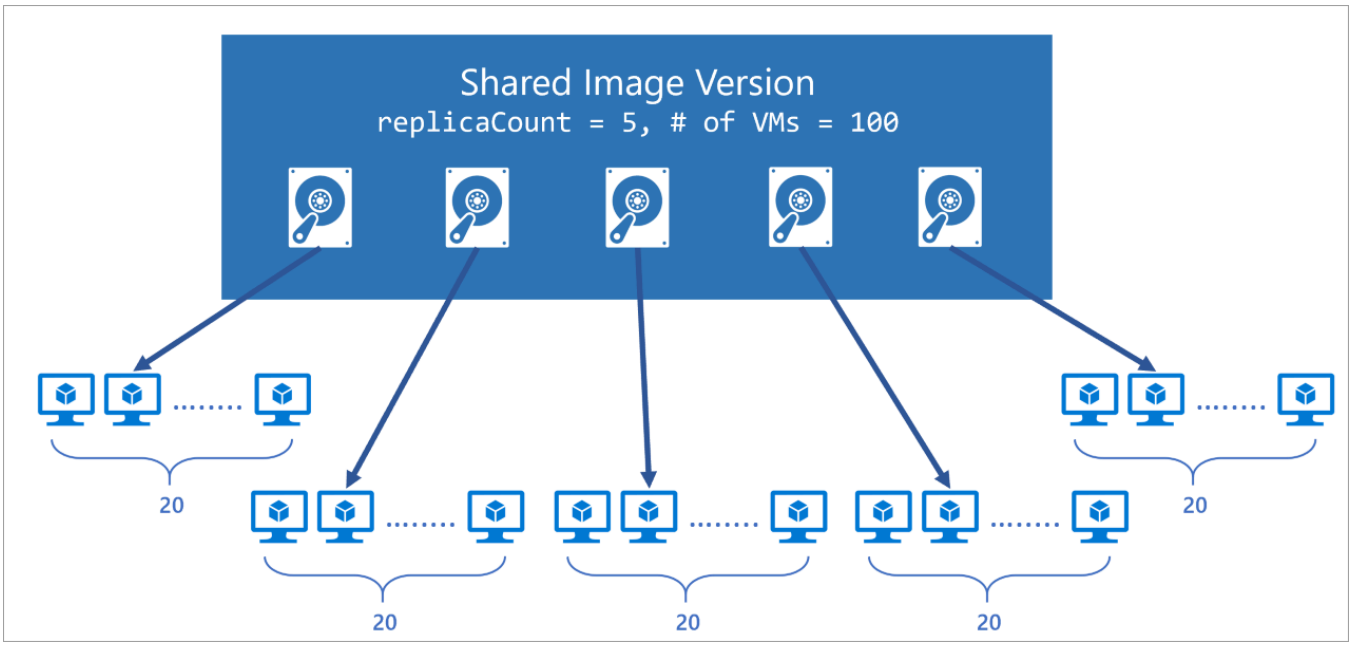
Azure Compute Gallery allows you to specify the number of replicas that you want to keep. In multiple-VM deployment scenarios, you can spread the VM deployments to other replicas. This action reduces the chance that overloading of a single replica throttles the process of instance creation.

With Compute Gallery, you can deploy up to 1,000 VM instances in a scale set. You can set a different replica count in each target region, based on the scale needs for the region. Because each replica is a copy of your resource, this approach helps scale your deployments linearly with each extra replica.

Although we understand that no two resources or regions are the same, here are general guidelines on how to use replicas in a region:

- For every 50 VMs that you create concurrently, we recommend that you keep one replica. For example, if you're creating 500 VMs concurrently by using the same image in a region, we suggest you keep at least 10 replicas of your image.
- For each scale set that you create concurrently, we recommend that you keep one replica.

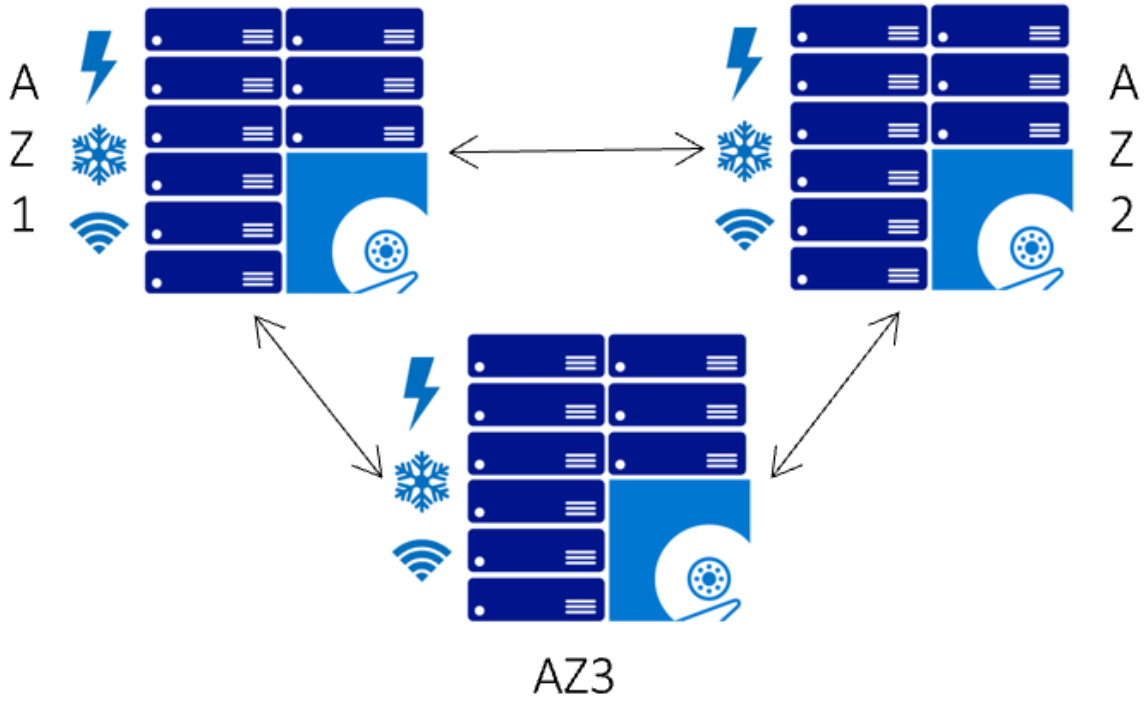
We always recommend that you overprovision the number of replicas due to factors like resource size, content, and OS type.



High availability

Azure ZRS provides resilience against the failure of an availability zone in a region. With the general availability of Azure Compute Gallery, you can choose to store your images in ZRS accounts in regions that have availability zones.

You can also choose the account type for each target region. The default storage account type is Standard LRS, but you can choose Standard ZRS for regions that have availability zones. For more information on the regional availability of ZRS, see Azure Storage redundancy.

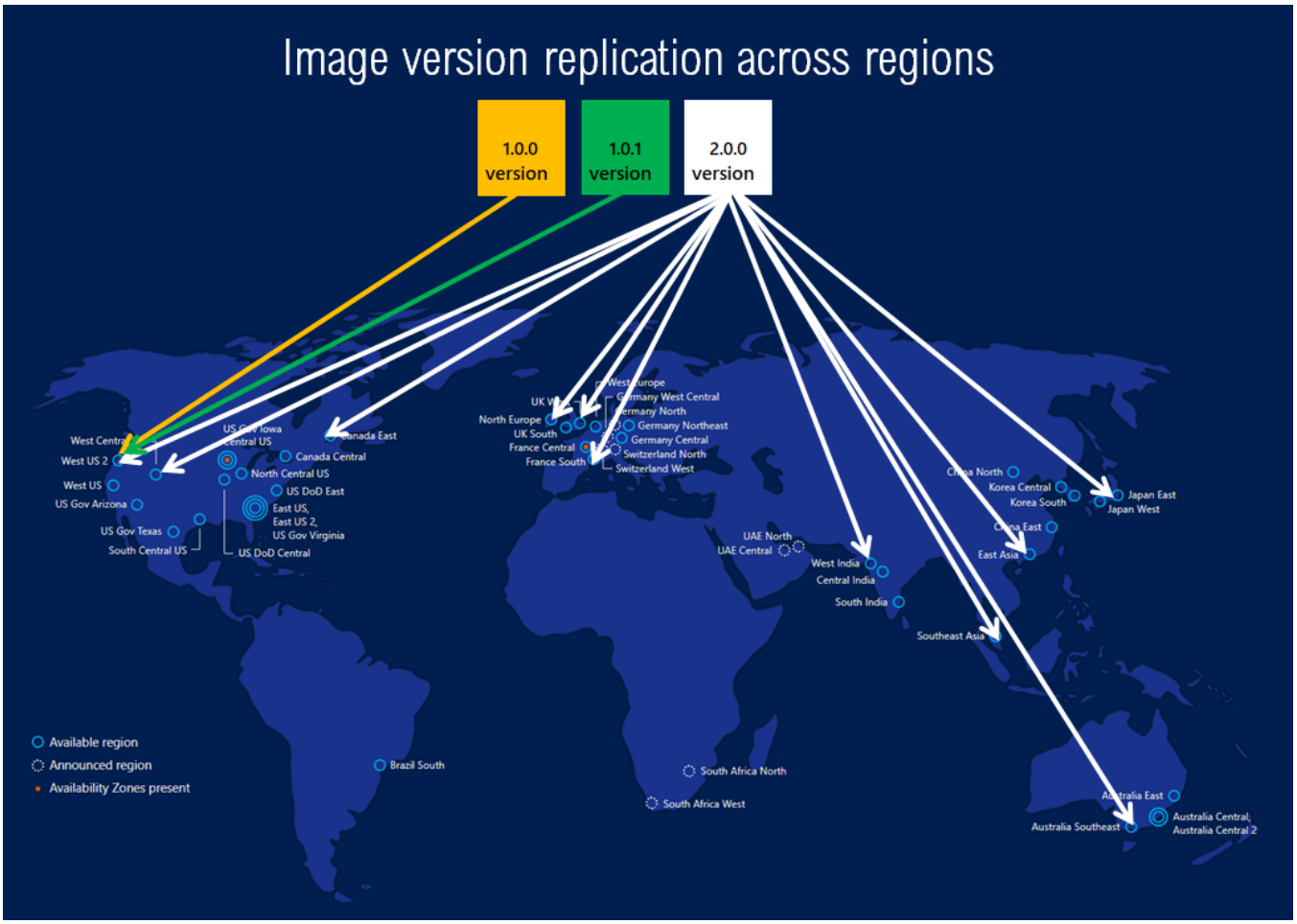


Replication

Azure Compute Gallery also allows you to replicate your resources to other Azure regions automatically. You can replicate each image version to different regions, depending on what makes sense for your organization. One example is to always replicate the latest image in multiple regions while all older image versions are available in only one region. This approach can help you save storage costs.

After creation time, you can update the regions that a resource is replicated to. The time that replication to other regions takes depends on the amount of data that you're copying and the number of regions that the version is replicated to. This process can take a few hours in some cases.

While the replication is happening, you can view its status per region. After the image replication is complete in a region, you can deploy a VM or scale set by using that resource in the region.



Trusted Launch validation for Azure Compute Gallery images (preview)

Trusted Launch validation for Azure Compute Gallery images is currently in preview. This preview is intended for testing, evaluation, and feedback purposes only. We don't recommend it for production workloads.

When you register for the preview, you agree to the supplemental terms of use. Some aspects of this feature might change with general availability.

What's changing

Starting with API 2025-03-03, all new Compute Gallery image definitions default to:

- `hyperVGeneration`: `V2`
- `SecurityType`: `TrustedLaunchSupported`

How Trusted Launch validation works

When you specify `TrustedLaunchSupported` or `TrustedLaunchandConfidentialVMSupported` in the Compute Gallery image definition, the platform automatically validates that the image is Trusted Launch capable. The platform adds the validation result to the image version property. This action ensures that the VM and scale set deployments that use these images can default to Trusted Launch if the validation is successful.

How to enable the preview

To try out Trusted Launch validation for Compute Gallery images, complete the following steps:

1. Register for Trusted Launch as a default feature.
2. Register for the Trusted Launch Validation preview.

After you enable the two features, all new VM and scale set deployments that use Compute Gallery image versions and that are validated successfully for Trusted Launch default to the Trusted Launch security type.

VM deployments from Azure Compute Gallery images

The following sections compare the current and new behaviors of VM deployments from Compute Gallery images, depending on Trusted Launch validation.

Current behavior without Trusted Launch validation

To create a Trusted Launch-supported, Gen2 OS image definition for Compute Gallery, you need to add the following `features` element in your deployment:

JSON

```
"features": [
    {
      "name": "SecurityType",
      "value": "TrustedLaunchSupported"
    }
],
"hyperVGeneration": "V2"
```

New behavior with Trusted Launch validation

`TrustedLaunchSupported` is enabled by default on new Compute Gallery image definitions if the deployment meets any of the following criteria:

- The use of API version 2025-03-03 or later for the `Microsoft.Compute/galleries` resource
- The absence of the `SecurityType` feature
- A `null` value for the `SecurityType` feature

Additionally, the Azure platform triggers validation for the OS image to ensure that it supports Trusted launch capabilities. The validation takes a minimum of one hour. Results are available as the image version property:

JSON

```
"validationsProfile": {
  "executedValidations": [
      {
        "type": "TrustedLaunch",
        "status": "Succeeded",
        "version": "0.0.2",
        "executionTime": "2025-07-10T21:27:33.0113984+00:00"
      }
    ],
}
```

You can choose to explicitly bypass the default for new Compute Gallery image definitions by setting `Standard` as the value for `SecurityType` under `features`:

JSON

```
"features": [
    {
```

```
        "name": "SecurityType",
        "value": "Standard"
      }
  ],
  "hyperVGeneration": "V2"
```

Sharing

There are three main ways to share images in Azure Compute Gallery, depending on which users you want to share with:

Expand table

Sharing with:	People	Groups	Service principal	All users in a specific subscription or tenant	Publicly with all users in Azure
Role-based access control (RBAC) sharing	Yes	Yes	Yes	No	No
RBAC + direct shared gallery	Yes	Yes	Yes	Yes	No
RBAC + community gallery	Yes	Yes	Yes	No	Yes

> **Note**
>
> You can use images with read permissions on them to deploy virtual machines and disks.
>
> When you use a direct shared gallery, images are distributed widely to all users in a subscription or tenant. A community gallery distributes images publicly. When you share images that contain intellectual property, use caution to prevent widespread distribution.

RBAC

Because the gallery, definition, and version are all resources, you can share them by using the built-in Azure RBAC roles. By using Azure RBAC roles, you can share these resources with other users, service principals, and groups. You can even share access with individuals outside the tenant where you created the resources.

After users have access to the resource version, they can use it to deploy a VM or a virtual machine scale set. Here's a sharing matrix that can help you understand what the users get access to:

Expand table

Shared with users	Compute gallery	Image definition	Image version
Compute gallery	Yes	Yes	Yes
Image definition	No	Yes	Yes

We recommend sharing at the gallery level for the best experience. We don't recommend sharing individual image versions. For more information about Azure RBAC, see Assign Azure roles and Share by using RBAC.

Directly sharing with a tenant or subscription

You can give specific subscriptions or tenants access to a direct shared gallery. Sharing a gallery with tenants and subscriptions gives them read-only access to your gallery. For more information, see Share a gallery with all users in a subscription or tenant.

Important considerations

- The feature of direct shared galleries is currently in preview and is subject to the preview terms for Azure Compute Gallery.
- To publish images to a direct shared gallery during the preview, you need to register for the preview. Creating VMs from a direct shared gallery is open to all Azure users.
- During the preview, you need to create a new gallery with the `sharingProfile.permissions` property set to `Groups`. When you use the Azure CLI to create a gallery, use the `--permissions groups` parameter. You can't use an existing gallery, and the property can't currently be updated.
- You can't currently create a flexible virtual machine scale set from an image that another tenant shares with you.

Limitations

During the preview:

- You can share only with subscriptions that are also in the preview.
- You can share up to 30 subscriptions and 5 tenants.
- A direct shared gallery can't contain encrypted image versions.

- Only the owner of a subscription, or a user or service principal assigned to the Compute Gallery Sharing Admin role at the subscription or gallery level, can enable group-based sharing.

Community gallery

To share a gallery with all Azure users, you can create a community gallery. Anyone who has an Azure subscription can use community galleries. Someone who creates a VM can browse through images shared with the community by using the Azure portal, the REST API, or the Azure CLI. Sharing images to the community is a new capability in Compute Gallery.

You can make your image galleries public and share them with Azure customers. When a gallery is marked as a community gallery, all images in it become available to all Azure customers as a new resource type under `Microsoft.Compute/communityGalleries`. Azure customers can see the galleries and use them to create VMs. Your original resources of the type `Microsoft.Compute/galleries` are still under your subscription and private.

For more information, see Share images by using a community gallery.

Activity log

The activity log displays recent activity on the gallery, image, or version. This information includes any configuration changes and when the item was created and deleted.

You can view the activity log in the Azure portal or create a diagnostic setting to send the log to a Log Analytics workspace. In the workspace, you can view events over time or analyze them with other collected data.

The following table lists a few example operations that relate to gallery operations in the activity log. For a complete list of possible log entries, see the Microsoft.Compute Resource Provider options.

Expand table

Operation	Description
`Microsoft.Compute/galleries/write`	Creates a new gallery or updates an existing one
`Microsoft.Compute/galleries/delete`	Deletes the gallery
`Microsoft.Compute/galleries/share/action`	Shares a gallery with different scopes
`Microsoft.Compute/galleries/images/read`	Gets the properties of a gallery image

Operation	Description
`Microsoft.Compute/galleries/images/write`	Creates a new gallery image or updates an existing one
`Microsoft.Compute/galleries/images/versions/read`	Gets the properties of a gallery image version

Billing

There's no extra charge for using the Azure Compute Gallery service. However, you're charged for the following resources:

- Storage costs for each replica. For images, the storage cost is charged as a snapshot. It's based on the occupied size of the image version, the number of replicas of the image version, and the number of regions that the version is replicated to.
- Network egress charges for replication of the first resource version from the source region to the replicated regions. Subsequent replicas are handled within the region, so there are no additional charges.

For example, let's say:

- You have an image of a 127-GB OS disk that occupies only 10 GB of storage, and you have one empty 32-GB data disk. The occupied size of each image would be only 10 GB.
- The image is replicated to three regions, and each region has two replicas. There are six total snapshots, each using 10 GB.

In this example, you're charged for the storage cost of each snapshot, based on the occupied size of 10 GB. You pay network egress charges for the first replica to be copied to the additional two regions.

For more information on the pricing of snapshots in each region, see Managed disks pricing. For more information on network egress, see Bandwidth pricing.

Best practices

- To prevent accidental deletion of images, use resource locks at the gallery level. For more information, see Lock your Azure resources to protect your infrastructure.
- Use ZRS wherever it's available, for high availability. You can configure ZRS on the replication tab when you create a version of the image or VM application. For information about which regions support ZRS, see Azure regions list.

- Keep a minimum of three replicas for production images. For every 20 VMs that you create concurrently, we recommend that you keep one replica.

  For example, if you create 1,000 VMs concurrently, you should keep 50 replicas. (You can have a maximum of 50 replicas per region.) To update the replica count, go to the gallery and select **Image Definition** > **Image Version** > **Update replication**.

- Maintain separate galleries for production and test images. Don't put both image types in a single gallery.

- For disaster recovery scenarios, have at least two galleries in separate regions. You can still use image versions in other regions, but if the region for one gallery goes down, you can't create new gallery resources or update existing ones.

- When you create an image definition, keep the **Publisher**, **Offer**, and **SKU** values consistent with Azure Marketplace images to easily identify OS versions.

  For example, if you're customizing a Windows Server 2019 image from Azure Marketplace and you store it as a Compute Gallery image, use the same **Publisher**, **Offer**, and **SKU** values that you used in the Azure Marketplace image.

- If you want to exclude a specific image version during creation of a VM or scale set, use `excludeFromLatest` when you publish images. See Gallery image versions - create or update.

- If you want to exclude a version in a specific region, use `regionalExcludeFromLatest` instead of the global `excludeFromLatest`. You can set both the global and regional `excludeFromLatest` flag, but the regional flag takes precedence.

```
"publishingProfile": {
  "targetRegions": [
    {
      "name": "brazilsouth",
      "regionalReplicaCount": 1,
      "regionalExcludeFromLatest": false,
      "storageAccountType": "Standard_LRS"
    },
    {
      "name": "canadacentral",
      "regionalReplicaCount": 1,
      "regionalExcludeFromLatest": true,
      "storageAccountType": "Standard_LRS"
    }
  ],
  "replicaCount": 1,
  "excludeFromLatest": true,
```

```
    "storageAccountType": "Standard_LRS"
  }
```

- To prevent accidental deletion of replicated regions and prevent outages, set `safetyProfile.allowDeletionOfReplicatedLocations` to `false` on image versions. You can also set this property by using allow-replicated-location-deletion in the Azure CLI.

```
{
"properties": {
  "publishingProfile": {
    "targetRegions": [
      {
        "name": "West US",
        "regionalReplicaCount": 1,
        "storageAccountType": "Standard_LRS",
        // encryption info
      }
    ],
     "replicaCount": 1,
     "publishedDate": "2018-01-01T00:00:00Z",
     "storageAccountType": "Standard_LRS"
  },
  "storageProfile": {
     "source": {
       "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroup}/providers/Mic
rosoft.Compute/images/{imageName}"
    },
  },
  "safetyProfile": {
     "allowDeletionOfReplicatedLocations" : false
   },
},
"location": "West US",
"name": "1.0.0"
}
```

- To block deletion of the image before its end-of-support date, set `BlockDeletionBeforeEndOfLife`. This feature helps protect against accidental deletion. You set it by using blockdeletionbeforeendoflife in the REST API.

SDK support

The following SDKs support creating galleries:

- .NET

- Java
- Node.js
- Python
- Go

Templates

You can create Azure Compute Gallery resources by using quickstart templates:

- Create a gallery
- Create an image definition in a gallery
- Create an image version in a gallery

Related content

- Learn how to deploy images and VM apps by using Azure Compute Gallery.

What's new for Azure Compute Gallery

08/06/2025

This article lists updates to Azure Compute Gallery features.

July 2025 updates

- Starting with API version 2025-03-03, new Compute Gallery image definitions default to `hyperVGeneration` set to `V2` and `SecurityType` set to `TrustedLaunchSupported`.
- Trusted Launch validation for Azure Compute Gallery images is in preview.
- The ExcludefromLatest property excludes the use of an image version for creating resources to test images and safely rolling out the image to production. You can also set this property at the region level to exclude specific regions.
- The allowDeletionOfReplicatedLocations property indicates whether or not removing the gallery image version from replicated regions is allowed and prevents accidental deletion of regions.
- The BlockDeletionBeforeEndOfLife property enables customers to prevent deletion of images before the end-of-support date. You can opt out by explicitly setting `BlockDeletionBeforeEndOfLife` to `False`.
- Platform metadata, such as publisher, offer, or product information from a virtual machine (VM) created from an Azure Marketplace image, is carried over to the Compute Gallery image.

January 2023 updates

- Direct shared gallery is in preview.
- Best practices information is available.
- Replica count for image versions increased from 50 to 100.

Supported features

- ARM64 images
- Trusted Launch
- Confidential VMs
- Accelerated networking
- Replication mode: full and shallow
  - Image size for shallow replication support is up to 32 TB.
  - Shallow replication is only for test images.

Related content

- For updates and announcements about Azure, see the Microsoft Azure Blog.

Create a gallery for storing and sharing resources

08/06/2025

Azure Compute Gallery (formerly known as Shared Image Gallery) simplifies sharing resources, like images and application packages, across your organization.

Compute Gallery lets you share custom virtual machine (VM) images and application packages with others in your organization, within or across regions, or within a tenant. Choose what resources you want to share, the regions where you want to make them available, and the users you want to share them with. You can create multiple galleries so that you can logically group resources.

The gallery is a top-level resource that can be shared in multiple ways:

Expand table

Sharing with:	People	Groups	Service principal	All users in a specific subscription or tenant	Publicly with all users in Azure
Role-based access control (RBAC) sharing	Yes	Yes	Yes	No	No
RBAC + direct shared gallery	Yes	Yes	Yes	Yes	No
RBAC + community gallery	Yes	Yes	Yes	No	Yes

Requirements for gallery names

Allowed characters for a gallery name are uppercase letters (A-Z), lowercase letters (a-z), digits (0-9), dots (.), and underscores (_). The gallery name can't contain dashes (-). Gallery names must be unique within your subscription.

Create a private gallery

Portal

1. Sign in to the Azure portal.

2. Enter **Azure Compute Gallery** in the search box, and select **Azure Compute Gallery** in the results.
3. On the **Azure Compute Gallery** page, select **Add**.
4. On the **Create Azure compute gallery** pane, select the correct subscription.
5. For **Resource group**, select a resource group from the dropdown list. Or, select **Create new** and enter a name for the new resource group.
6. For **Name**, enter a name for the gallery.
7. For **Region**, select a region from the dropdown list.
8. You can enter a short description of the gallery, like **My gallery for testing**. Then select **Review + create**.
9. After validation passes, select **Create**.
10. When the deployment finishes, select **Go to resource**.

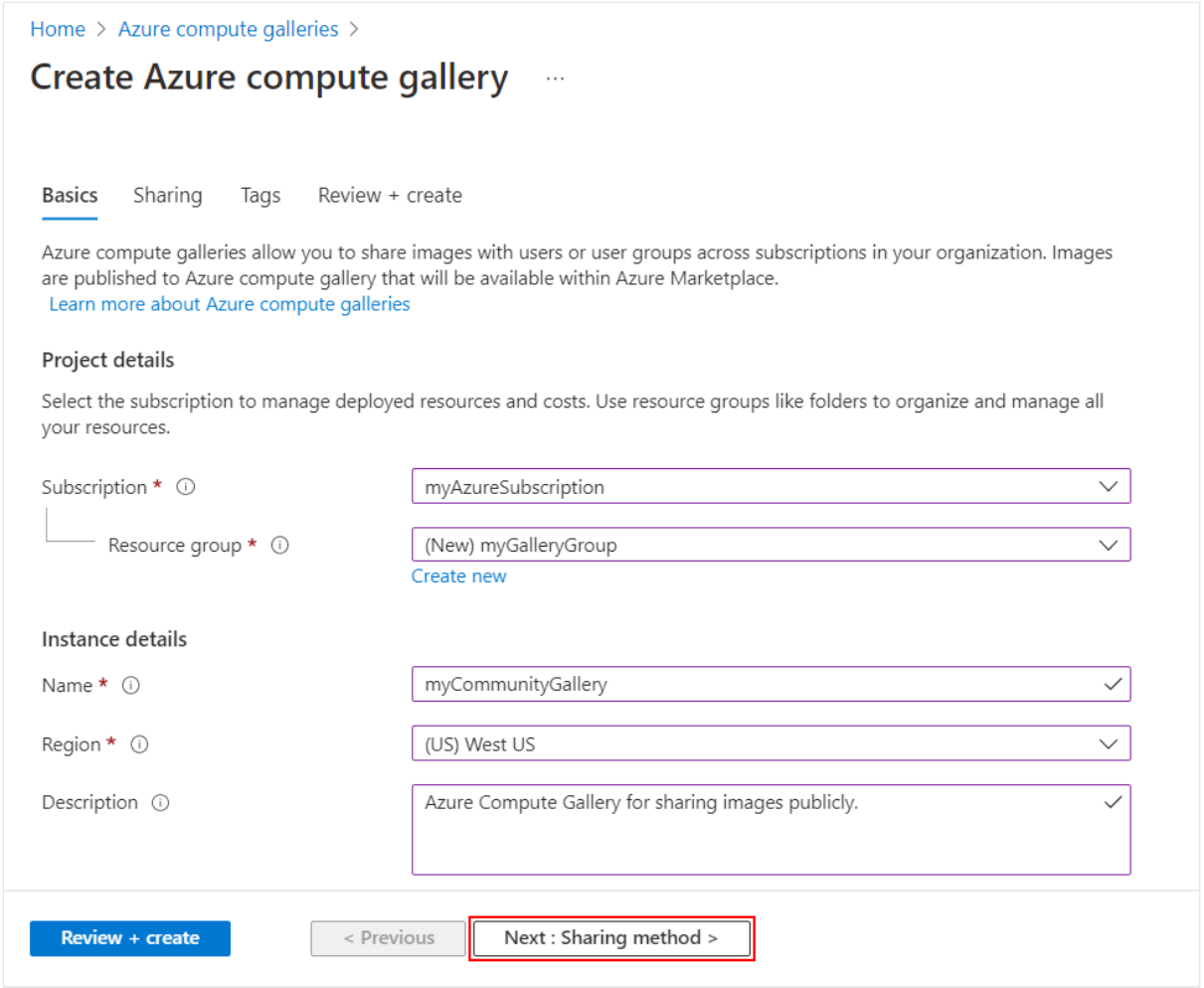
Create a direct shared gallery

> **Important**
>
> The *direct shared gallery* feature is currently in preview and is subject to the preview terms for Azure Compute Gallery.
>
> During the preview, you need to create a new gallery with the `sharingProfile.permissions` property set to `Groups`. When you use the Azure CLI to create a gallery, use the `--permissions groups` parameter. You can't use an existing gallery, and the property can't currently be updated.
>
> You can't currently create a flexible virtual machine scale set from an image that another tenant shared with you.

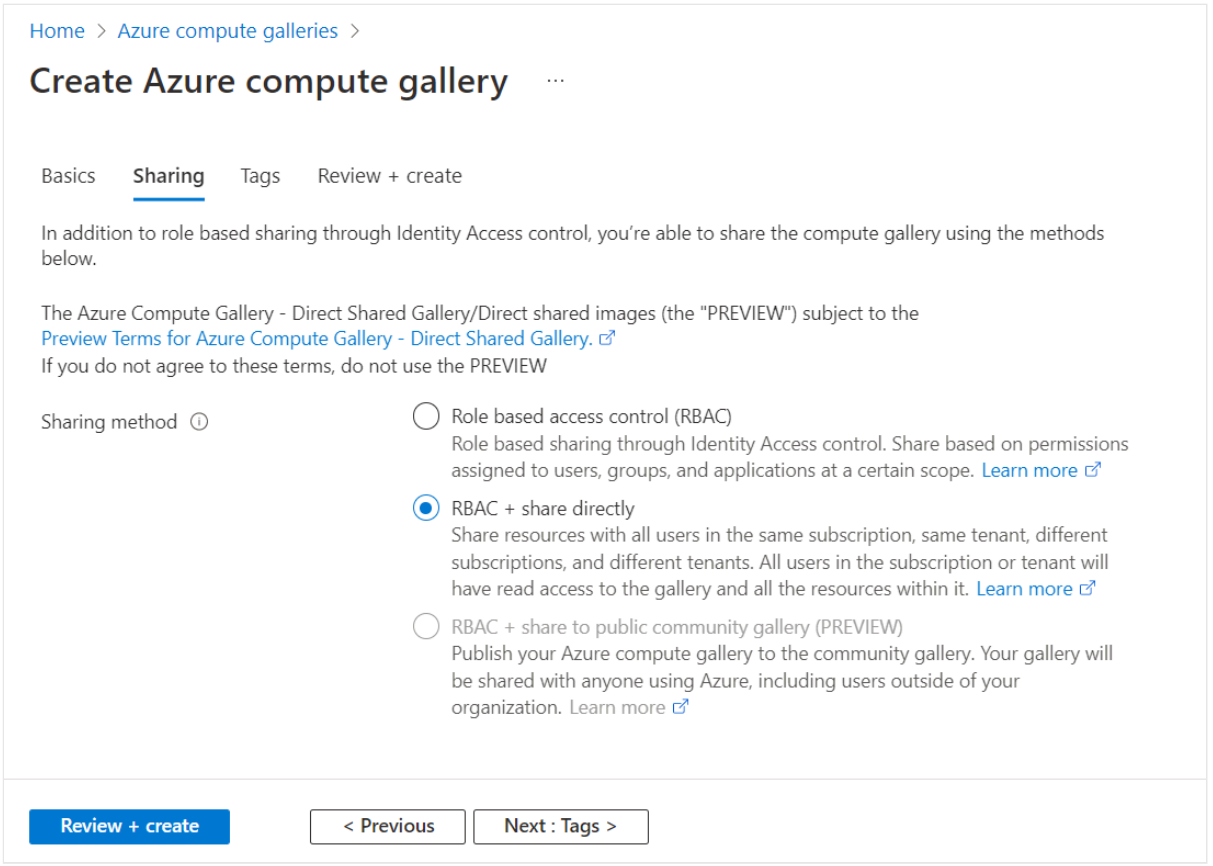
To start sharing a direct shared gallery with a subscription or tenant, see Share a gallery with a subscription or tenant.

Portal

1. Sign in to the Azure portal.
2. Enter **Azure Compute Gallery** in the search box, and select **Azure Compute Gallery** in the results.
3. On the **Azure Compute Gallery** page, select **Add**.
4. On the **Create Azure compute gallery** pane, on the **Basics** tab, select the correct subscription.
5. Complete all of the details on the **Basics** tab.
6. At the bottom of the pane, select **Next: Sharing method**.



7. On the **Sharing** tab, select **RBAC + share directly**.



8. When you finish, select **Review + create**.
9. After validation passes, select **Create**.
10. When the deployment finishes, select **Go to resource**.

To start sharing the gallery with a subscription or tenant, see Share a gallery with a subscription or tenant.

Create a community gallery

A community gallery is shared publicly with everyone. To create a community gallery, you create the gallery first and then enable it for sharing. The name of public instance of your gallery is the prefix that you provide, plus a GUID. To share your gallery publicly, be sure to create your gallery, image definitions, and image versions in the same region.

When you create an image to share with the community, you need to provide contact information. This information is *publicly available*, so be careful when you provide:

- Community gallery prefix
- Publisher support email
- Publisher URL
- Legal agreement URL

Information from your image definitions is also publicly available, like what you provide for **Publisher**, **Offer**, and **SKU**.

Only the following people can enable a gallery to go public to the community:

- The owner of a subscription
- A user or a service principal assigned to the Compute Gallery Sharing Admin role at the subscription or gallery level

To assign a role to a user, group, service principal, or managed identity, see Steps to assign an Azure role.

CLI

Use the `--public-name-prefix` value to create a name for the public version of your gallery. The `--public-name-prefix` value is the first part of the public name. The last part is a GUID, created by the platform, that's unique to your gallery.

Azure CLI

```
location=westus
galleryName=contosoGallery
resourceGroup=myCGRG
publisherUri=https://www.contoso.com
publisherEmail=support@contoso.com
eulaLink=https://www.contoso.com/eula
prefix=ContosoImages

az group create --name $resourceGroup --location $location

az sig create \
   --gallery-name $galleryName \
   --permissions community \
   --resource-group $resourceGroup \
   --publisher-uri $publisherUri \
   --publisher-email $publisherEmail \
   --eula $eulaLink \
   --public-name-prefix $prefix
```

The output of this command gives you the public name for your community gallery in the `sharingProfile` section, under `publicNames`.

To start sharing the gallery to all Azure users, see Share images by using a community gallery.

Related content

- Create an image definition and an image version.
- Create a VM from a generalized or specialized image in a gallery.
- Create a VM application in your gallery.

Soft Delete in Azure Compute Gallery (Preview)

09/30/2025

> **Important**
>
> The Soft Delete feature is currently in PREVIEW. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

The Soft Delete feature in Azure Compute Gallery (ACG) lets you recover accidentally deleted resources within a 7 day retention period. After this timeframe, the platform permanently deletes the resources. Currently, this feature is only available for ACG images.

Soft Delete is available for the following gallery types:

- Private
- Direct Shared Gallery
- Community Gallery

When a resource, such as an image, is deleted using Soft Delete, it isn't immediately removed from the system. Instead, it enters a *soft-deleted* state, during which it remains recoverable for up to seven days. This grace period gives you time to review and restore any resources that were mistakenly deleted, preventing permanent loss.

After the retention window expires, the deleted items are automatically purged and can't be recovered. Soft Delete is useful in environments where accidental deletions can disrupt workflows or cause data loss.

Prerequisites

- Register the Azure Compute Gallery Soft Delete feature in the Azure portal.
- Alternatively, register Azure Compute Gallery Soft Delete feature by running the following `az feature register` command from CLI:

```
az feature register --name SIGSoftDelete --namespace Microsoft.Compute
```

- The minimum API version required to use Soft Delete is `2024-03-03`.

Limitations

- Soft Delete is only supported in the Azure portal or Rest API with some limitations.
- You can't currently update the retention policy.
- Virtual Machine Apps Recovery isn't supported.
- Deleting a gallery or image definition with Soft Deleted resources isn't supported.
- Only the following users are allowed to enable Soft Delete: the owner of a subscription, user or service principal assigned to the Compute Gallery Sharing Admin role at the subscription or gallery level
- Not supported in National Clouds.
- You can't enable Soft Delete if the gallery or image definition and image version are in different locations.
- Virtual Machine Scale Sets in Flexible orchestration aren't compatible with the Soft Delete feature.
- When an image in a gallery with Soft Delete enabled has a failed provisioning state, it can't be deleted. The recommended workaround is to temporarily disable Soft Delete on the gallery, delete the image, and then re-enable Soft Delete.

Billing

Soft Delete is provided at no cost. After an image is soft deleted, only one replica per region is retained. You are charged for that single replica until the image is permanently deleted.

Enable Soft Delete on a gallery

You can enable Soft Delete on a new or existing gallery.

REST

To enable Soft Delete on an existing gallery, send the following `PUT` request. As part of the request, include and set the `Softdeletepolicy` and `isSoftDeleteEnabled` values to `true`.

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{rg
Name}/providers/Microsoft.Compute/galleries/{gallery-name}?api-version=2024-
03-03
{
"properties": {
"softDeletePolicy": {
"isSoftDeleteEnabled": true
}
```

```
},
"location": "{location}
}
```

Soft delete an image

Once Soft Delete is enabled on a gallery, all images in the gallery are soft deleted.

REST

To soft delete an image, send the following `DELETE` request on the resource you intend to delete.

rest

```
DELETE
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{rg
Name}/providers/Microsoft.Compute/galleries/{gallery-name}/images/{image-
defination-name}/versions/{version-name}?api-version=2024-03-03
```

List all soft deleted images

View a list of soft deleted images in a gallery or image definition.

REST

Send the following `GET` request on the resource of the image definition to list all the soft deleted images.

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{rg
Name}/providers/Microsoft.Compute/galleries/{gallery-
name}/softDeletedArtifactTypes/Images/artifacts/{image-defination-
name}/versions?api-version=2024-03-03
```

Recover a soft deleted image

In this example, we demonstrate how to recover a soft deleted image.

REST

To recover soft deleted images in a gallery, send the following `PUT` request on the image version to be recovered along with the home region of the image.

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{rg
Name}/providers/Microsoft.Compute/galleries/{gallery-name}/images/{image-
defination-name}/versions/{version-name}?api-version=2024-03-03
{
    "location": "eastus2euap"
}
```

Hard Delete an image

Hard Delete permanently deletes the image without the possibility of a recovery.

REST

To hard delete an image in a gallery, send the following `DELETE` request on the image version.

rest

```
DELETE
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{rg
Name}/providers/Microsoft.Compute/galleries/{gallery-
name}/softDeletedArtifactTypes/Images/artifacts/{image-defination-
name}/versions/{version-name}?api-version=2024-03-03
```

Frequently Asked Questions

Does Soft Delete become effective immediately upon being enabled in the gallery?

Yes, as long as the soft delete operation completes successfully.

Can I enable Soft Delete on an existing gallery and instantly delete and recover images, or is there a delay?

Yes, as long as the soft delete operation completes successfully.

Can I update the retention period beyond seven days?

No, currently you can't update the retention period.

Can I delete a gallery or image definition that has soft deleted images?

No, you can either delete all the soft deleted images or disable soft delete on the gallery to delete the image definition.

Share gallery resources across subscriptions and tenants by using RBAC

08/06/2025

Because a gallery, definition, and version are all resources in Azure Compute Gallery, you can share them by using the built-in Azure role-based access control (RBAC) roles. By using Azure RBAC roles, you can share these resources with other users, service principals, and groups. You can even share access with individuals outside the tenant where they were created.

After users have access, they can use the gallery resources to deploy a virtual machine (VM) or a virtual machine scale set. Here's a sharing matrix that can help you understand what the users get access to:

Expand table

Shared with users	Gallery	Image definition	Image version
Gallery	Yes	Yes	Yes
Image definition	No	Yes	Yes

We recommend sharing at the gallery level for the best experience. We don't recommend sharing individual image versions. For more information about Azure RBAC, see Assign Azure roles.

There are three main ways to share images in Compute Gallery, depending on which users you want to share with:

Expand table

Sharing with:	People	Groups	Service principal	All users in a specific subscription or tenant	Publicly with all users in Azure
RBAC sharing	Yes	Yes	Yes	No	No
RBAC + direct shared gallery	Yes	Yes	Yes	Yes	No
RBAC + community gallery	Yes	Yes	Yes	No	Yes

You can also create an app registration to share images between tenants.

> **Note**
>
> You can use images with read permissions on them to deploy virtual machines and disks.
>
> When you use a direct shared gallery, images are distributed widely to all users in a subscription or tenant. A community gallery distributes images publicly. When you share images that contain intellectual property, use caution to prevent widespread distribution.

Share by using RBAC

When you share a gallery by using RBAC, you need to provide the `imageID` value to anyone who creates a VM or scale set from the image. The person who's deploying the VM or scale set can't list the images that were shared with them via RBAC.

If you share gallery resources with someone outside your Azure tenant, they need your `tenantID` value to sign in and have Azure verify that they have access to the resource before they can use it within their own tenant. You need to provide the `tenantID` value. There is no way for someone outside your organization to query for this value.

For instructions on consuming an image shared via RBAC and creating a VM or a scale set, see:

- RBAC - Shared within your organization
- RBAC - Shared from another tenant

Portal

1. Sign in to the Azure portal.
2. On the page for your gallery, on the left menu, select **Access control (IAM)**.
3. Under **Add**, select **Add role assignment**. The **Add role assignment** pane opens.
4. Under **Role**, select **Reader**.
5. On the **Members** tab, ensure that the user is selected. For **Assign access to**, keep the default of **User, group, or service principal**.
6. Choose **Select members**. On the pane that opens, choose a user account.

7. If the user is outside your organization, the following message appears: **This user will be sent an email that enables them to collaborate with Microsoft**. Select the user with the email address, and then select **Save**.

Related content

- Create an image definition and an image version.
- Create a VM from a generalized or specialized image in a gallery.

Share gallery VM images across Azure tenants by using an app registration

08/06/2025

With Azure Compute Gallery, you can share a virtual machine (VM) image with another organization by using an app registration. For information about other sharing options, see Share gallery resources across subscriptions and tenants by using RBAC.

But, if you want to share images outside of your Azure tenant, at scale, you should create an app registration. Using an app registration can enable more complex sharing scenarios, like:

- Managing shared images when one company acquires another, and the Azure infrastructure is spread across separate tenants.
- Azure Partners manage Azure infrastructure on behalf of their customers. Customization of images is done within the partners tenant, but the infrastructure deployments will happen in the customer's tenant.

Create the app registration

Create an application registration that will be used by both tenants to share the image gallery resources.

1. Open the App registrations in the Azure portal.
2. Select **New registration** from the menu at the top of the page.
3. In **Name**, type *myGalleryApp*.
4. In **Supported account types**, select **Accounts in any organizational directory (Any Microsoft Entra directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)**.
5. In **Redirect URI**, select *Web* from the **Select a platform** dropdown and type *https://www.microsoft.com*, then select **Register**. After the app registration has been created, the overview page will open.
6. On the overview page, copy the **Application (client) ID** and save for use later.
7. Select **Certificates & secrets**, and then select **New client secret**.
8. In **Description**, type *Gallery cross-tenant app secret*.
9. In **Expires**, change from the default of **6 months (recommended)** to **12 months** and then select **Add**.
10. Copy the value of the secret and save it to a safe place. You cannot retrieve it after you leave the page.

Give the app registration permission to use the gallery.

1. In the Azure portal, select the Azure Compute Gallery that you want to share with another tenant.
2. Select **select Access control (IAM)**, and under **Add role assignment** select *Add*.
3. Under **Role**, select **Reader**.
4. Under **Assign access to:**, leave this as **Microsoft Entra user, group, or service principal**.
5. Under **Select members**, type *myGalleryApp* and select it when it shows up in the list. When you are done, select **Review + assign**.

Give Tenant 2 access

Give Tenant 2 access to the application by requesting a sign-in using a browser. Replace *<Tenant2 ID>* with the tenant ID for the tenant that you would like to share your image gallery with. Users can see their tenant ID using the Azure CLI command `az account show`.

Replace *<Application (client) ID>* with the application ID of the app registration you created. When done making the replacements, paste the URL into a browser and follow the sign-in prompts to sign into Tenant 2.

```
https://login.microsoftonline.com/<Tenant 2 ID>/oauth2/authorize?client_id=
<Application (client)
ID>&response_type=code&redirect_uri=https%3A%2F%2Fwww.microsoft.com%2F
```

In the Azure portal sign in as Tenant 2 and give the app registration access to the resource group where you want to create the VM.

1. Select the resource group and then select **Access control (IAM)**. Under **Add role assignment** select **Add**.
2. Under **Role**, type **Contributor**.
3. Under **Assign access to:**, leave this as **Microsoft Entra user, group, or service principal**.
4. Under **Select members** type *myGalleryApp* then select it when it shows up in the list. When you are done, select **Review + assign**.

> **Note**
>
> You need to wait for the image version to completely finish being built and replicated before you can use the same managed image to create another image version.

> **Important**

You can't use the portal to deploy a VM from an image in another Azure tenant. To create a VM from an image shared between tenants, you must use the Azure CLI or Azure PowerShell.

Create the VM

You need the following information before you create a VM from an image shared with you by using an app registration:

- The tenant IDs from both the source subscription and the subscription where you want to create the VM
- The client ID of the app registration and the secret
- The image ID of the image that you want to use

CLI

Sign in the service principal for Tenant 1 by using the app ID, the app key, and the ID of Tenant 1. You can use `az account show --query "tenantId"` to get the tenant ID, if necessary.

The following example shows how to create a VM from a generalized image. If you're using a specialized image, see Create a VM using a specialized image version.

Azure CLI

```
tenant1='<ID for tenant 1>'
tenant2='<ID for tenant 2>'
appid='<client ID of the app registration>'
secret='<secret from the app registration>'

az account clear
az login --service-principal -u $appid -p $secret --tenant $tenant1
az account get-access-token
```

Sign in the service principal for Tenant 2 by using the app ID, the app key, and the ID of Tenant 2:

Azure CLI

```
az login --service-principal -u $appid -p $secret --tenant $tenant2
az account get-access-token
```

Create the VM. Replace the information in the following example with your own.

Azure CLI

```
imageid="<ID of the image that you want to use>"
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --image $imageid \
  --admin-username azureuser \
  --generate-ssh-keys
```

Related content

- If you have any problems, you can troubleshoot galleries.

Share a gallery with all users in a subscription or tenant (preview)

08/06/2025

This article covers how to share a gallery with specific subscriptions or tenants by using a direct shared gallery in Azure Compute Gallery. Sharing a gallery with tenants and subscriptions gives them read-only access to it.

There are three main ways to share images in Compute Gallery, depending on which users you want to share with:

Expand table

Sharing with:	People	Groups	Service principal	All users in a specific subscription or tenant	Publicly with all users in Azure
Role-based access control (RBAC) sharing	Yes	Yes	Yes	No	No
RBAC + direct shared gallery	Yes	Yes	Yes	Yes	No
RBAC + community gallery	Yes	Yes	Yes	No	Yes

Important considerations

- The feature of direct shared galleries is currently in preview and is subject to the preview terms for Azure Compute Gallery.
- To publish images to a direct shared gallery during the preview, you need to register for the preview. No additional access is required to consume images.
- Creating virtual machines (VMs) from a direct shared gallery is open to all Azure users in the target subscription or tenant that the gallery is shared with.
- Access to the feature of direct shared galleries is gated because in most scenarios, RBAC or app sharing is sufficient for same-subscription and cross-tenant scenarios. Request access to the feature only if you want to share images widely with all users in the subscription or tenant, and if your business case requires access to direct shared galleries.
- You can use images with read permissions on them to deploy VMs and disks.

- When you use a direct shared gallery, images are distributed widely to all users in a subscription or tenant. A community gallery distributes images publicly. When you share images that contain intellectual property, use caution to prevent widespread distribution.

Limitations

During the preview:

- You can share up to 30 subscriptions and 5 tenants.
- Only images can be shared. You can't directly share a VM application during the preview.
- A direct shared gallery can't contain encrypted image versions.
- Only the owner of a subscription, or a user or service principal assigned to the Compute Gallery Sharing Admin role at the subscription or gallery level, can enable group-based sharing.
- Only REST API, Azure CLI, and Azure portal support is available in the feature preview. PowerShell and Terraform support is currently not available.
- Although portal support is available for this feature, consumption of images in the portal is available only from the pane for creating a VM or virtual machine scale set. There's no way to browse through direct shared images in the portal.
- The image version's region in the gallery should be the same as the home region. The creation of cross-region versions where the home region is different from the gallery is not supported. However, after the image is in the home region, it can be replicated to other regions.
- The feature is not available in government clouds.
- *Known issue*: When you create a VM from a direct shared image by using the Azure portal, if you select a region and an image and then change the region, you get an error message: "You can only create VM in the replication regions of this image." The message appears even when the image is replicated to that region.

  To eliminate the error, select a different region, and then switch back to the region that you want. If the image is available, it should clear the error message.

Prerequisites

You need to create a new direct shared gallery.

During the preview, a direct shared gallery has the `sharingProfile.permissions` property set to `Groups`. When you use the Azure CLI to create a gallery, use the `--permissions groups` parameter. You can't use an existing gallery, and the property can't currently be updated.

Share a gallery

Here's an overview of the sharing process:

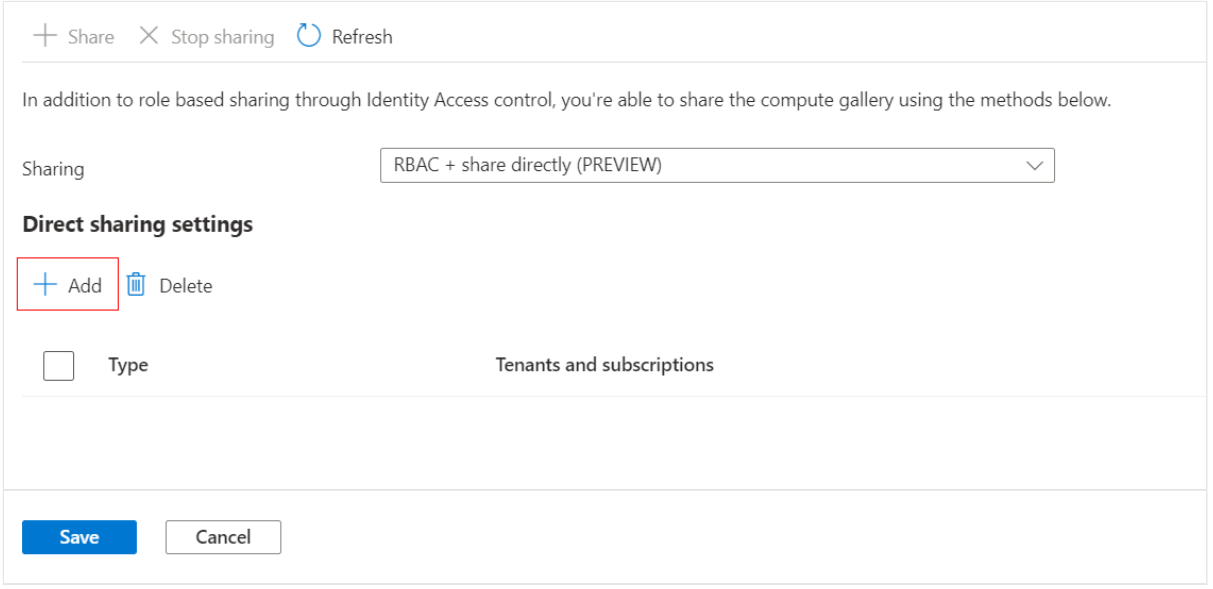
1. You create a gallery under `Microsoft.Compute/Galleries` and choose `groups` as a sharing option.
2. When you're ready, you share your gallery with subscriptions and tenants.
3. The Azure infrastructure creates proxy read-only regional resources under `Microsoft.Compute/SharedGalleries`.

Only subscriptions and tenants that you've shared your gallery with can interact with the proxy resources. They never interact with your private resources.

As the publisher of the private resource, you should consider the private resource as your handle to the public proxy resources. The subscriptions and tenants that you've shared your gallery with see the gallery name as the subscription ID where the gallery was created, followed by the gallery name.

Portal

1. On the page for the gallery, on the left menu, select **Sharing**.
2. Under **Direct sharing settings**, select **Add**.



3. If you want to share with someone within your organization, for **Type**, select **Subscription** or **Tenant**. Then choose the appropriate item in the **Tenants and subscriptions** dropdown list.

   If you want to share with someone outside your organization, select either **Subscription outside of my organization** or **Tenant outside of my organization**. Then paste or enter the ID in the text box.

   When a gallery is shared with the tenant, all subscriptions within the tenant get access to the image. They don't have to share it with individual subscriptions in the tenant.

4. When you finish adding items, select **Save**.

> ⓘ **Note**
>
> If you get the error "Failed to update Azure compute gallery," make sure that you have owner or Compute Gallery Sharing Admin permission on the gallery.

Related content

- Create an image definition and an image version.
- Create a VM from a generalized or specialized image from a direct shared image in the target subscription or tenant.

Share images by using a community gallery

08/06/2025

To share a gallery with all Azure users, you can create a community gallery. Anyone who has an Azure subscription can use community galleries. Someone who creates a virtual machine (VM) can browse through images shared with the community by using the Azure portal, the REST API, or the Azure CLI. Sharing images with the community is a new capability in Azure Compute Gallery.

You can make your image galleries public and share them with Azure customers. When a gallery is marked as a community gallery, all images in it become available to all Azure customers as a new resource type under `Microsoft.Compute/communityGalleries`. Azure customers can see the galleries and use them to create VMs. Your original resources of the type `Microsoft.Compute/galleries` are still under your subscription and private.

> **Important**
>
> Microsoft does not provide support for community images. For any image-related issues, contact the image publishers. However, Microsoft does provide commercially reasonable support to isolate any platform issue that involves a community image. You can find the publisher contact by querying the image.

There are three main ways to share images in Compute Gallery, depending on which users you want to share with:

Expand table

Sharing with:	People	Groups	Service principal	All users in a specific subscription or tenant	Publicly with all users in Azure
Role-based access control (RBAC) sharing	Yes	Yes	Yes	No	No
RBAC + direct shared gallery	Yes	Yes	Yes	Yes	No
RBAC + community gallery	Yes	Yes	Yes	No	Yes

> **Note**
>
> You can use images with read permissions on them to deploy virtual machines and disks.

When you use a direct shared gallery, images are distributed widely to all users in a subscription or tenant. A community gallery distributes images publicly. When you share images that contain intellectual property, use caution to prevent widespread distribution.

Disclaimer

Microsoft doesn't verify or test community images and associated publisher information. You're solely responsible for any community image that you deploy or use. You're responsible for your dealings with the publishers of images. For approved operating system base images, see the list in GitHub. For other images created by verified publishers, see Azure Marketplace.

Limitations for sharing images with the community

- You can't convert an existing private gallery (RBAC-enabled gallery) to a community gallery.
- You can't use a partner image from Azure Marketplace and publish it to the community.
- Encrypted images are not supported.
- The feature isn't available in government clouds.
- Image resources need to be created in the same region as the gallery. For example, if you create a gallery in West US, the image definitions and image versions should be created in West US if you want to make them available.
- You currently can't share VM applications with the community.

How sharing with the community works

You create a gallery resource under `Microsoft.Compute/Galleries` and choose `community` as a sharing option.

When you're ready, you flag your gallery as ready to be shared publicly. Only the owner of a subscription, or a user or service principal with the Compute Gallery Sharing Admin role at the subscription or gallery level, can enable a gallery to go public to the community. At this point, the Azure infrastructure creates proxy read-only regional resources under `Microsoft.Compute/CommunityGalleries`. These proxy resources are public.

Users can interact only with the proxy resources. They never interact with your private resources. As the publisher of a private resource, you should consider the private resource as your handle to the public proxy resources. The `prefix` value that you provide when you create the gallery is used, along with a GUID, to create the public-facing name for your gallery.

Azure users can see the latest image versions shared with the community in the portal, or query for them by using the Azure CLI. Only the latest version of an image is listed in the community gallery.

When you create a community gallery, you need to provide contact information for your images. This information helps facilitate communication between the consumer of the image and the publisher if the consumer needs assistance. Microsoft doesn't offer support for these images. This information is *publicly available*, so be careful when you provide it:

- Community gallery prefix.
- Publisher support email.
- Publisher URL.
- Legal agreement URL. Do not put secrets, passwords, share access signature (SAS) URIs, or any other sensitive information in this URL.

Information from your image definitions is also publicly available, including what you provide for **Publisher**, **Offer**, and **SKU**.

> ⚠ **Warning**
>
> If you want to stop sharing a gallery publicly, you can update the gallery to stop sharing. However, making the gallery private will prevent existing users of virtual machine scale sets from scaling their resources.

How an image publisher chooses between publishing to Azure Marketplace and a community gallery

When to publish to Azure Marketplace:

- You signed Azure Marketplace terms.
- You want to publish commercial images.
- You want to publish a stable version or major release.

When to publish to a community gallery:

- You can't sign Azure Marketplace terms and still want to share images publicly on Azure.
- You want to publish non-commercial images.
- You want to publish daily or nightly builds.

How an image consumer chooses between Azure Marketplace and community images

A consumer might choose Azure Marketplace images when that consumer needs:

- Microsoft-certified images.
- Images for production workloads.
- Microsoft and partner images.
- Paid images with additional software offerings.
- Images that Microsoft supports.

A consumer might choose community images when that consumer needs:

- A community version of an image published by an open-source community.
- Images from a trusted publisher with known contact information.
- Images for testing.
- Free images.
- Images supported by the image owners, not Microsoft.

Reporting issues with a community image

Using community-submitted VM images has several risks. Images could contain malware, security vulnerabilities, or intellectual property. To help create a secure and reliable experience for the community, you can report images when you see these issues.

The easiest way to report issues with a community gallery is to use the portal, which pre-fills information for the report:

- For issues with links or other information in the fields of an image definition, select **Report community image**.
- For malicious code or other issues in a specific image version, select **Report** under the **Report version** column in the table of image versions.

You can also use the following links to report issues, but the forms aren't pre-filled:

- For malicious images, report abuse.
- For violations of intellectual property, submit an infringement report.

Best practices

- Images published to the community gallery should be generalized and have no sensitive or machine-specific details. To learn more about preparing an image, see the OS-specific

information for Linux or Windows.

- If you want to block sharing images with the community at the organization level, create an Azure policy with the following policy rule:

```
  "policyRule": {
      "if": {
        "allOf": [
          {
            "field": "type",
            "equals": "Microsoft.Compute/galleries"
          },
          {
            "field":
"Microsoft.Compute/galleries/sharingProfile.permissions",
            "equals": "Community"
          }
        ]
      },
      "then": {
        "effect": "[parameters('effect')]"
      }
    }
```

FAQ

What are the charges for using a gallery that's shared with the community?

There are no charges for using the service itself. However, content publishers do receive:

- Storage charges for application versions and replicas in each of the regions (source and target). These charges are based on the chosen type of storage account.
- Network egress charges for replication across regions.

Consumers of the image might have to pay additional software costs if the base image is using an Azure Marketplace image with software charges.

Is it safe to use images shared with the community?

Users should exercise caution while using images from unverified sources. These images aren't subject to certification, and they're not scanned for malware or vulnerabilities.

If an image that's shared with the community doesn't work, who provides support?

Azure isn't responsible for any issues that users might encounter with community-shared images. The image publisher provides the support. If you need support, use the image publisher's contact information.

Is community gallery sharing part of Azure Marketplace?

No, community gallery sharing isn't part of Azure Marketplace. It's a feature of Azure Compute Gallery. Anyone who has an Azure subscription can use a community gallery to make their images public.

How do I request that an image shared with the community is replicated to a specific region?

Only the content publishers have control over the regions where their images are available. If you don't find an image in a specific region, contact the publisher directly.

Start sharing publicly

For a gallery to be shared publicly, it needs to be created as a community gallery. For more information, see Create a community gallery.

CLI

When you're ready to make the gallery available to the community, use the az sig share enable-community command. Only a user in the Owner role definition can enable a gallery for community sharing.

Azure CLI

```
az sig share enable-community \
   --gallery-name $galleryName \
   --resource-group $resourceGroup
```

To go back to only RBAC-based sharing, use the az sig share reset command.

To delete a gallery that's shared with the community, you must first run `az sig share reset` to stop sharing. Then you can delete the gallery.

> **Important**
>
> If you're listed as the owner of your subscription, but you're having trouble sharing the gallery publicly, you might need to explicitly add yourself as owner again.

To go back to only RBAC-based sharing, use the az sig share reset command.

To delete a gallery that's shared with the community, you must first run `az sig share reset` to stop sharing. Then you can delete the gallery.

Related content

- Create an image definition and an image version.
- Create a VM from a generalized or specialized image in a community gallery.

Store and share images in an Azure Compute Gallery

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

An image is a copy of either a full VM (including any attached data disks) or just the OS disk, depending on how it's created. When you create a VM from the image, a copy of the VHDs in the image are used to create the disks for the new VM. The image remains in storage and can be used over and over again to create new VMs.

If you have a large number of images that you need to maintain, and would like to make them available throughout your company, you can use an Azure Compute Gallery as a repository.

When you use a gallery to store images, multiple resource types are created:

Expand table

Resource	Description
Image source	This is a resource that can be used to create an **image version** in a gallery. An image source can be an existing Azure VM that is either generalized or specialized, a managed image, a snapshot, a VHD or an image version in another gallery.
Gallery	Like the Azure Marketplace, a **gallery** is a repository for managing and sharing images and other resources, but you control who has access.
Image definition	Image definitions are created within a gallery and they carry information about the image and any requirements for using it to create VMs. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements. It's a definition of a type of image.
Image version	An **image version** is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an **image version** to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times.

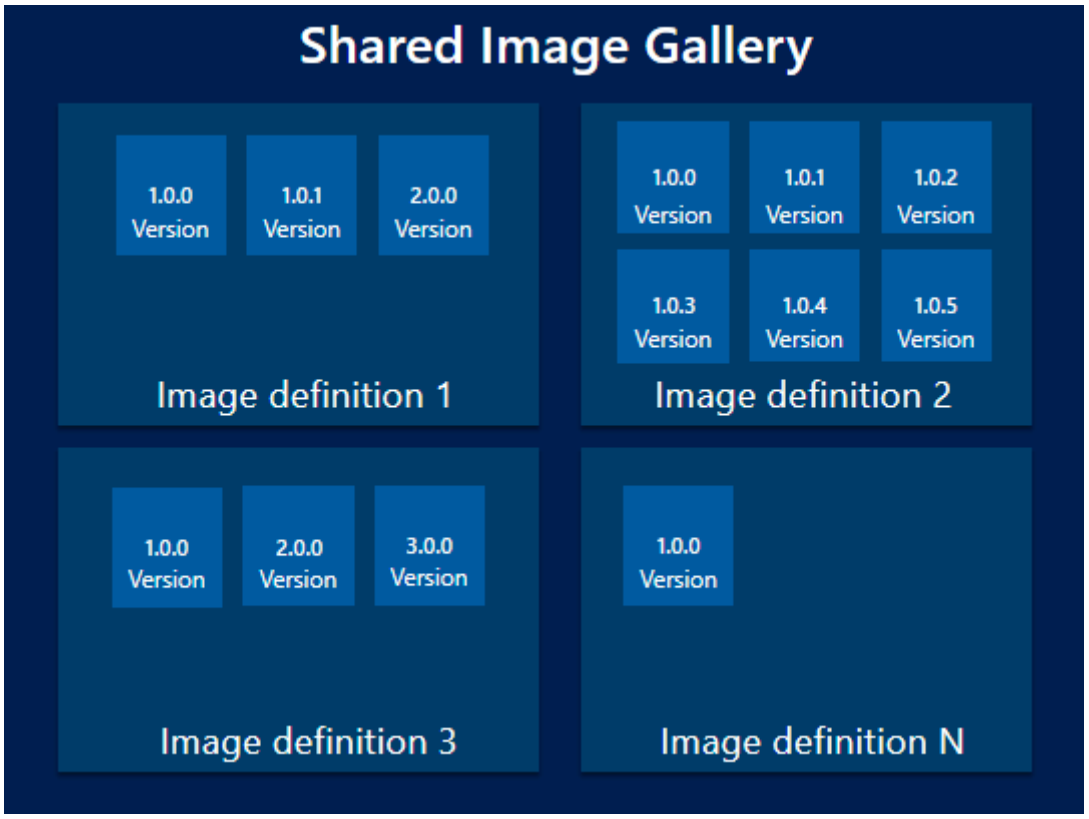


Image definitions

Image definitions are a logical grouping for versions of an image. The image definition holds information about why the image was created and also contains Image metadata such as, what OS it is for, features it supports and other information about using the image. An image definition is like a plan for all of the details around creating a specific image. You don't deploy a VM from an image definition, but from the image versions created from the definition.

There are three parameters for each image definition that are used in combination - **Publisher**, **Offer** and **SKU**. These are used to find a specific image definition. You can have image definitions that share one or two, but not all three values. For example, here are three image definitions and their values:

Expand table

Image Definition	Publisher	Offer	Sku
myImage1	Contoso	Finance	Backend
myImage2	Contoso	Finance	Frontend
myImage3	Testing	Finance	Frontend

All three of these have unique sets of values. The format is similar to how you can currently specify publisher, offer, and SKU for Azure Marketplace images in Azure PowerShell to get the latest version of a Marketplace image. Each image definition needs to have a unique set of these values.

The following parameters determine which types of image versions they can contain:

- Operating system state - You can set the OS state to generalized or specialized. This field is required.
- Operating system - can be either Windows or Linux. This field is required.
- Hyper-V generation - specify whether the image was created from a generation 1 or generation 2 Hyper-V VHD. Default is generation 1.

Image definitions contain metadata for the image to allow grouping of images that support the same features, plan, OS State, OS type and others. The following are other parameters that can be set on your image definition so that you can more easily track your resources:

- Description - use description to give more detailed information on why the image definition exists. For example, you might have an image definition for your front-end server that has the application preinstalled.
- EULA - can be used to point to an end-user license agreement specific to the image definition.
- Privacy Statement and Release notes - store release notes and privacy statements in Azure storage and provide a URI for accessing them as part of the image definition.
- End-of-life date - establish a default date after which the image shouldn't be used, for all image versions in the image definition. End-of-life dates are informational; users will still be able to create VMs from images and versions past the end-of-life date.
- Tag - you can add tags when you create your image definition. For more information about tags, see Using tags to organize your resources
- Minimum and maximum vCPU and memory recommendations - if your image has vCPU and memory recommendations, you can attach that information to your image definition.
- Disallowed disk types - you can provide information about the storage needs for your VM. For example, if the image isn't suited for standard HDD disks, you add them to the disallow list.

- Purchase plan information for Marketplace images - `-PurchasePlanPublisher`, `-PurchasePlanName`, and `-PurchasePlanProduct`. For more information about purchase plan information, see Find images in the Azure Marketplace and Supply Azure Marketplace purchase plan information when creating images.
- Architecture
  - x64 or ARM64 Architecture
- Features allow you to specify additional features and SecurityType(s) that are supported on the image, based on the type of gallery:

Expand table

Features	Accepted Values	Definition	Supported in
IsHibernateSupported	True, False	Create VMs with support for hibernation.	Private, direct shared, community
IsAcceleratedNetworkSupported	True, False	Create VMs with accelerated networking enabled. When set to `True` on Image definition, capturing VMs that don't support accelerated networking is not supported.	Private, direct shared, community
DiskControllerType	["SCSI", "NVMe"], ["SCSI"]	Set this to use either SCSI or NVMe disk type. NVMe VMs and disks can only be captured in image definitions that are tagged to be supporting NVMe.	Private, direct shared, community

When you specify a SecurityType using the `features` parameter, it limits the security features that are enabled on the VM. Some types limited, based on the type of gallery that they are stored in:

Expand table

SecurityType	Definition	Supported in
ConfidentialVMSupported	It's a generic Gen2 image that does not contain VMGS blob. Gen2 VM or Confidential VM can be created from this image type	Private, Direct shared, Community
Confidential VM	Only Confidential VMs can be created from this image type	Private
TrustedLaunchSupported	It's a generic Gen2 image that does not contain the VMGS blob. Gen2 VM or TrustedLaunch VM can be created from this image type.	Private, direct shared, community
TrustedLaunch	Only TrustedLaunch VM can be created from this image type	Private
TrustedLaunchAndConfidentialVmSupported	It's a generic Gen2 image that does not contain the VMGS blob. Gen2 VM, TrustedLaunch VM, or a ConfidentialVM can be created from this image type.	Private, direct shared, community

For more information, see the CLI examples for adding image definition features and SecurityType or the PowerShell examples.

**ConfidentialVM is only supported in the regions where it's available, You can find the supported regions here.

Image versions

An **image version** is what you use to create a VM. You can have multiple versions of an image as needed for your environment. When you use an **image version** to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times.

The properties of an image version are:

- Version number. This is used as the name of the image version. It is always in the format: MajorVersion.MinorVersion.Patch. When you specify to use **latest** when creating a VM, the latest image is chosen based on the highest MajorVersion, then MinorVersion, then Patch.
- Source. The source can be a VM, managed disk, snapshot, managed image, or another image version.
- End of life date. Indicate the end-of-life date for the image version. End-of-life dates are informational; users will still be able to create VMs from versions past the end-of-life date.

Generalized and specialized images

There are two operating system states supported by Azure Compute Gallery. Typically images require that the VM used to create the image has been generalized before taking the image. Generalizing is a process that removes machine and user specific information from the VM. For Linux, you can use waagent `-deprovision` or `-deprovision+user` parameters. For Windows, the Sysprep tool is used.

Specialized VMs haven't been through a process to remove machine specific information and accounts. Also, VMs created from specialized images don't have an `osProfile` associated with them. This means that specialized images will have some limitations in addition to some benefits.

- VMs and scale sets created from specialized images can be up and running quicker. Because they're created from a source that has already been through first boot, VMs created from these images boot faster.
- Accounts that could be used to log into the VM can also be used on any VM created using the specialized image that is created from that VM.
- VMs will have the **Computer name** of the VM the image was taken from. You should change the computer name to avoid collisions.
- The `osProfile` is how some sensitive information is passed to the VM, using `secrets`. This may cause issues using KeyVault, WinRM and other functionality that uses `secrets` in the `osProfile`. In some cases, you can use managed service identities (MSI) to work around these limitations.

> **Note**
>
> Generalized and specialized VM images contain an operating system disk and all the attached disks, if there any.

Updating resources

Once created, you can make some changes to the gallery resources. These are limited to:

Azure Compute Gallery:

- Description

Image definition:

- Recommended vCPUs
- Recommended memory
- Description
- End of life date
- ReleaseNotes

Image version:

- Regional replica count
- Target regions
- Exclude from latest
- End of life date

Sharing

There are three main ways to share images an Azure Compute Gallery, depending on who you want to share with:

Expand table

Sharing with:	People	Groups	Service Principal	All users in a specific subscription (or) tenant	Publicly with all users in Azure
RBAC Sharing	Yes	Yes	Yes	No	No
RBAC + Direct shared gallery	Yes	Yes	Yes	Yes	No
RBAC + Community gallery	Yes	Yes	Yes	No	Yes

RBAC Permissions required to create an ACG Image:

ACG images can be created by users from various sources, including virtual machines, disks/snapshots, and VHDs. The section outlines the various user permissions necessary for creating an Azure Compute Gallery image. Identifies without the necessary permissions will not be able to create ACG images.

VM as source

- Users will require write permission on the Virtual Machine to create an ACG Image version.
- For Azure SDK, use the property properties.storageProfile.source.virtualMachineId, This property requires API version 2023-07-03 or Version 1.4.0 (or higher) of .NET SDK

Expand table

Source type	Permissions Required
Virtual machine	Write
Disk/snapshot	Write
VHD	Write (listKeys)
Managed Image	Read
Gallery Image	Read

Refer to our documentation for additional information regarding Azure built-in roles, for granting RBAC permissions

Shallow replication

When you create an image version, you can set the replication mode to shallow for development and test. Shallow replication skips copying the image, so the image version is ready faster. But, it also means you can't deploy a large number of VMs from that image version. This is similar to the way that the older managed images worked.

Shallow replication can also be useful if you have large images (up to 32 TB) that aren't frequently deployed. Because the source image isn't copied, larger disks can be used.

But, they also can't be used for deploying large numbers of VMs concurrently.

To set an image for shallow replication, use `--replication-mode Shallow` with the Azure CLI.

SDK support

The following SDKs support creating Azure Compute Galleries:

- .NET
- Java
- Node.js
- Python
- Go

Templates

You can create Azure Compute Gallery resource using templates. There are several quickstart templates available:

- Create a gallery
- Create an image definition in a gallery
- Create an image version in a gallery

Frequently asked questions

- How can I list all the Azure Compute Gallery resources across subscriptions?
- Can I move my existing image to an Azure Compute Gallery?
- Can I create an image version from a specialized disk?
- Can I move the Azure Compute Gallery resource to a different subscription after it has been created?
- Can I replicate my image versions across clouds such as Microsoft Azure operated by 21Vianet, Azure Germany, or Azure Government Cloud?
- Can I replicate my image versions across subscriptions?
- Can I share image versions across Microsoft Entra tenants?
- How long does it take to replicate image versions across the target regions?
- What is the difference between source region and target region?
- How do I specify the source region while creating the image version?
- How do I specify the number of image version replicas to be created in each region?

- Can I create the gallery in a different location than the one for the image definition and image version?
- What are the charges for using an Azure Compute Gallery?
- What API version should I use when creating images?
- What API version should I use to create a VM or Virtual Machine Scale Set out of the image version?
- Can I update my Virtual Machine Scale Set created using managed image to use Azure Compute Gallery images?
- How can I update my code to use the new property and ensure permissions are granted accurately during VM image creation?
- Does deleting the Azure Compute Gallery images affect existing VMs created from it?

How can I list all the Azure Compute Gallery resources across subscriptions?

To list all the Azure Compute Gallery resources across subscriptions that you have access to on the Azure portal, follow the steps below:

1. Open the Azure portal.
2. Scroll down the page and select **All resources**.
3. Select all the subscriptions under which you'd like to list all the resources.
4. Look for resources of the **Azure Compute Gallery** type.

Azure CLI

To list all the Azure Compute Gallery resources, across subscriptions that you have permissions to, use the following command in the Azure CLI:

Azure CLI

```
   az account list -otsv --query "[].id" | xargs -n 1 az sig list --
subscription
```

Can I move my existing image to an Azure Compute Gallery?

Yes. There are 3 scenarios based on the types of images you may have.

Scenario 1: If you have a managed image, then you can create an image definition and image version from it. For more information, see Create and image definition and an image version.

Scenario 2: If you have an unmanaged image, you can create a managed image from it, and then create an image definition and image version from it.

Scenario 3: If you have a VHD in your local file system, then you need to upload the VHD to a managed image, then you can create an image definition and image version from it.

- If the VHD is of a Windows VM, see Upload a VHD.
- If the VHD is for a Linux VM, see Upload a VHD

Can I create an image version from a specialized disk?

Yes, can create a VM from a specialized image.

Can I move the Azure Compute Gallery resource to a different subscription after it has been created?

No, you can't move the gallery image resource to a different subscription. You can replicate the image versions in the gallery to other regions or copy an image from another gallery.

Can I replicate my image versions across clouds such as Azure operated by 21Vianet or Azure Germany or Azure Government Cloud?

No, you can't replicate image versions across clouds.

Can I replicate my image versions across subscriptions?

No, you may replicate the image versions across regions in a subscription and use it in other subscriptions through RBAC.

Can I share image versions across Microsoft Entra tenants?

Yes, you can use RBAC to share to individuals across tenants. But, to share at scale, see "Share gallery images across Azure tenants" using PowerShell or CLI.

How long does it take to replicate image versions across the target regions?

The image version replication time is entirely dependent on the size of the image and the number of regions it's being replicated to. However, as a best practice, it's recommended that you keep the image small, and the source and target regions close for best results. You can check the status of the replication using the -ReplicationStatus flag.

What is the difference between source region and target region?

Source region is the region in which your image version will be created, and target regions are the regions in which a copy of your image version will be stored. For each image version, you can only have one source region. Also, make sure that you pass the source region location as one of the target regions when you create an image version.

How do I specify the source region while creating the image version?

While creating an image version, you can use the **--location** argument in CLI and the **-Location** parameter in PowerShell to specify the source region. Ensure the managed image that you're using as the base image to create the image version is in the same location as the location in which you intend to create the image version. Also, make sure that you pass the source region location as one of the target regions when you create an image version.

How do I specify the number of image version replicas to be created in each region?

There are two ways you can specify the number of image version replicas to be created in each region:

1. The regional replica count which specifies the number of replicas you want to create per region.
2. The common replica count which is the default per region count in case regional replica count isn't specified.

Azure CLI

To specify the regional replica count, pass the location along with the number of replicas you want to create in that region: "South Central US=2".

If regional replica count isn't specified with each location, then the default number of replicas will be the common replica count that you specified.

To specify the common replica count in Azure CLI, use the **--replica-count** argument in the `az sig image-version create` command.

Can I create the gallery in a different location than the one for the image definition and image version?

Yes, it's possible. But, as a best practice, we encourage you to keep the resource group, gallery, image definition, and image version in the same location.

What are the charges for using an Azure Compute Gallery?

There are no charges for using an Azure Compute Gallery, except the storage charges for storing the image versions and network egress charges for replicating the image versions from source region to target regions.

What API version should I use when creating images?

To work with galleries, image definitions, and image versions, we recommend you use API version 2018-06-01. Zone Redundant Storage (ZRS) requires version 2019-03-01 or later.

What API version should I use to create a VM or Virtual Machine Scale Set out of the image version?

For VM and Virtual Machine Scale Set deployments using an image version, we recommend you use API version 2018-04-01 or higher.

Can I update my Virtual Machine Scale Set created using managed image to use Azure Compute Gallery images?

Yes, you can update the scale set image reference from a managed image to an Azure Compute Gallery image, as long as the OS type, Hyper-V generation, and the data disk layout matches between the images.

How can I update my code to use the new property and ensure permissions are granted accurately during VM image creation?

For Virtual Machine ID field, use VirtualMachineId field under GallerySource(GalleryImageVersionStorageProfile.GallerySource.VirtualMachineID). The new property requires api-version 2023-07-03 or version 1.4.0 (or higher) of .NET SDK

```
StorageProfile = new GalleryImageVersionStorageProfile()
            {
                GallerySource = new GalleryArtifactVersionFullSource()
                {
                    VirtualMachineId = new
ResourceIdentifier(virtualMachineId),
                }
            },
```

For VHD as a source, use StorageAccountID field under GallerySource under OSDiskImage or Data disk Image(GalleryImageVersionStorageProfile.OSDiskImage.GallerySource.StorageAccountId). The new property requires api-version 2022-03-03

```
StorageProfile = new GalleryImageVersionStorageProfile()
            {
                OSDiskImage = new GalleryOSDiskImage()
                {
                    GallerySource = new GalleryDiskImageSource()
                    {
                        StorageAccountId = new
ResourceIdentifier(storageAccountId),
                        Uri = new Uri(blobUri),
                    }
                }
            },
```

Does deleting the Azure Compute Gallery affect VMs created from it?

VMs created from the Azure Compute Gallery image remains unaffected due to their persistent disks. However, VMSS scale out operation will fail as they rely on the source image ID reference which would be lost once the Azure Compute Gallery image is deleted.

Troubleshoot

If you have issues with performing any operations on the gallery resources, consult the list of common errors in the troubleshooting guide.

In addition, you can post and tag your question with `azure-virtual-machines-images` at Q&A.

Next steps

Learn how to deploy images using the Azure Compute Gallery.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an image of a VM in the portal

An image can be created from a VM and then used to create multiple VMs.

For images stored in an Azure Compute Gallery (formerly known as Shared Image Gallery), you can use VMs that already have accounts created on them (specialized) or you can generalize the VM before creating the image to remove machine accounts and other machines specific information. To generalize a VM, see Generalized a VM. For more information, see Generalized and specialized images.

> **Important**
>
> Once you mark a VM as `generalized` in Azure, you cannot restart the VM. Legacy **managed images** are automatically marked as generalized.
>
> > When capturing an image of a virtual machine in Azure, the virtual machine will be temporarily stopped to ensure data consistency and prevent any potential issues during the image creation. This is because capturing an image requires a point-in-time snapshot of the virtual machine's disk. To avoid disruptions in a production environment, it's recommended you schedule the image capture process during a maintenance window or a time when the temporary downtime won't impacting critical services.

Capture a VM in the portal

1. Go to the Azure portal, then search for and select **Virtual machines**.
2. Select your VM from the list.
    - If you want a generalized image, see Generalize OS disk for Linux/Windows.
    - If you want a specialized image, no additional action is required.
3. On the page for the VM, on the upper menu, select **Capture**.

   The **Create an image** page appears.
4. For **Resource group**, either select **Create new** and enter a name, or select a resource group to use from the drop-down list. If you want to use an existing gallery, select the resource group for the gallery you want to use.
5. To create the image in a gallery, select **Yes, share it to a gallery as an image version**.

To only create a managed image, select **No, capture only a managed image**. The VM must have been generalized to create a managed image. The only other required information is a name for the image.

6. If you want to delete the source VM after the image has been created, select **Automatically delete this virtual machine after creating the image**. This is not recommended.
7. For **Gallery details**, select the gallery or create a new gallery by selecting **Create new**.
8. In **Operating system state** select generalized or specialized. For more information, see Generalized and specialized images.
9. Select an image definition or select **create new** and provide a name and information for a new Image definition.
10. Enter an image version number. If this is the first version of this image, type *1.0.0*.
11. If you want this version to be included when you specify *latest* for the image version, then leave **Exclude from latest** unchecked.
12. Select an **End of life** date. This date can be used to track when older images need to be retired.
13. Under Replication, select a default replica count and then select any additional regions where you would like your image replicated.
14. When you are done, select **Review + create**.
15. After validation passes, select **Create** to create the image.

Next steps

- Azure Compute Galleries overview

Last updated on 02/06/2026

Create an image definition and an image version

An Azure Compute Gallery simplifies custom image sharing across your organization. Custom images are like marketplace images, but you create them yourself. Images can be created from a VM, VHD, snapshot, managed image, or another image version.

The Azure Compute Gallery lets you share your custom VM images with others in your organization, within or across regions, within a Microsoft Entra tenant, or publicly using a community gallery. Choose which images you want to share, which regions you want to make them available in, and who you want to share them with. You can create multiple galleries so that you can logically group images. Many new features like ARM64, Accelerated Networking and TrustedVM are only supported through Azure Compute Gallery and not available for managed images.

The Azure Compute Gallery feature has multiple resource types:

Expand table

Resource	Description
Image source	This is a resource that can be used to create an **image version** in a gallery. An image source can be an existing Azure VM that is either generalized or specialized, a managed image, a snapshot, or an image version in another gallery.
Gallery	Like the Azure Marketplace, a **gallery** is a repository for managing and sharing images and VM applications, but you control who has access.
Image definition	Image definitions are created within a gallery and carry information about the image and requirements for using it internally. This includes whether the image is Windows or Linux, release notes, and minimum and maximum memory requirements. It is a definition of a type of image.
Image version	An **image version** is what you use to create a VM when using a gallery. You can have multiple versions of an image as needed for your environment. Like a managed image, when you use an **image version** to create a VM, the image version is used to create new disks for the VM. Image versions can be used multiple times.

Prerequisites

To complete this article, you must have an existing Azure Compute Gallery, and a source for your image available in Azure. Image sources can be:

- A VM in your subscription. You can capture an image from both specialized and generalized VMs.
- A Managed image,
- Managed OS and data disks.
- OS and data disks as VHDs in a storage account.
- Other image versions either in the same gallery or another gallery in the same subscription.

If the image will contain data disks, the data disk size cannot be more than 1 TB.

Image definition names can be made up of uppercase or lowercase letters, digits, dots, dashes and periods. For more information about the values you can specify for an image definition, see Image definitions.

Allowed characters for the image version are numbers and periods. Numbers must be within the range of a 32-bit integer. Format: *MajorVersion*.*MinorVersion*.*Patch*.

When working through this article, replace the resource names where needed.

For generalized images, see the OS specific guidance before capturing the image:

- **Linux**
  - Generic steps
  - CentOS
  - Debian
  - Flatcar
  - FreeBSD
  - Oracle Linux
  - OpenBSD
  - Red Hat
  - SUSE
  - Ubuntu
- **Windows**

  If you plan to run Sysprep before uploading your virtual hard disk (VHD) to Azure for the first time, make sure you have prepared your VM.

Community gallery

If you will be sharing your images using a community gallery, make sure that you create your gallery, image definitions, and image versions in the same region.

When users search for community gallery images, only the latest version of an image is shown.

> **Important**
>
> Information from your image definitions will be publicly available, like what you provide for **Publish**, **Offer**, and **SKU**.

Create an image

Choose an option below for creating your image definition and image version:

Portal

To create an image from a VM in the portal, see Capture an image of a VM.

To create an image using a source other than a VM, follow these steps.

1. Go to the Azure portal, then search for and select **Azure Compute Gallery**.
2. Select the gallery you want to use from the list.
3. On the page for your gallery, select **Add** from the top of the page and then select **VM image definition** from the drop-down.
4. on the **Add new image definition to Azure Compute Gallery** page, in the **Basics** tab, select a **Region**.
5. For **Image definition name**, type a name like *myImageDefinition*.
6. For **Operating system**, select the correct option based on your source.
7. For **VM generation**, select the option based on your source. In most cases, this will be *Gen 1*. For more information, see Support for generation 2 VMs.
8. For **Operating system state**, select the option based on your source. For more information, see Generalized and specialized.

9. For **Publisher**, type a unique name like *myPublisher*.
10. For **Offer**, type a unique name like *myOffer*.
11. For **SKU**, type a unique name like *mySKU*.
12. At the bottom of the page, select **Review + create**.
13. After the image definition passes validation, select **Create**.
14. When the deployment is finished, select **Go to resource**.
15. In the page for your image definition, on the **Get started** tab, select **Create a version**.
16. In **Region**, select the region where you want the image created. In some cases, the source must be in the same region where the image is created. If you aren't seeing your source listed in later drop-downs, try changing the region for the image. You can always replicate the image to other regions later.
17. For **Version number**, type a number like *1.0.0*. The image version name should follow *major.minor.patch* format using integers.
18. In **Source**, select the type of file you are using for your source from the drop-down. See the table below for specific details for each source type.

Expand table

Source	Other fields
Disks or snapshots	- For **OS disk** select the disk or snapshot from the drop-down. - To add a data disk, type the LUN number and then select the data disk from the drop-down.
Image version	- Select the **Source gallery** from the drop-down. - Select the correct image definition from the drop-down. - Select the existing image version that you want to use from the drop-down.
Managed image	Select the **Source image** from the drop-down. The managed image must be in the same region that you chose in **Instance details**.
VHD in a storage account	Select **Browse** to choose the storage account for the VHD.

19. In **Exclude from latest**, leave the default value of *No* unless you don't want this version used when creating a VM using `latest` instead of a version number.
20. For **End of life date**, select a date from the calendar for when you think this version should stop being used.
21. In the **Replication** tab, select the storage type from the drop-down.
22. Set the **Default replica count**, you can override this for each region you add.
23. You need to replicate to the source region, so the first replica in the list will be in the region where you created the image. You can add more replicas by selecting the region from the drop-down and adjusting the replica count as necessary.
24. When you are done, select **Review + create**. Azure will validate the configuration.
25. When image version passes validation, select **Create**.
26. When the deployment is finished, select **Go to resource**.

It can take a while to replicate the image to all of the target regions.

You can also capture an existing VM as an image, from the portal. For more information, see Create an image of a VM in the portal.

Create an image version in one tenant using the source image version in another tenant

In the subscription where the source image version exists, grant reader permissions to the user. Once the user has reader permission to the source image version, sign in to both accounts (source and target).

You will need the `tenantID` of the source image version, the `subscriptionID` for the subscription where the new image version will be stored (target), and the `resourceID` of the source image version. The target gallery image definition must already exist before you create the image version with `az sig image-version create` or `New-AzGalleryImageVersion`. Additionally, ensure that the source image version's region or replica and target region are the same.

CLI

Use the Azure CLI to create an image version in one tenant using the source image version in another tenant.

Azure CLI

```
# Set some variables
tenantID="<tenant ID for the source image version>"
subID="<subscription ID where the image version will be created>"
sourceImageVersionID="<resource ID of the source image version>"

# Sign in to the subscription where the new image version will be created
az login

# Sign in to the tenant where the source image version is available
az login --tenant $tenantID

# Sign back in to the subscription where the image version will be created and
ensure subscription context is set
az login
az account set --subscription $subID

# Create the image version from the source image version
az sig image-version create \
   --gallery-image-definition myImageDef \
   --gallery-image-version 1.0.0 \
   --gallery-name myGallery \
   --resource-group myResourceGroup \
   --image-version $sourceImageVersionID \
   --location myLocation
```

Next steps

For information about how to supply purchase plan information, see Supply Azure Marketplace purchase plan information when creating images.

Last updated on 04/10/2026

Create a VM from a generalized image version

Article • 08/22/2024

Create a VM from a generalized image version stored in an Azure Compute Gallery (formerly known as Shared Image Gallery). If you want to create a VM using a specialized image, see Create a VM from a specialized image.

This article shows how to create a VM from a generalized image:

- In your own gallery
- Shared within your organization using RBAC
- Shared across tenants using RBAC
- Shared to everyone in a community gallery
- Directly shared to your subscription or tenant

Your own gallery

CLI

List the image definitions in a gallery using az sig image-definition list to see the name and ID of the definitions.

Azure CLI

```
resourceGroup=myGalleryRG
gallery=myGallery
az sig image-definition list --resource-group $resourceGroup --gallery-
name $gallery --query "[].[name, id]" --output tsv
```

Create a VM using az vm create. To use the latest version of the image, set `--image` to the ID of the image definition.

This example is for creating a Linux VM secured with SSH. For Windows or to secure a Linux VM with a password, remove `--generate-ssh-keys` to be prompted for a password. If you want to supply a password directly, replace `--generate-ssh-keys` with `--admin-password`. Replace resource names as needed in this example.

Azure CLI

```
imgDef="/subscriptions/<subscription ID where the gallery is
located>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/gallerie
s/myGallery/images/myImageDefinition"
vmResourceGroup=myResourceGroup
location=eastus
vmName=myVM
adminUsername=azureuser

az group create --name $vmResourceGroup --location $location

az vm create\
   --resource-group $vmResourceGroup \
   --name $vmName \
   --image $imgDef \
   --admin-username $adminUsername \
   --generate-ssh-keys
```

You can also use a specific version by using the image version ID for the `--image` parameter. For example, to use image version *1.0.0* type: `--image "/subscriptions/<subscription ID where the gallery is located>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myGallery/images/myImageDefinition/versions/1.0.0"`.

RBAC - Shared within your organization

If the subscription where the gallery resides is within the same tenant, images shared through RBAC can be used to create VMs using the CLI and PowerShell.

You'll need to the `imageID` of the image you want to use and you need to make sure it's replicated to the region where you want to create the VM.

CLI

Make sure the state of the image is `Generalized`. If you want to use an image with the `Specialized` state, see Create a VM from a specialized image version.

Azure CLI

```
imgDef="/SharedGalleries/1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f-
MYDIRECTSHARED/Images/myDirectDefinition/Versions/latest"
vmResourceGroup=myResourceGroup
location=westus
vmName=myVM
adminUsername=azureuser
```

```
az group create --name $vmResourceGroup --location $location

az vm create\
   --resource-group $vmResourceGroup \
   --name $vmName \
   --image $imgDef \
   --admin-username $adminUsername \
   --generate-ssh-keys
```

RBAC - Shared from another tenant

If the image you want to use is stored in a gallery that isn't in the same tenant (directory) then you need to sign in to each tenant to verify you have access.

You also need the `imageID` of the image you want to use and you need to make sure it's replicated to the region where you want to create the VM. You'll also need the `tenantID` for the source gallery and the `tenantID` for where you want to create the VM.

CLI

In this example, we're showing how to create a VM from a generalized image. If you're using a specialized image, see Create a VM using a specialized image version.

You need to sign in to the tenant where the image is stored, get an access token, then sign into the tenant where you want to create the VM. In this case, tenant1 is where the image is stored, and tenant2 is where you want to create the VM. This is how Azure authenticates that you have access to the image.

Azure CLI

```
tenant1='<ID for tenant 1>'
tenant2='<ID for tenant 2>'

az account clear
az login --tenant $tenant1
az account get-access-token
az login --tenant $tenant2
az account get-access-token
```

Create the VM. Replace the information in the example with your own. Before you create the VM, make sure that the image is replicated into the region where you want to create the VM.

Azure CLI

```
imageid="<ID of the image that you want to use>"
resourcegroup="<name for the resource group>"
location="<location where the image is replicated>"
user='<username for the VM>'
name='<name for the VM>'

az group create --location $location --resource-group $resourcegroup
az vm create \
  --resource-group $resourcegroup \
  --name $name \
  --image $imageid \
  --admin-username $user \
  --generate-ssh-keys
```

Community gallery

> **Important**
>
> Microsoft does not provide support for images in the community gallery.

Reporting issues with a community image

Using community-submitted virtual machine images has several risks. Images could contain malware, security vulnerabilities, or violate someone's intellectual property. To help create a secure and reliable experience for the community, you can report images when you see these issues.

The easiest way to report issues with a community gallery is to use the portal, which will pre-fill information for the report:

- For issues with links or other information in the fields of an image definition, select **Report community image**.
- If an image version contains malicious code or there are other issues with a specific version of an image, select **Report** under the **Report version** column in the table of image versions.

You can also use the following links to report issues, but the forms won't be pre-filled:

- Malicious images: Contact Abuse Report.
- Intellectual Property violations: Contact Infringement Report.

CLI

To create a VM using an image shared to a community gallery, use the unique ID of the image for the `--image` which will be in the following format:

```
/CommunityGalleries/<community gallery name, like: ContosoImages-
1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f>/Images/<image
name>/Versions/latest
```

Follow these instructions to get the list of Community images using CLI:

```
Step 1:  Show all 'Community images' in a specific location
az sig list-community --location westus2

Step 2: Once you have the public gallery name from Step 1, Get the Image
definition (Name) of the image by running the following command
az sig image-definition list-community --public-gallery-name <<public
gallery name>> --location westus2

Step 3: Finally, run the following command to list different image
versions available for the specific image
az sig image-version list-community --public-gallery-name
<<galleryname>> --gallery-image-definition <<image name>> --location
westus2
```

To get the public name of a community gallery from portal. Go to **Virtual machines** > **Create** > **Azure virtual machine** > **Image** > **See all images** > **Community Images** > **Public gallery name**.

In this example, we're creating a VM from a Linux image and creating SSH keys for authentication.

Azure CLI

```
imgDef="/CommunityGalleries/ContosoImages-1a2b3c4d-1234-abcd-1234-
1a2b3c4d5e6f>/Images/myLinuxImage/Versions/latest"
vmResourceGroup=myResourceGroup
location=eastus
vmName=myVM
adminUsername=azureuser

az group create --name $vmResourceGroup --location $location

az vm create\
```

```
    --resource-group $vmResourceGroup \
    --name $vmName \
    --image $imgDef \
    --admin-username $adminUsername \
    --generate-ssh-keys
```

When using a community image, you'll be prompted to accept the legal terms. The message looks like this:

Output

```
To create the VM from community gallery image, you must accept the
license agreement and privacy statement: http://contoso.com. (If you
want to accept the legal terms by default, please use the option '--
accept-term' when creating VM/VMSS) (Y/n):
```

Direct shared gallery

Important

Azure Compute Gallery – direct shared gallery is currently in PREVIEW and subject to the Preview Terms for Azure Compute Gallery.

To publish images to a direct shared gallery during the preview, you need to register at https://aka.ms/directsharedgallery-preview. Creating VMs from a direct shared gallery is open to all Azure users.

During the preview, you need to create a new gallery, with the property `sharingProfile.permissions` set to `Groups`. When using the CLI to create a gallery, use the `--permissions groups` parameter. You can't use an existing gallery, the property can't currently be updated.

CLI

To create a VM using an image shared to your subscription or tenant, you need the unique ID of the image in the following format:

```
/SharedGalleries/<uniqueID>/Images/<image name>/Versions/latest
```

To find the `uniqueID` of a gallery that is shared with you, use az sig list-shared. In this example, we are looking for galleries in the West US region.

Azure CLI

```
region=westus
az sig list-shared --location $region --query "[].name" -o tsv
```

Use the gallery name to find the images that are available. In this example, we list all of the images in *West US* and by name, the unique ID that is needed to create a VM, OS and OS state.

Azure CLI

```
galleryName="1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f-myDirectShared"
 az sig image-definition list-shared \
   --gallery-unique-name $galleryName \
   --location $region \
   --query [*]."{Name:name,ID:uniqueId,OS:osType,State:osState}" -o
table
```

Make sure the state of the image is `Generalized`. If you want to use an image with the `Specialized` state, see Create a VM from a specialized image version.

Use the `Id` from the output, appended with `/Versions/latest` to use the latest version, as the value for `--image` to create a VM. In this example, we're creating a VM from a Linux image that is directly shared to us, and creating SSH keys for authentication.

Azure CLI

```
imgDef="/SharedGalleries/1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f-
MYDIRECTSHARED/Images/myDirectDefinition/Versions/latest"
vmResourceGroup=myResourceGroup
location=westus
vmName=myVM
adminUsername=azureuser

az group create --name $vmResourceGroup --location $location

az vm create\
   --resource-group $vmResourceGroup \
   --name $vmName \
   --image $imgDef \
   --admin-username $adminUsername \
   --generate-ssh-keys
```

Next steps

- Create an Azure Compute Gallery
- Create an image in an Azure Compute Gallery

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a VM using a specialized image version

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

Create a VM from a specialized image version stored in an Azure Compute Gallery (formerly known as Shared Image Gallery). If you want to create a VM using a generalized image version, see Create a VM from a generalized image version.

This article shows how to create a VM from a specialized image:

- In your own gallery
- Shared within your organization using RBAC
- Shared across tenants using RBAC
- Shared to everyone in a community gallery
- Directly shared to your subscription or tenant

> **ⓘ Important**
>
> When you create a new VM from a specialized image, the new VM retains the computer name of the original VM. Other computer-specific information, like the CMID, is also kept. This duplicate information can cause issues. When copying a VM, be aware of what types of computer-specific information your applications rely on.

Create a VM from your gallery

Create a VM from an internal gallery.

CLI

List the image definitions in a gallery using az sig image-definition list to see the name and ID of the definitions.

Azure CLI

```
resourceGroup=myGalleryRG
gallery=myGallery
az sig image-definition list \
   --resource-group $resourceGroup \
```

```
    --gallery-name $gallery \
    --query "[].[name, id]" \
    --output tsv
```

Create the VM using az vm create using the `--specialized` parameter to indicate that the image is a specialized image.

Use the image definition ID for `--image` to create the VM from the latest version of the image that is available. You can also create the VM from a specific version by supplying the image version ID for `--image`.

In this example, we're creating a VM from the latest version of the *myImageDefinition* image.

Azure CLI

```
az group create --name myResourceGroup --location eastus
az vm create --resource-group myResourceGroup \
    --name myVM \
    --image "/subscriptions/<Subscription
ID>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myG
allery/images/myImageDefinition" \
    --specialized
```

RBAC - within your organization

If the subscription where the gallery resides is within the same tenant, images shared through RBAC can be used to create VMs using the CLI and PowerShell.

You'll need the `imageID` of the image you want to use and make sure the image is replicated to the region where you want to create the VM.

CLI

Azure CLI

```
image="/subscriptions/<Subscription
ID>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myG
allery/images/myImageDefinition"
vmResourceGroup='myResourceGroup'
location='westus'
vmName='myVM'

az group create --name $vmResourceGroup --location $location
```

```
az vm create\
   --resource-group $vmResourceGroup \
   --name $vmName \
   --image $image \
   --specialized
```

RBAC - from another tenant or organization

If the image you want to use is stored in a gallery that isn't in the same tenant (directory) then you will need to sign in to each tenant to verify you have access.

You'll need the `imageID` of the image you want to use and make sure the image is replicated to the region where you want to create the VM. You'll also need the `tenantID` for the source gallery and the `tenantID` for where you want to create the VM.

CLI

You need to sign in to the tenant where the image is stored, get an access token, then sign into the tenant where you want to create the VM. This is how Azure authenticates that you have access to the image.

Azure CLI

```
tenant1='<ID for tenant 1>'
tenant2='<ID for tenant 2>'

az account clear
az login --tenant $tenant1
az account get-access-token
az login --tenant $tenant2
az account get-access-token
```

Create the VM using az vm create using the `--specialized` parameter to indicate that the image is a specialized image.

Azure CLI

```
imageid=""/subscriptions/<Subscription
ID>/resourceGroups/myGalleryRG/providers/Microsoft.Compute/galleries/myG
allery/images/myImageDefinition""
resourcegroup="myResourceGroup"
location="West US 3"
```

```
name='myVM'

az group create --name $resourcegroup --location $location
az vm create --resource-group $resourcegroup \
    --name $name \
    --image $image \
    --specialized
```

Community gallery

> **Important**
>
> Microsoft does not provide support for images in the community gallery.

Reporting issues with a community image

Using community-submitted virtual machine images has several risks. Images could contain malware, security vulnerabilities, or violate someone's intellectual property. To help create a secure and reliable experience for the community, you can report images when you see these issues.

The easiest way to report issues with a community gallery is to use the portal, which will pre-fill information for the report:

- For issues with links or other information in the fields of an image definition, select **Report community image**.
- If an image version contains malicious code or there are other issues with a specific version of an image, select **Report** under the **Report version** column in the table of image versions.

You can also use the following links to report issues, but the forms won't be pre-filled:

- Malicious images: Contact Abuse Report.
- Intellectual Property violations: Contact Infringement Report.

CLI

To create a VM using an image shared to a community gallery, use the unique ID of the image for the `--image`, which will be in the following format:

```
```

```
/CommunityGalleries/<community gallery name, like: ContosoImages-
1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f>/Images/<image
name>/Versions/latest
```

As an end user, to get the public name of a community gallery, you need to use the portal. Go to **Virtual machines** > **Create** > **Azure virtual machine** > **Image** > **See all images** > **Community Images** > **Public gallery name**.

List all of the image definitions that are available in a community gallery using az sig image-definition list-community. In this example, we list all of the images in the *ContosoImage* gallery in *West US* and by name, the unique ID that is needed to create a VM, OS and OS state.

Azure CLI

```
 az sig image-definition list-community \
   --public-gallery-name "ContosoImages-1a2b3c4d-1234-abcd-1234-
1a2b3c4d5e6f" \
   --location westus \
   --query [*]."{Name:name,ID:uniqueId,OS:osType,State:osState}" -o
table
```

To create a VM from a generalized image in a community gallery, see Create a VM from a generalized image version.

Create the VM using az vm create using the `--specialized` parameter to indicate that the image is a specialized image.

In this example, we're creating a VM from the latest version of the *myImageDefinition* image.

Azure CLI

```
az group create --name myResourceGroup --location eastus
az vm create --resource-group myResourceGroup \
    --name myVM \
    --image "/CommunityGalleries/ContosoImages-f61bb1d9-3c5a-4ad2-99b5-
744030225de6/Images/LinuxSpecializedVersions/latest" \
    --specialized
```

When using a community image, you'll be prompted to accept the legal terms. The message will look like this:

Output

```
To create the VM from community gallery image, you must accept the
license agreement and privacy statement: http://contoso.com. (If you
want to accept the legal terms by default, please use the option '--
accept-term' when creating VM/VMSS) (Y/n):
```

Direct shared gallery

> **ⓘ Important**
>
> Azure Compute Gallery – direct shared gallery is currently in PREVIEW and subject to the Preview Terms for Azure Compute Gallery.
>
> To publish images to a direct shared gallery during the preview, you need to register at https://aka.ms/directsharedgallery-preview. Creating VMs from a direct shared gallery is open to all Azure users.
>
> During the preview, you need to create a new gallery, with the property `sharingProfile.permissions` set to `Groups`. When using the CLI to create a gallery, use the `--permissions groups` parameter. You can't use an existing gallery, the property can't currently be updated.

CLI

To create a VM using the latest version of an image shared to your subscription or tenant, you need the ID of the image in the following format:

```
/SharedGalleries/<uniqueID>/Images/<image name>/Versions/latest
```

To find the `uniqueID` of a gallery that is shared with you, use az sig list-shared. In this example, we're looking for galleries in the West US region.

Azure CLI

```azurecli
region=westus
az sig list-shared --location $region --query "[].name" -o tsv
```

Use the gallery name to find all of the images that are available. In this example, we list all of the images in *West US* and by name, the unique ID that is needed to create

a VM, OS and OS state.

Azure CLI

```
galleryName="1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f-myDirectShared"
 az sig image-definition list-shared \
   --gallery-unique-name $galleryName \
   --location $region \
   --query [*]."{Name:name,ID:uniqueId,OS:osType,State:osState}" -o
table
```

Make sure the state of the image is `Specialized`. If you want to use an image with the `Generalized` state, see Create a VM from a generalized image version.

Create the VM using az vm create using the `--specialized` parameter to indicate that the image is a specialized image.

Use the `Id`, appended with `/Versions/latest` to use the latest version, as the value for `--image`` to create a VM.

In this example, we're creating a VM from the latest version of the *myImageDefinition* image.

Azure CLI

```
imgDef="/SharedGalleries/1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f-
MYDIRECTSHARED/Images/myDirectDefinition/Versions/latest"
vmResourceGroup=myResourceGroup
location=westus
vmName=myVM

az group create --name $vmResourceGroup --location $location

az vm create\
   --resource-group $vmResourceGroup \
   --name $vmName \
   --image $imgDef \
   --specialized
```

Next steps

- Create an Azure Compute Gallery
- Create an image in an Azure Compute Gallery

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

List, update, and delete gallery resources

Article • 08/22/2024

You can manage your Azure Compute Gallery (formerly known as Shared Image Gallery) resources using the Azure CLI or Azure PowerShell.

List galleries shared with you

CLI

List Galleries shared with your subscription.

Azure CLI

```
region=westus
az sig list-shared --location $region
```

List Galleries shared with your tenant.

Azure CLI

```
region=westus
az sig list-shared --location $region --shared-to tenant
```

The output will contain the public `name` and `uniqueID` of the gallery that is shared with you. You can use the name of the gallery to query for images that are available through the gallery.

Here is example output:

Output

```
[
  {
    "location": "westus",
    "name": "1231b567-8a99-1a2b-1a23-123456789abc-MYDIRECTSHARED",
    "uniqueId": "/SharedGalleries/1231b567-8a99-1a2b-1a23-123456789abc-
MYDIRECTSHARED"
  }
]
```

Update resources

There are some limitations on what can be updated. The following items can be updated:

Azure Compute Gallery:

- Description

Image definition:

- Recommended vCPUs
- Recommended memory
- Description
- End of life date

Image version:

- Regional replica count
- Target regions
- Exclusion from latest
- End of life date

CLI

Update the description of a gallery using (az sig update.

Azure CLI

```
az sig update \
   --gallery-name myGallery \
   --resource-group myGalleryRG \
   --set description="My updated description."
```

Update the description of an image definition using az sig image-definition update.

Azure CLI

```
az sig image-definition update \
   --gallery-name myGallery\
   --resource-group myGalleryRG \
   --gallery-image-definition myImageDefinition \
   --set description="My updated description."
```

Update an image version to add a region to replicate to using az sig image-version update. This change will take a while as the image gets replicated to the new region.

Azure CLI

```
az sig image-version update \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   --gallery-image-version 1.0.0 \
   --add publishingProfile.targetRegions  name=eastus
```

This example shows how to use az sig image-version update to exclude this image version from being used as the *latest* image.

Azure CLI

```
az sig image-version update \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   --gallery-image-version 1.0.0 \
   --set publishingProfile.excludeFromLatest=true
```

This example shows how to use az sig image-version update to include this image version in being considered for *latest* image.

Azure CLI

```
az sig image-version update \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   --gallery-image-version 1.0.0 \
   --set publishingProfile.excludeFromLatest=false
```

Delete resources

You have to delete resources in reverse order, by deleting the image version first. After you delete all of the image versions, you can delete the image definition. After you delete all image definitions, you can delete the gallery.

CLI

Before you can delete a community shared gallery, you need to use az sig share reset to stop sharing the gallery publicly.

Delete an image version using az sig image-version delete.

Azure CLI

```
az sig image-version delete \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   --gallery-image-version 1.0.0
```

Delete an image definition using az sig image-definition delete.

Azure CLI

```
az sig image-definition delete \
   --resource-group myGalleryRG \
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition
```

Delete a gallery using az sig delete.

Azure CLI

```
az sig delete \
   --resource-group myGalleryRG \
   --gallery-name myGallery
```

Community galleries

ⓘ Important

Azure Compute Gallery – community galleries is currently in PREVIEW and subject to the **Preview Terms for Azure Compute Gallery - community gallery**.

To publish a community gallery, you'll need to **set up preview features in your Azure subscription**. Creating VMs from community gallery images is open to all Azure users.

To list your own galleries, and output the public names for your community galleries:

Azure CLI

```
az sig list --query [*]."
{Name:name,PublicName:sharingProfile.communityGalleryInfo.publicNames}"
```

> **Note**
>
> As an end user, to get the public name of a community gallery, you currently need to use the portal. Go to **Virtual machines** > **Create** > **Azure virtual machine** > **Image** > **See all images** > **Community Images** > **Public gallery name**.

List all of the image definitions that are available in a community gallery using az sig image-definition list-community.

In this example, we list all of the images in the *ContosoImage* gallery in *West US* and by name, the unique ID that is needed to create a VM, OS and OS state.

Azure CLI

```
 az sig image-definition list-community \
   --public-gallery-name "ContosoImages-1a2b3c4d-1234-abcd-1234-
1a2b3c4d5e6f" \
   --location westus \
   --query [*]."{Name:name,ID:uniqueId,OS:osType,State:osState}" -o table
```

List image versions shared in a community gallery using az sig image-version list-community:

Azure CLI

```
az sig image-version list-community \
   --location westus \
   --public-gallery-name "ContosoImages-1a2b3c4d-1234-abcd-1234-
1a2b3c4d5e6f" \
   --gallery-image-definition myImageDefinition \
   --query [*]."{Name:name,UniqueId:uniqueId}" \
   -o table
```

Direct shared galleries

> **Important**

Azure Compute Gallery – direct shared gallery is currently in PREVIEW and subject to the Preview Terms for Azure Compute Gallery.

To publish images to a direct shared gallery during the preview, you need to register at https://aka.ms/directsharedgallery-preview. Creating VMs from a direct shared gallery is open to all Azure users.

During the preview, you need to create a new gallery, with the property `sharingProfile.permissions` set to `Groups`. When using the CLI to create a gallery, use the `--permissions groups` parameter. You can't use an existing gallery, the property can't currently be updated.

To find the `uniqueID` of a gallery that is shared with you, use az sig list-shared. In this example, we are looking for galleries in the West US region.

Azure CLI

```
region=westus
az sig list-shared --location $region --query "[].uniqueId" -o tsv
```

List all of the image definitions that are shared directly with you, use az sig image-definition list-shared.

In this example, we list all of the images in the gallery in *West US* and by name, the unique ID that is needed to create a VM, OS and OS state.

Azure CLI

```
name="1a2b3c4d-1234-abcd-1234-1a2b3c4d5e6f-myDirectShared"
 az sig image-definition list-shared \
   --gallery-unique-name $name
   --location $region \
   --query [*]."{Name:name,ID:uniqueId,OS:osType,State:osState}" -o table
```

List image versions directly shared to you using az sig image-version list-shared:

Azure CLI

```
imgDef="myImageDefinition"
az sig image-version list-shared \
   --location $region \
   --public-gallery-name $name \
   --gallery-image-definition $imgDef \
   --query [*]."{Name:name,UniqueId:uniqueId}" \
   -o table
```

Next steps

- Create an image definition and an image version.
- Create a VM from a generalized or specialized image in an Azure Compute Gallery.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshoot images in an Azure Compute Gallery

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

If you have problems performing any operations on Azure Compute Gallery (formerly known as Shared Image Gallery) resources, like galleries, image definitions, and image versions, run the failing command again in debug mode. You activate debug mode by passing the `--debug` switch with the Azure CLI and the `-Debug` switch with PowerShell. After you've located the error, follow this article to troubleshoot it.

Creating or modifying a gallery

Expand table

Error Message	Cause	Mitigation
Gallery name is invalid. Allowed characters are English alphanumeric characters, with underscores, and periods allowed in the middle, up to 80 characters total. All other special characters, including dashes, are disallowed.	The name for the gallery does not meet the naming requirements	Choose a name that meets the following conditions: • Has an 80-character limit • Contains only English letters, numbers, underscores, and periods • Starts and ends with English letters or numbers
The provided resource name <galleryName> has these invalid trailing characters: <character>. The name can not end with characters: <character>	The name for the gallery ends with a period or underscore	Choose a name for the gallery that meets the following conditions: • Has an 80-character limit • Contains only English letters, numbers, underscores, and periods • Starts and ends with English letters or numbers
The provided location <region> is not available for resource type 'Microsoft.Compute/galleries'. List of available regions for the resource type is ...	The region specified for the gallery is incorrect or requires an access request	Check that the region name is correct. If the region name is correct,

Error Message	Cause	Mitigation
		submit an access request for the region
Can not delete resource before nested resources are deleted	You've tried to delete a gallery that contains at least one existing image definition. A gallery must be empty before it can be deleted	Delete all image definitions inside the gallery and then proceed to delete the gallery. If the image definition contains image versions, you must delete the image versions before you delete the image definitions
The gallery name <galleryName> is not unique within the subscription <subscriptionID>. Please pick another gallery name	You have an existing gallery with the same name and have tried to create another gallery with the same name	Choose a different name for the gallery
The resource <galleryName> already exists in location <region_1> in resource group <resourceGroup>. A resource with the same name cannot be created in location <region_2>. Please select a new resource name	You have an existing gallery with the same name and have tried to create another gallery with the same name	Choose a different name for the gallery

Creating or modifying image definitions

Expand table

Error Message	Cause	Mitigation
Changing property 'galleryImage.properties.<property>' is not allowed	You tried to change the OS type, OS state, Hyper-V generation, offer, publisher, or SKU. Changing any of these properties is not permitted	Create a new image definition instead
The resource <galleryName/imageDefinitionName> already exists in location <region_1> in resource group <resourceGroup>. A resource with the same name cannot be	You have an existing image definition in the same gallery and resource group with the same name. You've tried to create another image definition	Use a different name for the image definition, or put the image definition in a different gallery or resource group

Error Message	Cause	Mitigation
created in location <region_2>. Please select a new resource name.	with the same name and in the same gallery but in a different region.	
The provided resource name <galleryName>/<imageDefinitionName> has these invalid trailing characters: <character>. The name can not end with characters: <character>	The <imageDefinitionName> name ends with a period or underscore.	Choose a name for the image definition that meets the following conditions: • Has an 80-character limit • Contains only English letters, numbers, underscores, hyphens and periods • Starts and ends with English letters or numbers
The entity name <imageDefinitionName> is invalid according to its validation rule: ^[^_\W][\w-._]{0,79}(?<![-.])$"*	The <imageDefinitionName> name ends with a period or underscore.	Choose a name for the image definition that meets the following conditions: • Has an 80-character limit • Contains only English letters, numbers, underscores, hyphens, and periods • Starts and ends with English letters or numbers
Asset name galleryImage.properties.identifier.<property> is not valid. It cannot be empty. Allowed characters are uppercase or lowercase letters, digits, hyphen(-), period (.), underscore (_). Names are not allowed to end with period(.). The length of the name cannot exceed <number> characters.	The publisher, offer, or SKU value does not meet the naming requirements.	Choose a value that meets the following conditions: • Has a 128-character limit for publisher or 64-character limit for offer and SKU • Contains only English letters,

Error Message	Cause	Mitigation
		numbers, hyphens, underscores, and periods • Does not end with a period
Can not perform requested operation on nested resource. Parent resource <galleryName> not found.	There is no gallery with the name <galleryName> in the current subscription and resource group.	Check that the names of the gallery, subscription, and resource group are correct. Otherwise, create a new gallery named <galleryName>
The provided location <region> is not available for resource type 'Microsoft.Compute/galleries'. List of available regions for the resource type is ...	The <region> name is incorrect or requires an access request.	Check that the region name is spelled correctly. You can run this command to see what regions you have access to. If the region is not in the list, submit an access request.
Unable to serialize value: <value> as type: 'iso-8601'., ISO8601Error: ISO 8601 time designator 'T' missing. Unable to parse datetime string <value>	The value provided to the property is not properly formatted as a date.	Provide a date in the yyyy-MM-dd, yyyy-MM-dd'T'HH:mm:sszzz, or ISO 8601 -valid format.
Could not convert string to DateTimeOffset: <value>. Path 'properties.<property>'	The value provided to the property is not properly formatted as a date.	Provide a date in the yyyy-MM-dd, yyyy-MM-dd'T'HH:mm:sszzz, or ISO 8601 -valid format
EndOfLifeDate must be set to a future date.	The end-of-life date property is not properly formatted as a date that's after today's date.	Provide a date in the yyyy-MM-dd, yyyy-MM-dd'T'HH:mm:sszzz, or ISO 8601 -valid format

Error Message	Cause	Mitigation
argument --<property>: invalid int value: <value>	<property> accepts only integer values, and <value> is not an integer.	Choose an integer value.
The minimum value of <property> must not be greater than the maximum value of <property>.	The minimum value provided for <property> is higher than the maximum value provided for <property>.	Change the values so that the minimum is less than or equal to the maximum.
Gallery image: <imageDefinitionName> identified by (publisher:<Publisher>, offer: <Offer>, sku:<SKU>) already exists. Choose a different publisher, offer, sku combination.	You've tried to create a new image definition with the same publisher, offer, and SKU triplet as an existing image definition in the same gallery.	Within a gallery, all image definitions must have a unique combination of publisher, offer, and SKU. Choose a unique combination, or choose a new gallery and create the image definition again
Can not delete resource before nested resources are deleted.	You've tried to delete an image definition that contains image versions. An image definition must be empty before it can be deleted.	Delete all image versions inside the image definition and then proceed to delete the image definition
Cannot bind parameter <property>. Cannot convert value <value> to type <propertyType>. Unable to match the identifier name <value> to a valid enumerator name. Specify one of the following enumerator names and try again: <choice_1>, <choice_2>, ...	The property has a restricted list of possible values, and <value> is not one of them	Choose one of the possible <choice> values
Cannot bind parameter <property>. Cannot convert value <value> to type "System.DateTime"	The value provided to the property is not properly formatted as a date.	Provide a date in the yyyy-MM-dd, yyyy-MM-dd'T'HH:mm:sszzz, or ISO 8601 -valid format
Cannot bind parameter <property>. Cannot convert value <value> to type "System.Int32"	<property> accepts only integer values, and <value> is not an integer.	Choose an integer value

Error Message	Cause	Mitigation
ZRS storage account type is not supported in this region.	You've chosen standard zone-redundant storage (ZRS) in a region that does not yet support it	Change the storage account type to **Premium_LRS** or **Standard_LRS**. Check our documentation for the latest list of regions with ZRS preview enabled

Creating or updating image versions

Expand table

Error Message	Cause	Mitigation
The provided location <region> is not available for resource type 'Microsoft.Compute/galleries'. List of available regions for the resource type is	The <region> name is incorrect or requires an access request	Check that the region name is spelled correctly. You can run this command to see what regions you have access to. If the region is not in the list, submit an access request
Can not perform requested operation on nested resource. Parent resource <galleryName/imageDefinitionName> not found	There is no gallery with the name <galleryName/imageDefinitionName> in the current subscription and resource group	Check that the names of the gallery, subscription, and resource group are correct. Otherwise, create a new gallery with the name <galleryName> and/or an image definition named <imageDefinitionName> in the indicated resource group
Cannot bind parameter <property>. Cannot convert value <value> to type "System.DateTime"	The value provided to the property is not properly formatted as a date	Provide a date in the yyyy-MM-dd, yyyy-MM-dd'T'HH:mm:sszzz, or ISO 8601 -valid format
Cannot bind parameter <property>. Cannot convert value <value> to type "System.Int32"	<property> accepts only integer values, and <value> is not an integer	Choose an integer value
Gallery image version publishing profile regions <publishingRegions> must contain the location of image version <sourceRegion>	The location of the source image (<sourceRegion>) must be included in the <publishingRegions> list	Include <sourceRegion> in the <publishingRegions> list

Error Message	Cause	Mitigation
The value <value> of parameter <property> is out of range. The value must be between <minValue> and <maxValue>, inclusive	<value> is outside the range of possible values for <property>	Choose a value that's within the range of <minValue> and <maxValue>, inclusive
Source <resourceID> is not found. Please check source exists, and is in the same region as gallery image version being created	There is no source located at <resourceID>, or the source at <resourceID> is not in the same region as the gallery image being created	Check that the <resourceID> value is correct and that the source region of the gallery image version is the same as the region of the <resourceID> value
Changing property 'galleryImageVersion.properties.storageProfile.<diskImage>.source.id' is not allowed	The source ID of a gallery image version can't be changed after creation	Ensure that the source ID is the same as the existing source ID, change the version number of the image version, or delete the current image version and try again
Duplicated lun numbers have been detected in the input data disks. Lun number must be unique for each data disk	When you're creating an image version by using a list of disks and/or disk snapshots, two or more disks or disk snapshots have the same LUN	Remove or change any duplicate LUNs
Duplicated source ids are found in the input disks. Source id should be unique for each disk	When you're creating an image version by using a list of disks and/or disk snapshots, two or more disks or disk snapshots have the same resource ID	Remove or change any duplicate disk source IDs
Property id <resourceID> at path 'properties.storageProfile.<diskImages>.source.id' is invalid. Expect fully qualified resource Id that start with '/subscriptions/<subscriptionID>' or '/providers/<resourceProviderNamespace>/'	The <resourceID> value is incorrectly formatted	Check that the resource ID is correct
The source id: <resourceID> must either be a managed image, virtual machine or another gallery image version	The <resourceID> value is incorrectly formatted	If you're using a VM, managed image, or gallery image version as the source image, check that the resource ID of the VM, managed image, or gallery image version is correct

Error Message	Cause	Mitigation
The source id: <resourceID> must be a managed disk or snapshot	The <resourceID> value is incorrectly formatted	If you're using disks and/or disk snapshots as sources for the image version, check that the resource IDs of the disks and/or disk snapshots are correct
Cannot create Gallery Image Version from: <resourceID> since the OS State in the parent gallery image (<OsState_1>) is not <OsState_2>	The operating system state (Generalized or Specialized) does not match the operating system state specified in the image definition	Either choose a source based on a VM with the operating system state of <OsState_1> or create a new image definition for VMs based on <OsState_2>
The resource with id '<resourceID>' has a different Hypervisor generation ['<V#_1>'] than the parent gallery image Hypervisor generation ['<V#_2>']	The hypervisor generation of the image version does not match the hypervisor generation specified in the image definition. The image definition operating system is <V#_1>, and the image version operating system is <V#_2>	Either choose a source with the same hypervisor generation as the image definition or create/choose a new image definition that has the same hypervisor generation as the image version
The resource with id '<resourceID>' has a different OS type ['<OsType_1>'] than the parent gallery image OS type generation ['<OsType _2>']	The hypervisor generation of the image version does not match the hypervisor generation specified in the image definition. The image definition operating system is <OsType_1>, and the image version operating system is <OsType_2>	Either choose a source with the same operating system (Linux/Windows) as the image definition or create/choose a new image definition that has the same operating system generation as the image version
Source virtual machine <resourceID> cannot contain an ephemeral OS disk	The source at <resourceID> contains an ephemeral OS disk. The Azure Compute Gallery does not currently support ephemeral OS disks	Choose a different source based on a VM that does not use an ephemeral OS disk
Source virtual machine <resourceID> cannot contain disk ['<diskID>'] stored in an UltraSSD account type	The disk <diskID> is an Ultra SSD disk. The Azure Compute Gallery does not currently support Ultra SSD disks	Use a source that contains only Premium SSD, Standard SSD, and/or Standard HDD managed disks

Error Message	Cause	Mitigation
Source virtual machine <resourceID > must be created from Managed Disks	The virtual machine in <resourceID> uses unmanaged disks	Use a source based on a VM that contains only Premium SSD, Standard SSD, and/or Standard HDD managed disks
Too many requests on source '<resourceID>'. Please reduce the number of requests on the source or wait some time before retrying	The source for this image version is currently being throttled because of too many requests	Try to create the image version later
The disk encryption set '<diskEncryptionSetID>' must be in the same subscription '<subscriptionID>' as the gallery resource	Disk encryption sets can be used only in the same subscription and region in which they were created	Create or use an encryption set in the same subscription and region as the image version
Encrypted source: '<resourceID>' is in a different subscription ID than the current gallery image version subscription '<subscriptionID_1>'. Please retry with an unencrypted source(s) or use the source's subscription '<subcriptionID_2>' to create the gallery image version	The Azure Compute Gallery does not currently support creating image versions in another subscription from another source image if the source image is encrypted	Use an unencrypted source or create the image version in the same subscription as the source
The disk encryption set <diskEncryptionSetID> was not found	The disk encryption might be incorrect	Check that the resource ID of the disk encryption set is correct
The image version name is invalid. The image version name should follow Major(int).Minor(int).Patch(int) format, for e.g: 1.0.0, 2018.12.1 etc	The valid format for an image version is three integers separated by a period. The image version name did not meet the valid format	Use an image version name that follows the format Major(int).Minor(int).Patch(int). For example: 1.0.0. or 2018.12.1
The value of parameter galleryArtifactVersion.properties.publishingProfile.targetRegions.encryption.dataDiskImages.diskEncryptionSetId is invalid	The resource ID of the disk encryption set used on a data disk image uses an invalid format	Ensure that the resource ID of the disk encryption set follows the format /subscriptions/<subscriptionID>/resourceGroups/<resourceGroupName>/providers/Microsoft.Compute/<diskEncryptionSetName>
The value of parameter galleryArtifactVersion.properties.publishingProfile.targetRegions.encryption.osDiskImage.diskEncryptionSetId is invalid	The resource ID of the disk encryption set used on the OS disk image uses an invalid format	Ensure that the resource ID of the disk encryption set follows the format /subscriptions/<subscriptionID>/resourceGroups/<resourceGroupName>/pro

Error Message	Cause	Mitigation
		viders/Microsoft.Compute/<diskEncryptionSetName>
Cannot specify new data disk image encryption lun [<number>] with a disk encryption set in region [<region>] for update gallery image version request. To update this version, remove the new lun. If you need to change the data disk image encryption settings, you must create a new gallery image version with the correct settings.	You added encryption to the data disk of an existing image version. You can't add encryption to an existing image version	Create a new gallery image version or remove the added encryption settings
The gallery artifact version source can only be specified either directly under storageProfile or within individual OS or data disks. One and only one source type (user image, snapshot, disk, virtual machine) can be provided	The source ID is missing	Ensure that the source ID of the source is present
Source was not found: <resourceID>. Please make sure the source exists	The resource ID of the source might be incorrect	Ensure that the resource ID of the source is correct
A disk encryption set is required for disk 'galleryArtifactVersion.properties.publishingProfile.targetRegions.encryption.osDiskImage.diskEncryptionSetId' in target region '<region_1>' since disk encryption set '<diskEncryptionSetID>' is used for the corresponding disk in region '<region_2>'	Encryption has been used on the OS disk in <region_2>, but not in <region_1>	If you're using encryption on the OS disk, use encryption in all regions
A disk encryption set is required for disk 'LUN <number>' in target region '<region_1>' since disk encryption set '<diskEncryptionSetID>' is used for the corresponding disk in region '<region_2>'	Encryption has been used on the data disk at LUN <number> in <region_2>, but not in <region_1>	If you're using encryption on a data disk, use encryption in all regions
An invalid lun [<number>] was specified in encryption.dataDiskImages. The lun must be one of the following values ['0,9']	The LUN specified for the encryption does not match any of the LUNs for disks attached to the VM	Change the LUN in the encryption to the LUN of a data disk present in the VM
Duplicate luns '<number>' were specified in target region '<region>' en	The encryption settings used in <region> specified	Change the LUN in <region> to make sure that all the LUNs

Error Message	Cause	Mitigation
cryption.dataDiskImages	a LUN at least twice	are unique in <region>
OSDiskImage and DataDiskImage cannot point to same blob <sourceID>	The sources for the OS disk and at least one data disk are not unique	Change the source for the OS disk and/or data disks to ensure that the OS disk as well as each data disk is unique
Duplicate regions are not allowed in target publishing regions	A region is listed among the publishing regions more than once	Remove the duplicate region
Adding new Data Disks or changing the LUN of a Data Disk in an existing Image is not allowed	An update call to the image version either contains a new data disk or has a new LUN for a disk	Use the LUNs and data disks of the existing image version
The disk encryption set <diskEncryptionSetID> must be in the same subscription <subscriptionID> as the gallery resource	The Azure Compute Gallery does not currently support using a disk encryption set in a different subscription	Create the image version and disk encryption set in the same subscription
Replication failed in this region due to 'The GalleryImageVersion source resource size 2520 exceeds the max size 2048 supported	A data disk in the source is greater than 2TB	Resize the data disk to under 2 TB
Operation 'Update Gallery Image Version' is not on allowed on <version Number>; since it is marked for deletion. You can only retry the Delete operation (or wait for an ongoing one to complete)	You attempted to update a gallery image version that is in the process of being deleted	Wait for the deletion event to complete and recreate the image version again
Encryption is not supported for source resource '<sourceID>'. Please use a different source resource type which supports encryption or remove encryption properties	Currently the Azure Compute Gallery only supports encryption for VMs, disks, snapshots and managed images. One of the sources provided for the image version is not in the previous list of sources that support encryption	Remove the disk encryption set from the image version and contact the support team.

Creating or updating a VM or scale sets from an image version

Expand table

Error Message	Cause	Mitigation
There is no latest image version exists for "<imageDefinitionResourceID>"	The image definition you used to deploy the virtual machine does not contain any image versions that are included in latest	Ensure that there is at least one image version that has 'Exclude from latest' set to False
The gallery image /subscriptions/<subscriptionID>/resourceGroups/<resourceGroup>/providers/Microsoft.Compute/galleries/<galleryName>/images/<imageName>/versions/<versionNumber> is not available in <region> region. Please contact image owner to replicate to this region, or change your requested region.	The version selected for deployment does not exist or does not have a replica in the indicated region	Ensure that the name of the image resource is correct and that there is at least one replica in the indicated region
The gallery image /subscriptions/<subscriptionID>/resourceGroups/<resourceGroup>/providers/Microsoft.Compute/galleries/<galleryName>/images/<imageName> is not available in <region> region. Please contact image owner to replicate to this region, or change your requested region	The image definition selected for deployment does not have any image versions that are included in latest and also in the indicated region	Ensure that there is at least one image version in the region that has 'Exclude from latest' set to False
The client has permission to perform action 'Microsoft.Compute/galleries/images/versions/read' on scope <resourceID>, however the current tenant <tenantID> is not authorized to access linked subscription <subscriptionID>	The virtual machine or scale set was created through a gallery image in another tenant. You've tried to make a change to the virtual machine or scale set, but you don't have access to the subscription that owns the image	Contact the owner of the subscription of the image version to grant read access to the image version
The gallery image <resourceID> is not available in <region> region. Please contact image owner to replicate to this region, or change your requested region.	The VM is being created in a region that's not among the list of published regions for the gallery image.	Either replicate the image to the region or create a VM in one of the regions in the gallery image's publishing regions
Parameter 'osProfile' is not allowed	Admin username, password, or SSH keys were provided for a VM that was created from a specialized image version	Don't include the admin username, password, or SSH keys if you intend to create a VM from that image. Otherwise, use a generalized ima

Error Message	Cause	Mitigation
		ge version and supply the admin username, password, or SSH keys
Required parameter 'osProfile' is missing (null)	The VM is created from a generalized image, and it's missing the admin username, password, or SSH keys. Because generalized images don't retain the admin username, password, or SSH keys, these fields must be specified during creation of a VM or scale set	Specify the admin username, password, or SSH keys, or use a specialized image version
Cannot update Virtual Machine Scale Set <vmssName> as the current OS state of the VM Scale Set is Generalized which is different from the updated gallery image OS state which is Specialized	The current source image for the scale set is a generalized source image, but it's being updated with a source image that is specialized. The current source image and the new source image for a scale set must be of the same state	To update the scale set, use a generalized image version
Disk encryption set <diskEncryptionSetID> in Azure Compute Gallery <versionID> belongs to subscription <subscriptionID_1> and cannot be used with resource '' in subscription <subscriptionID_2>	The disk encryption set used to encrypt the image version resides in a different subscription than the subscription to host the image version	Use the same subscription for the image version and disk encryption set
The resource with id <vmID> has a different plan ['{"name":"<name>","publisher":"<publisher>","product":"<product>","promotionCode":"<promotionCode>"}'] than the parent gallery image plan ['null']	The parent image definition for the image version being deployed does not have a purchase plan information	Create an image definition with the same purchase plan details from the error message and create the image version within the image definition
The VM or virtual machine scale set creation takes a long time	NA	Verify that the **OSType** of the image version that you're trying to create the VM or virtual machine scale set from has the same **OSType** of the source that you used to create the image version

Creating a disk from an image version

Expand table

Error Message	Cause	Mitigation
The value of parameter imageReference is invalid.	You've tried to export from a gallery Image version to a disk but used a LUN position that does not exist on the image.	Check the image version to see what LUN positions are in use

Sharing resources

The sharing of gallery, image definition, and image version resources across subscriptions is enabled using Azure role-based access control (Azure RBAC).

Replication speed

Use the **--expand ReplicationStatus** flag to check if the replication to all the specified target regions has finished. If not, wait for up to six hours for the job to finish. If it fails, trigger the command again to create and replicate the image version. If there are many target regions that the image version is being replicated to, consider doing the replication in phases.

Azure limits and quotas

Azure limits and quotas apply to all Azure Compute Gallery, image definition, and image version resources. Make sure you're within the limits for your subscriptions.

Next steps

Learn more about Azure Compute Galleries.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an encrypted image version with customer-managed keys

Article • 02/20/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Images in an Azure Compute Gallery (formerly known as Shared Image Gallery) are stored as snapshots. These images are automatically encrypted through server-side 256-bit encryption AES encryption. Server-side encryption is also FIPS 140-2 compliant. For more information about the cryptographic modules underlying Azure managed disks, see Cryptography API: Next Generation.

You can rely on platform-managed keys for the encryption of your images, or use your own keys. You can also use both of these features together for doubled encryption. If you choose to manage encryption with your own keys, you can specify a *customer-managed key* to use for encrypting and decrypting all disks in your images.

Server-side encryption through customer-managed keys uses Azure Key Vault. You can either import your RSA keys to your key vault or generate new RSA keys in Azure Key Vault.

Prerequisites

This article requires that you already have a disk encryption set in each region where you want to replicate your image:

- To use only a customer-managed key, see the articles about enabling customer-managed keys with server-side encryption by using the Azure portal or PowerShell.
- To use both platform-managed and customer-managed keys (for double encryption), see the articles about enabling double encryption at rest by using the Azure portal or PowerShell.

> **Important**
>
> You must use the link https://aka.ms/diskencryptionupdates to access the Azure portal. Double encryption at rest is not currently visible in the public Azure portal unless you use that link.

Limitations

When you're using customer-managed keys for encrypting images in an Azure Compute Gallery, these limitations apply:

- Encryption key sets must be in the same subscription as your image.
- Encryption key sets are regional resources, so each region requires a different encryption key set.
- After you've used your own keys to encrypt an image, you can't go back to using platform-managed keys for encrypting those images.
- ACG Image version source encrypted with CMK cannot be used as source to create another ACG Image version.
- Some of the features like replicating an SSE+CMK image, creating an image from SSE+CMK encrypted disk etc. are not supported through portal.

Creating the Image

PowerShell

To specify a disk encryption set for an image version, use New-AzGalleryImageVersion with the `-TargetRegion` parameter:

Azure PowerShell

```
$sourceId = <ID of the image version source>
$osDiskImageEncryption = @{DiskEncryptionSetId='subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myRG/providers/Microsoft.Compute/diskEncryptionSets/myDESet'}
$dataDiskImageEncryption1 = @{DiskEncryptionSetId='subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myRG/providers/Microsoft.Compute/diskEncryptionSets/myDESet1';Lun=1}
$dataDiskImageEncryption2 = @{DiskEncryptionSetId='subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myRG/providers/Microsoft.Compute/diskEncryptionSets/myDESet2';Lun=2}
$dataDiskImageEncryptions = @($dataDiskImageEncryption1,$dataDiskImageEncryption2)
$encryption1 = @{OSDiskImage=$osDiskImageEncryption;DataDiskImages=$dataDiskImageEncryptions}
$region1 = @{Name='West
```

```
US';ReplicaCount=1;StorageAccountType=Standard_LRS;Encryption=$encryptio
n1}
$eastUS2osDiskImageEncryption =
@{DiskEncryptionSetId='subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myRG/providers/Microsoft.Compute/diskEncrypt
ionSets/myEastUS2DESet'}
$eastUS2dataDiskImageEncryption1 =
@{DiskEncryptionSetId='subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myRG/providers/Microsoft.Compute/diskEncrypt
ionSets/myEastUS2DESet1';Lun=1}
$eastUS2dataDiskImageEncryption2 =
@{DiskEncryptionSetId='subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myRG/providers/Microsoft.Compute/diskEncrypt
ionSets/myEastUS2DESet2';Lun=2}
$eastUS2DataDiskImageEncryptions =
@($eastUS2dataDiskImageEncryption1,$eastUS2dataDiskImageEncryption2)
$encryption2 =
@{OSDiskImage=$eastUS2osDiskImageEncryption;DataDiskImages=$eastUS2DataD
iskImageEncryptions}
$region2 = @{Name='East US
2';ReplicaCount=1;StorageAccountType=Standard_LRS;Encryption=$encryption
2}
$targetRegion = @($region1, $region2)
```

Create the image

```
New-AzGalleryImageVersion `
   -ResourceGroupName $rgname `
   -GalleryName $galleryName `
   -GalleryImageDefinitionName $imageDefinitionName `
   -Name $versionName -Location $location `
   -SourceImageId $sourceId `
   -ReplicaCount 2 `
   -StorageAccountType Standard_LRS `
   -PublishingProfileEndOfLifeDate '2020-12-01' `
   -TargetRegion $targetRegion
```

Create the VM

You can create a virtual machine (VM) from an Azure Compute Gallery and use customer-managed keys to encrypt the disks. The syntax is the same as creating a generalized or specialized VM from an image. Use the extended parameter set and add `Set-AzVMOSDisk -Name $($vmName +"_OSDisk") -DiskEncryptionSetId $diskEncryptionSet.Id -CreateOption FromImage` to the VM configuration.

For data disks, add the `-DiskEncryptionSetId $setID` parameter when you use Add-AzVMDataDisk.

Next steps

Learn more about server-side disk encryption.

For information about how to supply purchase plan information, see Supply Azure Marketplace purchase plan information when creating images.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Export an image version to a managed disk

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can export an image version's OS or data disk as a managed disk from an image version, which is stored in an Azure Compute Gallery (formerly known as Shared Image Gallery).

CLI

List the image versions in a gallery using az sig image-version list. In this example, we're looking for all of the image versions that are part of the *myImageDefinition* image definition in the *myGallery* gallery.

Azure CLI

```
az sig image-version list \
   --resource-group myResourceGroup\
   --gallery-name myGallery \
   --gallery-image-definition myImageDefinition \
   -o table
```

Set the `source` variable to the ID of the image version, then use az disk create to create the managed disk.

In this example, we export the OS disk of the image version to create a managed disk named *myManagedOSDisk*, in the *EastUS* region, in a resource group named *myResourceGroup*.

Azure CLI

```
source="/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Compute/galleries/<galleryName>/images/<galleryImageDefinition>/versions/<imageVersion>"

az disk create --resource-group myResourceGroup --location EastUS --name myManagedOSDisk --gallery-image-reference $source
```

If you want to export a data disk from the image version, add `--gallery-image-reference-lun` to specify the LUN location of the data disk to be exported.

In this example, we export the data disk located at LUN 0 of the image version to create a managed disk named *myManagedDataDisk*, in the *EastUS* region, in a resource group named *myResourceGroup*.

Azure CLI

```
source="/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Compute/galleries/<galleryName>/images/<galleryImageDefinition>/versions/<imageVersion>"

az disk create --resource-group myResourceGroup --location EastUS --name myManagedDataDisk --gallery-image-reference $source --gallery-image-reference-lun 0
```

PowerShell

List the image versions in a gallery using Get-AzResource.

Azure PowerShell

```
Get-AzResource `
   -ResourceType Microsoft.Compute/galleries/images/versions | `
   Format-Table -Property Name,ResourceId,ResourceGroupName
```

Once you have all of the information you need, you can use Get-AzGalleryImageVersion to get the source image version you want to use and assign it to a variable. In this example, we're getting the `1.0.0` image version, of the `myImageDefinition` definition, in the `myGallery` source gallery, in the `myResourceGroup` resource group.

Azure PowerShell

```
$sourceImgVer = Get-AzGalleryImageVersion `
   -GalleryImageDefinitionName myImageDefinition `
   -GalleryName myGallery `
   -ResourceGroupName myResourceGroup `
   -Name 1.0.0
```

After setting the `source` variable to the ID of the image version, use New-AzDiskConfig to create a disk configuration, then New-AzDisk to create the disk.

In this example, we export the OS disk of the image version to create a managed disk named *myManagedOSDisk*, in the *EastUS* region, in a resource group named *myResourceGroup*.

Create a disk configuration.

Azure PowerShell

```
$diskConfig = New-AzDiskConfig `
   -Location EastUS `
   -CreateOption FromImage `
   -GalleryImageReference @{Id = $sourceImgVer.Id}
```

Create the disk.

Azure PowerShell

```
New-AzDisk -Disk $diskConfig `
   -ResourceGroupName myResourceGroup `
   -DiskName myManagedOSDisk
```

If you want to export a data disk on the image version, add a LUN ID to the disk configuration to specify the LUN location of the data disk to export.

In this example, we export the data disk located at LUN 0 of the image version to create a managed disk named *myManagedDataDisk*, in the *EastUS* region, in a resource group named *myResourceGroup*.

Create a disk configuration.

Azure PowerShell

```
$diskConfig = New-AzDiskConfig `
   -Location EastUS `
   -CreateOption FromImage `
   -GalleryImageReference @{Id = $sourceImgVer.Id; Lun=0}
```

Create the disk.

Azure PowerShell

```
New-AzDisk -Disk $diskConfig `
   -ResourceGroupName myResourceGroup `
   -DiskName myManagedDataDisk
```

Next steps

You can also create an image version from a managed disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate Managed image to Azure compute gallery image version

Applies to: ✔ Linux Virtual Machine ✔ Windows Virtual Machine ✔ Virtual Machine Flex Scale Sets

Managed images is legacy method to generalize and capture Virtual Machine image. For the most current technology, customers are encouraged to use Azure compute gallery. All new features, like Arm64, Trusted launch, and Confidential Virtual Machine are only supported through Azure compute gallery. If you have an existing managed image, you can use it as a source and create an Azure compute gallery image.

Before you begin

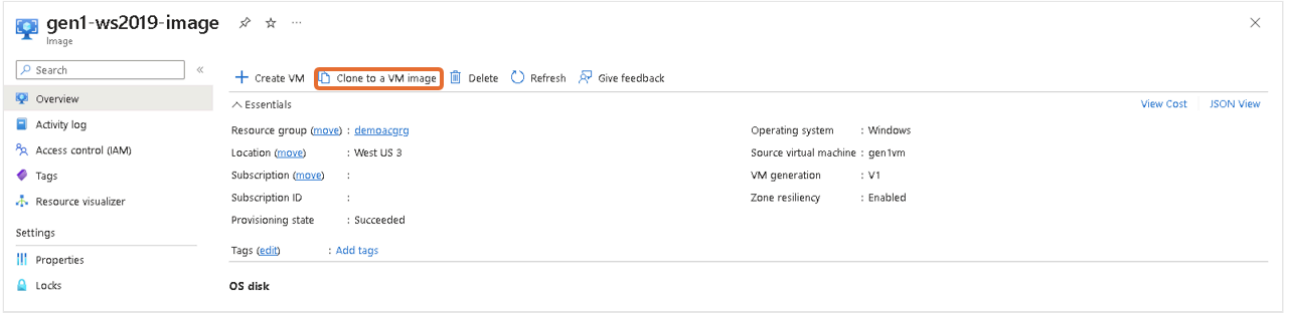
- Screenshot that shows the creation of an Azure Compute Gallery..
- Assign `Reader` permission on source managed image.
- Assign `Contributor` permission on target Azure compute gallery image definition.

Migrate Managed image to Azure Compute Gallery image

Portal

This section steps through using the Azure portal to migrate Managed image to existing Azure Compute Gallery.

1. Sign-in to Azure portal.
2. Navigate to `Managed image` to be migrated and select **Clone to a VM image**.



3. Provide following details on the `Create VM image version` page and select **Next : Replication >**:

   - Version number

- Target Azure compute gallery
- Target Virtual Machine image definition

4. Select following `Replication` configuration or select default values and select **Next : Encryption >**:

   - Default storage sku
   - Default replication count
   - Target replication regions

5. Select `SSE encryption type` or select default `Encryption at-rest with a platform managed key` and select **Next : Tags >**.

6. *Optional* Assign resource tags.

7. Validate the configuration on `Review + Create` page and select **Create** to complete the migration.

Next steps

Replace managed image reference with Azure compute gallery image version in the Virtual machine and flex scale sets deployment templates.

Last updated on 08/24/2025

Supply Azure Marketplace purchase plan information when creating images

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

If you're creating an image in a shared gallery using an Azure Marketplace image source, you'll likely need to track purchase plan information. This article shows you how to find purchase plan information for a VM when creating an image definition. We also cover using this information to simplify supplying the purchase plan information when creating a VM for an image.

For more information about finding and using Marketplace images, see Find and use Azure Marketplace images.

Get the source VM information

If you still have the original VM, you can get the plan name, publisher, and product information from it using Get-AzVM. This example gets a VM named *myVM* in the *myResourceGroup* resource group and then displays the purchase plan information for the VM.

Azure PowerShell

```
$vm = Get-azvm `
   -ResourceGroupName myResourceGroup `
   -Name myVM
$vm.Plan
```

Create the image definition

Get the gallery you want to use to store the image. You can list all of the galleries first.

Azure PowerShell

```
Get-AzResource -ResourceType Microsoft.Compute/galleries | Format-Table
```

Then create variables for the gallery you want to use. In this example, we're creating a variable named `$gallery` for *myGallery* in the *myGalleryRG* resource group.

Azure PowerShell

```
$gallery = Get-AzGallery `
   -Name myGallery `
   -ResourceGroupName myGalleryRG
```

Create the image definition, using the `-PurchasePlanPublisher`, `-PurchasePlanProduct`, and `-PurchasePlanName` parameters.

Azure PowerShell

```
 $imageDefinition = New-AzGalleryImageDefinition `
   -GalleryName $gallery.Name `
   -ResourceGroupName $gallery.ResourceGroupName `
   -Location $gallery.Location `
   -Name 'myImageDefinition' `
   -OsState specialized `
   -OsType Linux `
   -Publisher 'myPublisher' `
   -Offer 'myOffer' `
   -Sku 'mySKU' `
   -PurchasePlanPublisher $vm.Plan.Publisher `
   -PurchasePlanProduct $vm.Plan.Product `
   -PurchasePlanName  $vm.Plan.Name
```

Then create your image version using New-AzGalleryImageVersion.

Create the VM

When you go to create a VM from the image, you can use the information from the image definition to pass in the publisher information using Set-AzVMPlan.

Azure PowerShell

```
# Create some variables for the new VM.
$resourceGroup = "mySIGPubVM"
$location = "West Central US"
$vmName = "mySIGPubVM"

# Create a resource group
New-AzResourceGroup -Name $resourceGroup -Location $location

# Create the network resources.
$subnetConfig = New-AzVirtualNetworkSubnetConfig `
   -Name mySubnet `
   -AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork `
   -ResourceGroupName $resourceGroup `
```

```
    -Location $location `
    -Name MYvNET `
    -AddressPrefix 192.168.0.0/16 `
    -Subnet $subnetConfig
$pip = New-AzPublicIpAddress `
    -ResourceGroupName $resourceGroup `
    -Location $location `
   -Name "mypublicdns$(Get-Random)" `
   -AllocationMethod Static `
   -IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig `
    -Name myNetworkSecurityGroupRuleRDP  `
    -Protocol Tcp `
    -Direction Inbound `
    -Priority 1000 `
    -SourceAddressPrefix * `
    -SourcePortRange * `
    -DestinationAddressPrefix * `
    -DestinationPortRange 3389 -Access Deny
$nsg = New-AzNetworkSecurityGroup `
    -ResourceGroupName $resourceGroup `
    -Location $location `
    -Name myNetworkSecurityGroup `
    -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface `
    -Name $vmName `
    -ResourceGroupName $resourceGroup `
    -Location $location `
   -SubnetId $vnet.Subnets[0].Id `
   -PublicIpAddressId $pip.Id `
   -NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration using Set-AzVMSourceImage -Id
$imageDefinition.Id to use the latest available image version. Set-AZVMPlan
is used to pass the plan information in for the VM.

$vmConfig = New-AzVMConfig `
    -VMName $vmName `
    -VMSize Standard_D1_v2   | `
    Set-AzVMSourceImage -Id $imageDefinition.Id | `
    Set-AzVMPlan `
      -Publisher $imageDefinition.PurchasePlan.Publisher `
      -Product $imageDefinition.PurchasePlan.Product `
      -Name $imageDefinition.PurchasePlan.Name | `
    Add-AzVMNetworkInterface -Id $nic.Id

# Create the virtual machine
New-AzVM `
    -ResourceGroupName $resourceGroup `
    -Location $location `
    -VM $vmConfig
```

ARM

```
# Create VM using Plan information through JSON
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",

  "parameters": {
    "adminUsername": {
      "type": "string",
      "metadata": {
        "description": "Username for the Virtual Machine."
      }
    },
    "adminPassword": {
      "type": "securestring",
      "metadata": {
        "description": "Password for the Virtual Machine."
      }
    },

    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Location for all resources."
      }
    }
  },
  "variables": {
    "nicName": "myVMNic",
    "addressPrefix": "10.0.0.0/16",
    "subnetName": "Subnet",
    "subnetPrefix": "10.0.0.0/24",
    "subnetRef": "[resourceId('Microsoft.Network/virtualNetworks/subnets',
variables('virtualNetworkName'), variables('subnetName'))]",
    "vmName": "perflabwin10",
    "virtualNetworkName": "MyVNET",
    "publicIPAddressName": "myPublicIP",
    "dnsNameForPublicIP": "[uniqueString(resourceGroup().id)]",
    "networkSecurityGroupName": "default-NSG"
  },
  "resources": [
    {
      "apiVersion": "2017-06-01",
      "type": "Microsoft.Network/publicIPAddresses",
      "name": "[variables('publicIPAddressName')]",
      "location": "[parameters('location')]",
      "properties": {
        "publicIPAllocationMethod": "Dynamic",
        "dnsSettings": {
          "domainNameLabel": "[variables('dnsNameForPublicIP')]"
        }
      }
    },
```

```
    {
      "comments": "Default Network Security Group for template",
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2019-08-01",
      "name": "[variables('networkSecurityGroupName')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "default-allow-3389",
            "properties": {
              "priority": 1000,
              "access": "Allow",
              "direction": "Inbound",
              "destinationPortRange": "3389",
              "protocol": "Tcp",
              "sourceAddressPrefix": "*",
              "sourcePortRange": "*",
              "destinationAddressPrefix": "*"
            }
          }
        ]
      }
    },
    {
      "apiVersion": "2018-04-01",
      "type": "Microsoft.Network/virtualNetworks",
      "name": "[variables('virtualNetworkName')]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
      ],
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets": [
          {
            "name": "[variables('subnetName')]",
            "properties": {
              "addressPrefix": "[variables('subnetPrefix')]",
              "networkSecurityGroup": {
                "id": "
[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
              }
            }
          }
        ]
      }
    },
    {
```

```
      "apiVersion": "2018-04-01",
      "type": "Microsoft.Network/networkInterfaces",
      "name": "[variables('nicName')]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[variables('publicIPAddressName')]",
        "[variables('virtualNetworkName')]"
      ],
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": {
                "id": "
[resourceId('Microsoft.Network/publicIPAddresses',variables('publicIPAddress
Name'))]"
              },
              "subnet": {
                "id": "[variables('subnetRef')]"
              }
            }
          }
        ]
      }
    },
    {
      "apiVersion": "2018-04-01",
      "type": "Microsoft.Compute/virtualMachines",
      "name": "[variables('vmName')]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[variables('nicName')]"
      ],
                 "plan": {
          "name": "xxxx",
          "publisher": "xxxx",
          "product": "xxxx"
      },
      "properties": {
        "hardwareProfile": {
          "vmSize": "Standard_D4s_v3"
        },
        "osProfile": {
          "computerName": "[variables('vmName')]",
          "adminUsername": "[parameters('adminUsername')]",
          "adminPassword": "[parameters('adminPassword')]"
        },
        "storageProfile": {
          "imageReference": {
          "id": "<Provide Image URI>"

          },
          "osDisk": {
```

```
            "createOption": "FromImage"
          }
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "
[resourceId('Microsoft.Network/networkInterfaces',variables('nicName'))]"
            }
          ]
        }
      }

    }
  ]

}
```

Next steps

For more information about finding and using Marketplace images, see Find and use Azure Marketplace images.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deprecated images FAQ

This article answers common questions about what happens when Azure Marketplace images are deprecated.

Overview

When an Azure Marketplace image is deprecated, its availability for new deployments changes—but existing virtual machines (VMs) and virtual machine scale set (VMSS) instances continue running normally. This article explains platform behavior during image deprecation, which operations are impacted, and what actions (if any) you must take.

What image deprecation means

Deprecation removes the image from the Azure Marketplace catalog but doesn't remove OS disks of existing VMs.

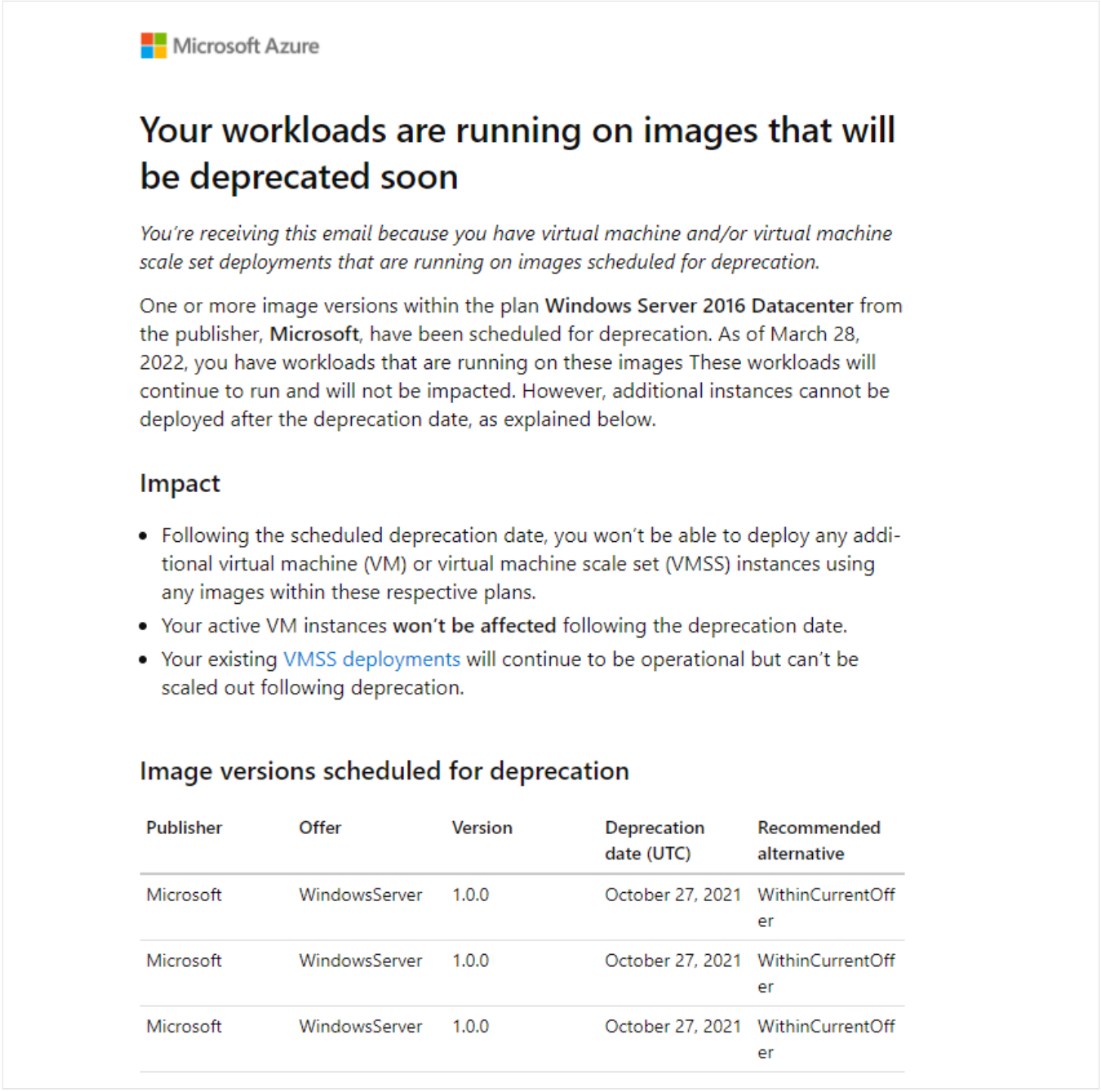
- You can't use a deprecated image for new VM or VMSS instance creation after the enforcement date.
- Existing VMs and VMSS instances that were created before the enforcement date continue operating normally.
- VMSS scale-out operations are blocked after the enforcement date because scaling out requires creating new instances from the deprecated image.

Key points

- **No action required for existing VMs:** Once created, VMs boot from their OS disk—not from the original image. Deprecation doesn't affect running VMs.
- **Most operations continue working:** Start, stop, restart, deallocate, and redeploy all work normally.
- **Reimage is blocked:** This is the only VM operation affected because it requires recreating the OS disk from the source image.
- **VMSS instances continue running:** Existing instances aren't affected, but scale-out is blocked after enforcement.

What does it mean when I receive an image deprecation notice?

You receive an email notification when your virtual machine (VM) or virtual machine scale set runs on an image that's scheduled for deprecation. Publishers deprecate images for various reasons, including security concerns or when images reach end of life.



Microsoft Azure

Your workloads are running on images that will be deprecated soon

You're receiving this email because you have virtual machine and/or virtual machine scale set deployments that are running on images scheduled for deprecation.

One or more image versions within the plan **Windows Server 2016 Datacenter** from the publisher, **Microsoft**, have been scheduled for deprecation. As of March 28, 2022, you have workloads that are running on these images These workloads will continue to run and will not be impacted. However, additional instances cannot be deployed after the deprecation date, as explained below.

Impact

- Following the scheduled deprecation date, you won't be able to deploy any additional virtual machine (VM) or virtual machine scale set (VMSS) instances using any images within these respective plans.
- Your active VM instances **won't be affected** following the deprecation date.
- Your existing VMSS deployments will continue to be operational but can't be scaled out following deprecation.

Image versions scheduled for deprecation

Publisher	Offer	Version	Deprecation date (UTC)	Recommended alternative
Microsoft	WindowsServer	1.0.0	October 27, 2021	WithinCurrentOffer
Microsoft	WindowsServer	1.0.0	October 27, 2021	WithinCurrentOffer
Microsoft	WindowsServer	1.0.0	October 27, 2021	WithinCurrentOffer

Deprecation levels

Deprecation occurs at three levels:

- **Image version deprecation**: A specific VM image version.
- **Plan/SKU deprecation**: An entire plan or SKU and all image versions within it.
- **Offer deprecation**: The entire VM offer, including all plans and image versions.

What is the deprecation timeline and how does it affect my workloads?

Notice period (90 days before enforcement)

During the notice period:

- You receive email notifications and Azure Advisor recommendations about the upcoming deprecation.
- Existing VMs and virtual machine scale set instances continue to run normally.
- You can deploy new instances by using Azure CLI, PowerShell, or APIs.
- Deprecation warnings appear in the Azure portal, Azure CLI, and APIs.
- You can migrate or update your deployments until the enforcement date.

After enforcement date (soft delete)

After the enforcement date:

- New VM or virtual machine scale set deployments that use deprecated images are blocked.
- Existing VMs and virtual machine scale set instances continue to run without disruption.
- Virtual machine scale set scale-out is blocked because it requires creating new instances from the deprecated image.
- Most VM operations (start, stop, restart, deallocate, redeploy) continue working. See Which operations are affected by image deprecation? for details.

Required actions

To avoid disruption:

- Migrate your virtual machine scale set to a supported image before the enforcement date if you need your virtual machine scale set to scale out after deprecation.
- If you configured automatic OS upgrades, only offer or plan deprecation affects you, not individual version deprecation.

Which operations are affected by image deprecation?

VM operations

After you create a VM, Azure creates an OS disk, and the VM boots exclusively from that disk—not from the original image reference. A deprecated or removed image doesn't affect an

already-running VM.

Expand table

Operation	Before enforcement	After enforcement	Explanation
Reimage	Works	Blocked	Reimage requires Azure to create a new OS disk from the source image. If the image is deprecated or removed, reimage fails.
Redeploy	Works	Works	Redeploy moves the VM to new hardware but doesn't recreate the OS disk.
Start / Restart	Works	Works	These operations don't require the source image.
Deallocate / Stop	Works	Works	These operations don't require the source image.

> **Note**
>
> The VM's image reference property might disappear from the Azure portal, but this doesn't affect VM operation. The OS disk is authoritative for boot.

Virtual machine scale set operations

Existing virtual machine scale set instances continue running normally even after the base image is deprecated.

Expand table

Operation	Before enforcement	After enforcement
Existing instances	Continue running	Continue running
Scale-out	Allowed	Blocked

Operation	Before enforcement	After enforcement
Reimage	Works	Blocked

Action required: None for existing workloads. Migrate to a supported image before the enforcement date if you need to scale out your virtual machine scale set after deprecation.

How do I find VMs using deprecated images?

Option 1: Use automated script (recommended)

Run Get-AzVMImageDeprecationStatus.ps1 to list all VMs in a subscription that were created from deprecated or soon-to-be-deprecated images.

Option 2: Use Azure Resource Graph

Run the following query in Azure Resource Graph to list all VMs and their image details:

Kusto

```
resources
| where type == "microsoft.compute/virtualmachines"
//| where properties.storageProfile.imageReference.publisher =~
'MicrosoftWindowsServer' //Optional: Filter by publisher
//| where properties.storageProfile.imageReference.sku =~ '2016-Datacenter'
//Optional: Filter by SKU (Plan)
//| where properties.storageProfile.imageReference.version ==
'14393.4467.2106061537' //Optional: Filter by version
| project name, subscriptionId, resourceGroup,
ImagePublisher=properties.storageProfile.imageReference.publisher,
ImageOffer=properties.storageProfile.imageReference.offer,
imageSku=properties.storageProfile.imageReference.sku,
imageVersion=properties.storageProfile.imageReference.exactVersion
```

Option 3: Use Azure CLI

To list all VMs using a specific SKU (replace `2016-Datacenter` with your deprecated SKU):

Azure CLI

```
az vm list --query "[?storageProfile.imageReference.sku=='2016-Datacenter'].{VM:id,
imageOffer:storageProfile.imageReference.offer,
imagePublisher:StorageProfile.imageReference.publisher, imageSku:
```

```
storageProfile.imageReference.sku,
imageVersion:storageProfile.imageReference.version}"
```

To list all VMs using a specific image version (replace with your deprecated version):

Azure CLI

```
az vm list --query "[?
storageProfile.imageReference.version=='14393.4402.2105052108'].{VM:id,
imageOffer:storageProfile.imageReference.offer,
imagePublisher:StorageProfile.imageReference.publisher, imageSku:
storageProfile.imageReference.sku,
imageVersion:storageProfile.imageReference.version}"
```

Option 4: Use PowerShell

To list all VMs using a specific SKU (replace `2016-Datacenter` with your deprecated SKU):

PowerShell

```
Get-AzVM | Where-Object {$_.StorageProfile.ImageReference.Sku -eq '2016-
Datacenter'} | Select-Object -Property ResourceGroupName, Name,
@{label='imageOffer'; expression={$_.StorageProfile.ImageReference.Offer}},
@{label='imagePublisher'; expression={$_.StorageProfile.ImageReference.Publisher}},
@{label='imageSKU'; expression={$_.StorageProfile.ImageReference.Sku}},
@{label='imageVersion'; expression={$_.StorageProfile.ImageReference.ExactVersion}}
```

To list all VMs using a specific image version (replace with your deprecated version):

PowerShell

```
Get-AzVM | Where-Object {$_.StorageProfile.ImageReference.ExactVersion -eq
'14393.4402.2105052108'} | Select-Object -Property ResourceGroupName, Name,
@{label='imageOffer'; expression={$_.StorageProfile.ImageReference.Offer}},
@{label='imagePublisher'; expression={$_.StorageProfile.ImageReference.Publisher}},
@{label='imageSKU'; expression={$_.StorageProfile.ImageReference.Sku}},
@{label='imageVersion'; expression={$_.StorageProfile.ImageReference.ExactVersion}}
```

How do I find virtual machine scale sets using deprecated images?

Option 1: Use automated script (recommended)

Run Get-AzVMImageDeprecationStatus.ps1 to list all virtual machine scale sets in a subscription that were created from deprecated or soon-to-be-deprecated images.

Option 2: Use Azure Resource Graph

Run the following query in Azure Resource Graph to list all virtual machine scale sets and their image details:

Kusto

```
resources
| where type == "microsoft.compute/virtualmachinescalesets"
//| where properties.virtualMachineProfile.storageProfile.imageReference.publisher
=~ 'MicrosoftWindowsServer' //Optional: Filter by publisher
//| where properties.virtualMachineProfile.storageProfile.imageReference.sku =~
'2016-Datacenter' //Optional: Filter by SKU (Plan)
//| where properties.virtualMachineProfile.storageProfile.imageReference.version ==
'14393.4467.2106061537' //Optional: Filter by version
//| where properties.virtualMachineProfile.storageProfile.imageReference.version !=
"latest" //Optional: Exclude scale sets using "latest"
| project name, subscriptionId, resourceGroup,
ImagePublisher=properties.virtualMachineProfile.storageProfile.imageReference.publi
sher,
ImageOffer=properties.virtualMachineProfile.storageProfile.imageReference.offer,
imageSku=properties.virtualMachineProfile.storageProfile.imageReference.sku,
imageVersion=properties.virtualMachineProfile.storageProfile.imageReference.version
```

Option 3: Use Azure CLI

To list all virtual machine scale sets using a specific SKU (replace `2016-Datacenter` with your deprecated SKU):

Azure CLI

```
az vmss list --query "[?
virtualMachineProfile.storageProfile.imageReference.sku=='2016-Datacenter'].
{VMSS:id, imageOffer:virtualMachineProfile.storageProfile.imageReference.offer,
imagePublisher:virtualMachineProfile.storageProfile.imageReference.publisher,
imageSku:virtualMachineProfile.storageProfile.imageReference.sku,
imageVersion:virtualMachineProfile.storageProfile.imageReference.version}"
```

To list all virtual machine scale sets using a specific image version (replace with your deprecated version):

Azure CLI

```
az vmss list --query "[?
virtualMachineProfile.storageProfile.imageReference.version=='14393.4402.2105052108
'].{VMSS:id, imageOffer:virtualMachineProfile.storageProfile.imageReference.offer,
imagePublisher:virtualMachineProfile.storageProfile.imageReference.publisher,
imageSku:virtualMachineProfile.storageProfile.imageReference.sku,
imageVersion:virtualMachineProfile.storageProfile.imageReference.version}"
```

Option 4: Use PowerShell

To find virtual machine scale sets using a specific SKU (replace `2016-Datacenter` with your deprecated SKU):

PowerShell

```
$vmsslist = Get-AzVmss
$vmsslist | Where-Object
{$_.virtualMachineProfile.storageProfile.imageReference.Sku -eq '2016-Datacenter'}
| Select-Object -Property ResourceGroupName, Name, @{label='imageOffer';
expression={$_.virtualMachineProfile.storageProfile.imageReference.Offer}},
@{label='imagePublisher'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Publisher}},
@{label='imageSKU'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Sku}},
@{label='imageVersion'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Version}}
```

To find virtual machine scale sets using a specific image version (replace with your deprecated version):

PowerShell

```
$vmsslist = Get-AzVmss
$vmsslist | Where-Object
{$_.virtualMachineProfile.storageProfile.imageReference.Version -eq
'14393.4402.2105052108'} | Select-Object -Property ResourceGroupName, Name,
@{label='imageOffer'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Offer}},
@{label='imagePublisher'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Publisher}},
@{label='imageSKU'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Sku}},
@{label='imageVersion'; expression=
{$_.virtualMachineProfile.storageProfile.imageReference.Version}}
```

When should I migrate my VMs or virtual machine scale sets to a different image?

Migration isn't always required. Review the following scenarios to determine whether you need to take action:

Migration required

- **Virtual machine scale sets that need to scale out:** Scale-out operations are blocked after the enforcement date because they require creating new instances from the deprecated

image. Migrate before enforcement if you need scale-out capability.

- **Workloads that require reimage operations:** Reimage recreates the OS disk from the source image, which fails after the image is deprecated.
- **Disaster recovery scenarios for non-endorsed images:** Azure Site Recovery and backup restore operations that rely on deprecated non-Microsoft-endorsed images don't work after enforcement. Microsoft-endorsed Linux distributions (like Canonical Ubuntu, Red Hat, SUSE) and Windows Server images aren't affected—Azure Site Recovery and backup operations continue to work normally for these images even after deprecation.
- **Images with plan info:** After deprecation, you can't create new workloads or perform restore operations because of missing purchase plan information. Migrate to a supported image.

Migration not required

- **Running VMs that don't need reimage:** After creation, VMs boot from their OS disk—not from the original image. Deprecation doesn't affect already-running VMs.
- **Virtual machine scale sets that don't scale out:** Existing instances continue to run normally.
- **Routine operations:** Start, stop, restart, deallocate, and redeploy all continue to work after enforcement.
- **Virtual machine scale sets using automatic OS upgrades:** If you configured automatic OS upgrades and only the image version is deprecated (not the entire offer or plan), you don't need to take action.

Recommended actions

- Review which VMs and virtual machine scale sets use the deprecated image. See How do I find VMs using deprecated images? and How do I find virtual machine scale sets using deprecated images?.
- Evaluate whether your workloads require any blocked operations after enforcement.
- If migration is required, plan and test before the enforcement date.

How do I migrate a VM to a different image?

Azure VMs don't support in-place OS upgrades. Create a new VM with a supported image, migrate your data, and decommission the old VM.

> **Important**

For deprecated paid Marketplace images, you can't create new VMs or perform backup restore operations after deprecation because of missing plan information. Migrate before the enforcement date.

Part 1: Create a new VM

1. **Identify the replacement image**: Check your deprecation email for the recommended replacement. For more information, see Find and use images using the Azure CLI or Find and use images using Azure PowerShell.
2. **Configure the new VM**: Match the region and networking settings of your existing VM for the simplest migration. You can change VM size or region if needed.
3. **Deploy and test**: Confirm the VM starts correctly and your application dependencies are present.

Part 2: Migrate data

Method A: Detach and attach data disks (recommended)

If your workload data is stored on data disks (not the OS disk):

1. Stop and deallocate the old VM.
2. Detach the data disks from the old VM. For more information, see Detach a data disk from a Linux VM or Detach a data disk from a Windows VM.
3. Attach the data disks to the new VM. For more information, see Attach a data disk to a Linux VM or Attach a data disk to a Windows VM.
4. Start the new VM and verify data accessibility.

> **Note**
>
> Data on the OS disk isn't preserved with this method. Only data on attached data disks is migrated.

Method B: Use snapshots (for OS disk data or backup requirements)

If you need to preserve OS disk data or create a backup before migration:

1. Create a snapshot of the data disks.
2. Create managed disks from the snapshots.
3. Attach the disks to the new VM. For more information, see Attach a data disk to a Linux VM or Attach a data disk to a Windows VM.

Method C: Use file-based transfer

Use OS transfer tools for selective data migration:

- **Windows**: Robocopy or SMB share
- **Linux**: rsync, scp, or sftp
- **Cross-OS**: AzCopy with Azure Storage as an intermediate location

Part 3: Switch over

1. **Validate**: Verify network connectivity, application functionality, firewall rules, and extensions.
2. **Update references**: Point DNS names, load balancers, or API endpoints to the new VM.
3. **Decommission**: Keep the old VM briefly for rollback, then deallocate and delete it.

How do I migrate a virtual machine scale set to a different image?

The migration approach depends on whether an image version, plan/SKU, or entire offer is deprecated.

Version deprecation

If only a specific image version is deprecated:

- **Using "latest" version**: No action required for new instances. Azure resolves to the newest available version at deployment time. Existing instances remain on their original version.
- **Using automatic OS upgrades**: No action required. Azure upgrades existing instances to the latest version automatically, regardless of whether your image reference uses "latest" or a pinned version.
- **Using a pinned version without automatic OS upgrades**: Update the image reference to a supported version. See Modify an Azure virtual machine scale set. After updating the model, upgrade existing instances using one of these methods:
  - **Manual upgrade**: Use `az vmss update-instances` (CLI) or `Update-AzVmssInstance` (PowerShell) to upgrade specific instances.
  - **Rolling upgrade policy**: Configure the upgrade policy to "Rolling" so instances automatically upgrade when the model changes.

> **Tip**

Enable **automatic OS image upgrade** to avoid manual intervention for future version deprecations.

Plan/SKU deprecation

If an entire plan or SKU is deprecated, update to a different SKU from the same offer using a scale-out/scale-in approach:

1. Find an alternative SKU from your deprecation email. For more information, see Find and use images using the Azure CLI or Find and use images using Azure PowerShell.
2. Update the VMSS model to use the new SKU:

   Azure CLI

   ```
   az vmss update --resource-group <resource-group> --name <vmss-name> \
     --set virtualMachineProfile.storageProfile.imageReference.sku=<new-sku> \
     --set virtualMachineProfile.storageProfile.imageReference.version=latest
   ```

3. Scale out to create new instances using the updated image, then scale in to remove old instances. Or delete specific old instances and let the scale set replace them with new ones.

> **Important**
>
> Upgrading or reimaging existing instances replaces their OS disks. Use scale-out/scale-in to avoid data loss on OS disks.

Offer deprecation

If an entire offer is deprecated, create a new virtual machine scale set with a supported image:

1. Find a replacement offer from your deprecation email or search in the Azure Marketplace.
2. Create a new virtual machine scale set using the replacement image. Match your existing configuration (networking, extensions, scaling rules).
3. Test application compatibility thoroughly. A different publisher or offer may have significant OS or configuration differences.
4. Migrate traffic to the new scale set, then delete the old one.

> ⓘ Note
>
> We recommend that you create a new VMSS instead of doing in-place updates when you change publishers or offers, because the differences between images are typically significant.

How do I receive email notifications about image deprecation?

Azure notifies you about image deprecation through two channels:

- **Email notifications**: Sent directly to subscription administrators with the **Owner** or **Account Admin** role on the subscription.
- **Azure Advisor recommendations**: Deprecation recommendations appear in Azure Advisor.

Notify additional recipients

To notify team members who don't have Owner or Account Admin roles on the subscription, create an Advisor alert with an action group. Use action groups to specify email addresses, SMS numbers, or webhooks that receive notifications when new deprecation recommendations are published.

How do I check if a specific image is deprecated or scheduled for deprecation?

Option 1: Use Azure CLI

Azure CLI

```
az vm image show --location eastus --urn MicrosoftWindowsServer:WindowsServer:2016-Datacenter:14393.4169.2101090332 --query [imageDeprecationStatus]
```

Option 2: Use PowerShell

To check a single image version:

PowerShell

```
Get-AzVMImage -Location "eastus" -PublisherName "MicrosoftWindowsServer" -Offer "WindowsServer" -Skus "2016-Datacenter" -Version "14393.4169.2101090332" | Select-
```

```
Object -ExpandProperty "ImageDeprecationStatus"
```

To check all versions of a SKU with deprecation details:

PowerShell

```
Get-AzVMImage -Location "westus" `
  -PublisherName "center-for-internet-security-inc" `
  -Offer "cis-rocky" `
  -Skus "cis-rockylinux-9-l1-gen2-azure-observability" `
  -Expand "properties/imageDeprecationStatus" |
  Select-Object Version, @{
    Name = "DeprecationDetails"
    Expression = { $_.ImageDeprecationStatus | Out-String }
  } | Format-Table -Wrap -AutoSize
```

Option 3: Use REST API

Use the Virtual Machine Images - Get API. If the image is deprecated, you receive a response like:

JSON

```
{
  "error": {
    "code": "ImageVersionDeprecated",
    "message": "VM Image from publisher: MicrosoftWindowsServer with - Offer:
WindowsServer, Sku: 2016-Datacenter, Version: 14393.4169.2101090332 is deprecated."
  }
}
```

Option 4: Use automated script

Run Get-AzVMImageDeprecationStatus.ps1 in Cloud Shell or locally to check all VMs and virtual machine scale sets in a subscription for deprecated images.

How do I list active (non-deprecated) image versions for a specific SKU?

Active images are images that aren't scheduled for deprecation or already deprecated.

Option 1: Use Azure CLI

Azure CLI

```
az vm image list --location eastus --publisher "MicrosoftWindowsServer" --offer
"WindowsServer" --sku "2022-Datacenter" --all --query "[?
imageDeprecationStatus.imageState=='Active']"
```

Option 2: Use PowerShell

PowerShell

```
Get-AzVMImage -Location "westus" `
  -PublisherName "center-for-internet-security-inc" `
  -Offer "cis-rocky" `
  -Skus "cis-rockylinux-9-l1-gen2-azure-observability" `
  -Expand "properties/imageDeprecationStatus" |
  Where-Object { $_.ImageDeprecationStatus.ImageState -eq "Active" } |
  Select-Object Version
```

What happens to SQL Server if a SQL Server Azure Marketplace image is deprecated?

Your SQL Server remains unaffected when a SQL Server Azure Marketplace image is deprecated:

- Your Azure VM continues to run without interruption.
- Your SQL Server instance continues to function normally.
- You can't deploy new VMs using the deprecated SQL Server image.

Existing deployments require no action.

VM Applications overview

Summarize this article for me

VM Applications are a resource type in Azure Compute Gallery that simplifies application management for your virtual machines (VMs) and Virtual Machine Scale Sets. You package your applications once, store them in your gallery, and deploy them to any VM in your organization.

With VM Applications, you decouple application deployment from your base VM images. This separation enables you to update applications independently without rebuilding images, reducing maintenance overhead, accelerating deployment cycles, and effectively handle critical vulnerabilities and disaster recovery.

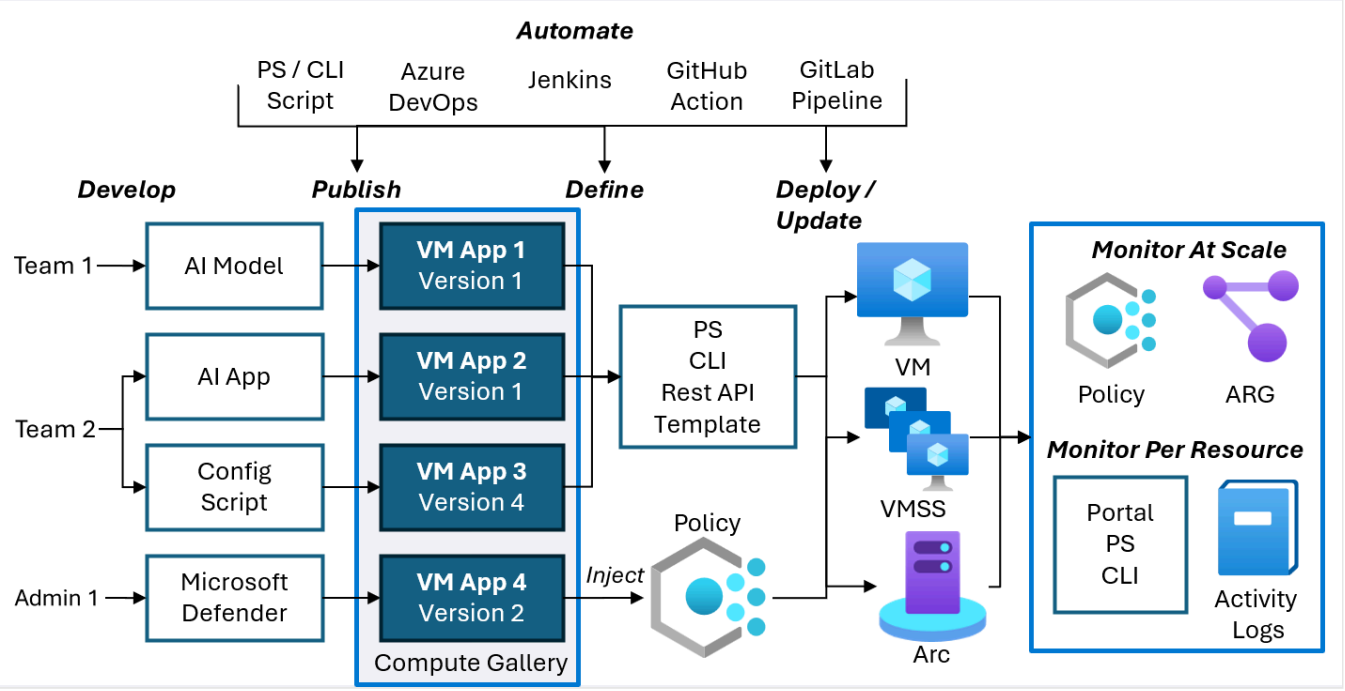
When to use VM Applications

VM Applications are the recommended approach when you need to publish, deploy, and manage software across Azure VMs. Consider VM Applications when you want to:

- Deploy applications across large VM fleets with consistent configuration.
- Update applications frequently without rebuilding VM images.
- Manage application versions and roll back when issues arise.
- Respond to zero-day vulnerabilities by publishing new versions quickly and updating your fleet centrally with Azure Policy.
- Enforce application presence and configuration using Azure Policy.
- Reduce deployment latency for high-scale and time-sensitive workloads like AI inferencing and gaming.
- Gain visibility into deployed applications with post-deployment inventory and monitoring.
- Modularize applications and scripts for reusability and operational efficiency.
- Consolidate application management into a single Azure-managed solution.

How VM Applications work

VM Applications integrate with your existing development workflows and Azure infrastructure to provide end-to-end application lifecycle management.



- **Develop and publish**: Different teams develop applications, scripts, and configurations independently. Each team publishes their packages as Azure VM Application to Azure Compute Gallery, which serves as a personal app repository.
- **Deploy or update**: Deploy published applications to VMs and Virtual Machine Scale Sets using Azure portal, PowerShell, CLI, REST API, or ARM/Bicep templates.
- **Enforce**: Use Azure Policy to automatically inject required applications across your fleet.
- **Automate**: Automate publishing, deployment and updates using CI/CD pipelines like Azure DevOps, GitHub Actions, GitLab pipelines, Jenkins, and scripts.
- **Monitor**: View application inventory & state across your infrastructure. Use Azure Policy and Azure Resource Graph for fleet-wide compliance monitoring, or view per-resource details in the Azure portal, PowerShell, CLI, and activity logs.

Key capabilities

Private application store for your organization

- Store and manage all application packages in Azure Compute Gallery.
- Package applications, scripts, configurations, and files in any format: `.zip`, `.msi`, `.exe`, `.tar.gz`, `.deb`, `.rpm`, `.sh`, or others.
- Maintain multiple versions of each application and deploy `latest` or specific version.
- Deploy test versions without being considered as `latest`.

Access control and sharing

- Control who can publish applications to gallery and access published app to deploy using Azure role-based access control (RBAC).
- Share gallery across subscriptions, Microsoft tenant or publicly.

Security

- Download packages from Azure-managed infrastructure instead of external URLs eliminating need to keep VM internet facing.
- Use Managed Identity assigned to Azure Compute Gallery for securely publishing VM applications.
- Handle critical security vulnerabilities by quickly publishing new version and updating your fleet using Azure Policy or Scripts.
- Use Azure Policy to enforce specific applications, security components, or application configuration across the infrastructure.

Flexible deployment options

- Deploy to individual VMs, flexible, or uniform Virtual Machine Scale Sets.
- Install, update, or remove applications independently without rebuilding VM images.
- Define custom install, update, and remove commands for each application.
- Specify deployment order when installing multiple applications.
- Configure reboot handling using the `scriptBehaviorAfterReboot` property.
- Use `treatFailureAsDeploymentFailure` to mark VM deployment as failed when an application fails to install.

Failure resiliency and high-scale performance

- Automatically replicate application packages across Azure regions and availability zones providing resiliency to region, zone, or content delivery network (CDN) failure.
- Create up to 10 replicas per region to distribute load during high-scale deployments.
- Deploy 25 packages up to 2 GB each, 50 GB total per VM.
- Use block blobs for chunked uploads and background streaming of large packages.

Governance and compliance

- Use Azure Policy to audit and enforce application presence and configuration across your fleet.
- Monitor application inventory and state using Azure portal, Azure Policy, and Azure Resource Graph.
- Monitor and apply pending updates across your infrastructure.

Cost

There's no extra charge for using VM Application Packages, but you're charged for the following resources:

- Storage costs of storing each package and any replicas.
- Network egress charges for replication of the first version from the source region to the replicated regions. Subsequent replicas are handled within the region, so there are no extra charges.

For more information on network egress, see Bandwidth pricing.

Publish application as Azure VM Application

To deploy VM applications, first the application needs to be published to Azure Compute Gallery. To publish Azure VM Application

1. First, create a **VM Application Resource** which is a logical resource containing metadata about the application.
2. Then create a **VM Application Version Resource** within the VM Application resource containing the application package and instructions on how to install, update, delete and replicate the VM application.

Azure resources required to publish Azure VM Application

Expand table

Resource	ARM Resource Type	Description
Azure Compute Gallery	Microsoft.Compute/galleries	A gallery is a repository for managing and sharing application packages. Users can share the gallery resource and all the child resources are shared automatically. The gallery name must be unique per subscription. For example, you may have one gallery to store all your OS images and another gallery to store all your VM applications.
VM Application	Microsoft.Compute/galleries/applications	The definition of your VM application. This logical resource stores the common metadata for all its versions, including the name, description, supported OS type, and end-of-life

Resource	ARM Resource Type	Description
		information. Think of it as a container that holds all versions of a single application. For example, you might have a VM Application called *Apache Tomcat* that contains version 9.0.0, 9.0.1, and 10.0.0.
VM Application Version	Microsoft.Compute/galleries/applications/versions	The deployable resource that contains your actual application package and version-specific configuration. Each version points to the application binary or script in your storage account and defines the install, update, and remove commands. You can replicate versions to multiple Azure regions to improve deployment reliability and reduce latency. Before you deploy an application to a VM, the version must be replicated to the VM's region.
Storage Account	Microsoft.Storage/storageAccounts	Application packages are first uploaded to your storage account. Azure Compute Gallery then downloads the application package from this storage account using SAS URLs and stores it within the VM Application version. Azure Compute Gallery also replicates this package across regions & regional replicas per the VM Application version definition. The application package in the storage account can be deleted after VM application version is created in Azure Compute Gallery.

Properties in VM Application Resource

The VM Application resource defines follows properties:

Expand table

Property	Description	Updatable	Limitations
name	Name of the application	Yes	Max length of 117 characters. Allowed characters are uppercase or lowercase

Property	Description	Updatable	Limitations
			letters, digits, hyphen(-), period (.), underscore (_). Names not allowed to end with period(.).
location	Location of the resource	No	
supportedOSType	Define the supported OS type	No	"Linux" or "Windows"
endOfLifeDate	Optional. A future end of life date for the application. The date is for reference only and isn't enforced.	Yes	
description	Optional. A description of the VM application	Yes	
eula	Optional. Reference to End User License Agreement (EULA)	Yes	
privacyStatementUri	Optional. Reference to privacy statement for the application.	Yes	
releaseNoteUri	Optional. Reference to release notes for the application.	Yes	

Properties in VM Application Version Resource

VM application versions are the deployable resources within the VM Application resource. Versions are defined with the following properties:

Expand table

Property	Description	Updatable	Limitations
location	Source location for the VM Application version.	No	Valid Azure Region
source/mediaLink	Link to the application package file in a storage account	Partially.	Valid and existing storage url. Only SASToken within the

Property	Description	Updatable	Limitations
			SASURL can be changed.
source/defaultConfigurationLink	Optional. A link to the configuration file for the VM application. It can be overridden at deployment time.	Partially	Valid and existing storage url. Only SASToken within the SASURL can be changed.
manageActions/install	Install script as string to properly install the application	No	Valid command for the given OS in string format.
manageActions/remove	Remove script as string to properly remove the application	No	Valid command for the given OS in string format
manageActions/update	Optional. Update script as string to properly update the VM application to a newer version.	No	Valid command for the given OS in string format
targetRegions/name	Name of target regions to which to replicate. Improves resiliency to region failure and create latency.	Yes	Valid Azure region
targetRegions/regionalReplicaCount	Optional. The number of replicas to create in the region. Improves load handling and create latency. Defaults to 1.	Yes	Integer between 1 and 3 inclusive
replicaCount	Optional. Defines the number of replicas across each region. Takes effect if regionalReplicaCount isn't defined. Improves resiliency to region or	Yes	Integer between 1 and 3 inclusive.

Property	Description	Updatable	Limitations
	cluster failure and create latency during high scale.		
endOfLifeDate	Optional. A future end of life date for the application version. This property is for customer reference only and isn't enforced.	Yes	Valid future date
excludeFromLatest	Optional. Exclude version from being used as the latest version of the application when 'latest' keyword is used in applicationProfile.	Yes	Defaults to false.
storageAccountType	Optional. Type of storage account to use in each region for storing application package. Defaults to Standard_LRS.	No	
safetyProfile/allowDeletionOfReplicatedLocations	Optional. Indicates whether or not removing this Gallery Image Version from replicated regions is allowed.	Yes	
settings/packageFileName	Optional. Package file name to use when the package is downloaded to the VM.	No	Character limit is of 4,096 characters.
settings/configFileName	Optional. Configuration file name to be use when the configuration is downloaded to the VM.	No	Character limit is of 4,096 characters.
settings/scriptBehaviorAfterReboot	Optional. The action to be taken for installing, updating, or removing gallery application after the VM is rebooted.	No	

Deploy Azure VM Applications

After the VM Application version is published to Azure Compute Gallery, you can deploy the version across Azure Virtual Machines (VM) and Azure Virtual Machine Scale Sets. This deployment is done by referencing the ARM ID of the VM application in the `applicationProfile` of the Azure Virtual Machine and Virtual Machine Scale Sets.

Properties in applicationProfile of VM and Virtual Machine Scale Sets

The `applicationProfile` in Azure VM and Virtual Machine Scale Sets defines the following properties:

Expand table

Property	Description	Limitations
galleryApplications	Gallery Applications to deploy	
packageReferenceId	Reference to application version to deploy	Valid application version reference
configurationReference	Optional. The full url of a storage blob containing the configuration for this deployment. This overrides any value provided for defaultConfiguration earlier.	Valid storage blob reference
order	Optional. Order in which to deploy applications. When not set, the application is installed last after all ordered applications are installed.	Valid integer
treatFailureAsDeploymentFailure	Optional. Mark application failure as VM deployment failure for failure handling	True or False

The order field may be used to specify dependencies between applications. The rules for order are as follows:

Expand table

Case	Install Meaning	Failure Meaning
No order specified	Unordered applications are installed after ordered applications. There's no guarantee of	Installation failures of other applications, be it ordered or unordered doesn't affect the installation of unordered applications.

Case	Install Meaning	Failure Meaning
	installation order among the unordered applications.	
Duplicate order values	Application is installed in any order compared to other applications with the same order. All applications of the same order are installed after the ones with lower orders and before the ones with higher orders.	If a previous application with a lower order failed to install, no applications with this order install. If any application with this order fails to install, no applications with a higher order install.
Increasing orders	Application will be installed after the ones with lower orders and before the ones with higher orders.	If a previous application with a lower order failed to install, this application doesn't install. If this application fails to install, no application with a higher order installs.

Technical Details for VM Applications

Considerations and Current Limits for VM Applications

- **Up to 10 replicas per region**: When you're creating a VM Application version, the maximum number of replicas per region is 10 for both page blob and block blob.
- **Up to 300 versions per region**: When creating a VM Application version, you can have up to 300 application versions per region across all applications.
- **Manual retry for failed installations**: Currently, the only way to retry a failed installation is to remove the application from the profile, then add it back.
- **Up to 25 applications per VM**: A maximum of 25 applications can be deployed to a single virtual machine.
- **2 GB application size**: The maximum file size of an application version `mediaLink` is 2 GB. The maximum file size for `defaultConfigurationLink` is 1 GB.
- **Requires a VM Agent**: The VM agent must exist on the VM and be able to receive goal states.
- **Single version of the application per VM**: Only one version of a given application can be deployed to a VM.
- **Move operations currently not supported**: Moving VMs with associated VM Applications across resource groups is currently not supported.

> ⓘ Note
>
> For Azure Compute Gallery and VM Applications, Storage SAS can be deleted after replication. However, any subsequent update operation requires a valid SAS.

Download directory within the VM

The download location of the application package and the configuration files are:

- Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>`
- Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.9\Downloads\<application name>\<application version>`

The install/update/remove commands should be written assuming the application package and the configuration file are in the current directory.

File naming for deployed VM application

When the application file gets downloaded to the VM, Azure uses VM Application name as the file name without any file extension. For example, if VM Application name is "TestPythonApp", the python.exe file uploaded in the VM Application gets downloaded as TestPythonApp. Azure is unable to retain the original file name and file extension.

Similarly, if configuration file is passes via `defaultConfigurationLink` in `publishingProfile` or via `configurationReference` in `applicationProfile`, Azure downloads the file with < `vm application name` >-config. For example, if the configuration file name is `config.json' and VM Application name is "TestPythonApp", the downloaded file is named TestPythonApp-config.

VM Applications provide `packageFileName` and `configFileName` properties in the `publishingProfile\settings` to override the default file names and allow customers to set custom file names. For example, if `packageFileName="pythonApp.exe"` and `configFileName="pythonConfig.json`, Azure downloads the files with respective names.

Alternatively, include a command for renaming the files before execution in the `installScript`.

Command interpreter used by VM application

The default command interpreters are:

- Linux: `/bin/bash`
- Windows: `cmd.exe`

It's possible to use a different interpreter like Chocolatey or PowerShell, as long as it's installed on the machine, by calling the executable and passing the command to it. For example, to have your command run in PowerShell on Windows instead of cmd, you can pass `powershell.exe -Command '<powershell commmand>'`

How VM Application updates are handled

When you update an application version on a VM or Virtual Machine Scale Sets, the update command you provided during deployment is used. If the updated version doesn't have an update command, then the current version is removed and the new version is installed.

Update commands should be written with the expectation that it could be updating from any older version of the VM Application.

Treat VM Application failure as deployment failure

By default, if an application fails to install, update, or remove, the VM Application extension still reports its status as *success*. The extension only reports a failure on its own when there's a problem with the extension itself or the underlying infrastructure, not with your application scripts.

To change this behavior, set the `treatFailureAsDeploymentFailure` property to `true` on the application in the VM's `applicationProfile`. When you enable this setting, any application installation, update, or removal failure causes the VMAppExtension provisioning status to report failure. This failure also causes the VM provisioning status to report as failed.

Error messages

These are the error messages that you might encounter when publishing and deploying your VM applications.

Expand table

Message	Description
Current VM Application Version {name} was deprecated at {date}.	You tried to deploy a VM Application version that was deprecated. Try using `latest` instead of

Message	Description
	specifying a specific version.
Current VM Application Version {name} supports OS {OS}, while current OSDisk's OS is {OS}.	You tried to deploy a Linux application to Windows instance or vice versa.
The maximum number of VM applications (max=5, current={count}) has been exceeded. Use fewer applications and retry the request.	We currently only support five VM applications per VM or Virtual Machine Scale Sets.
More than one VM Application was specified with the same packageReferenceId.	The same application was specified more than once.
Subscription not authorized to access this image.	The subscription doesn't have access to this application version.
Storage account in the arguments doesn't exist.	There are no applications for this subscription.
The platform image {image} isn't available. Verify that all fields in the storage profile are correct. For more information about storage profile, see https://aka.ms/storageprofile .	The application doesn't exist.
The gallery image {image} isn't available in {region} region. Contact image owner to replicate to this region, or change your requested region.	The gallery application version exists, but it wasn't replicated to this region.
The SAS isn't valid for source uri {uri}.	A `Forbidden` error was received from storage when attempting to retrieve information about the url (either mediaLink or defaultConfigurationLink).
The blob referenced by source uri {uri} doesn't exist.	The blob provided for the mediaLink or defaultConfigurationLink properties doesn't exist.
The gallery application version url {url} can't be accessed due to the following error: remote name not found. Ensure that the blob exists and that it's either publicly accessible or is a SAS url with read privileges.	The most likely case is that a SAS uri with read privileges wasn't provided.

Message	Description
The gallery application version url {url} can't be accessed due to the following error: {error description}. Ensure that the blob exists and that it's either publicly accessible or is a SAS url with read privileges.	There was an issue with the storage blob provided. The error description provides more information.
Operation {operationName} isn't allowed on {application} since it's marked for deletion. You can only retry the delete operation (or wait for an ongoing one to complete).	Attempt to update an application that's currently being deleted.
The value {value} of parameter 'galleryApplicationVersion.properties.publishingProfile.replicaCount' is out of range. The value must be between one and three, inclusive.	Only between one and three replicas are allowed for VM Application versions.
Changing property 'galleryApplicationVersion.properties.publishingProfile.manageActions.install' isn't allowed. (Or update, delete)	It isn't possible to change any of the manage actions on an existing VmApplication. A new VmApplication version must be created.
Changing property ' galleryApplicationVersion.properties.publishingProfile.settings.packageFileName ' isn't allowed. (Or configFileName)	It isn't possible to change any of the settings, such as the package file name or config file name. A new VmApplication version must be created.
The blob referenced by source uri {uri} is too large: size = {size}. The maximum blob size allowed is '1 GB'.	The maximum size for a blob referred to by mediaLink or defaultConfigurationLink is currently 1 GB.
The blob referenced by source uri {uri} is empty.	An empty blob was referenced.
{type} blob type isn't supported for {operation} operation. Only page blobs and block blobs are supported.	VmApplications only supports page blobs and block blobs.
The SAS isn't valid for source uri {uri}.	The SAS uri supplied for mediaLink or defaultConfigurationLink isn't a valid SAS uri.
Can't specify {region} in target regions because the subscription is missing required feature {featureName}. Either register your subscription with the	To use VmApplications in certain restricted regions,

Message	Description
required feature or remove the region from the target region list.	one must have the feature flag registered for that subscription.
Gallery image version publishing profile regions {regions} must contain the location of image version {location}.	The list of regions for replication must contain the location where the application version is.
Duplicate regions aren't allowed in target publishing regions.	The publishing regions may not have duplicates.
Gallery application version resources currently don't support encryption.	The encryption property for target regions isn't supported for VM Applications
Entity name doesn't match the name in the request URL.	The gallery application version specified in the request url doesn't match the one specified in the request body.
The gallery application version name is invalid. The application version name should follow Major(int32). Minor(int32). Patch(int32) format, where `int` is between 0 and 2,147,483,647 (inclusive). For example, 1.0.0, 2018.12.1 etc.	The gallery application version must follow the format specified.

Next steps

Learn how to

- Create and deploy VM application packages.
- Manage and delete Azure VM Applications.
- Create application package for VM Applications
- Publish VM application using Managed Identity
- Govern and enforce applications using Azure Policy

Check out

- Azure dev blogs about Azure VM Applications .
- Reimagining VM Application Management for an AI-powered, secure future

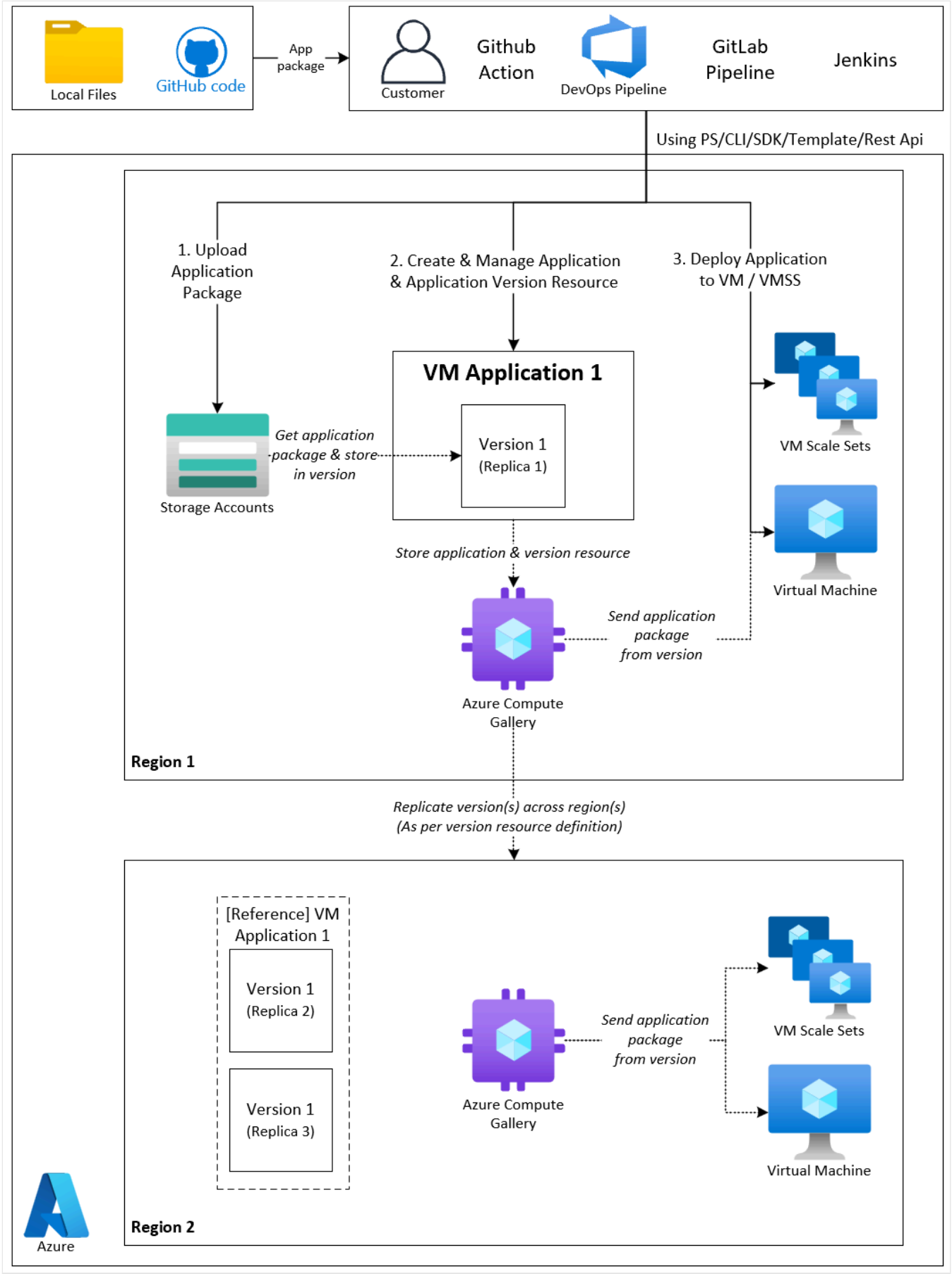
Last updated on 03/06/2026

Create and deploy VM Application

Summarize this article for me

VM Application is a resource type in Azure Compute Gallery that simplifies publishing, deployment, management, sharing, and global distribution of applications and scripts for your virtual machines. VM Applications support a wide range of scenarios, including high-scale deployments, low-latency requirements, failure resiliency, secure trusted rollouts, fleet-wide consistency, microservice architectures, and post-deployment management. Learn more about VM Applications.

To create and deploy applications on Azure VM, first package and upload your application to Azure Storage Account as a storage blob. Then create `Azure VM application` resource and `VM application version` resource referencing these storage blobs. Azure stores and replicates these blobs in regional managed storage accounts. Finally, deploy the application on any VM or Virtual Machine Scale Sets by passing application reference in `applicationProfile`.



Prerequisites

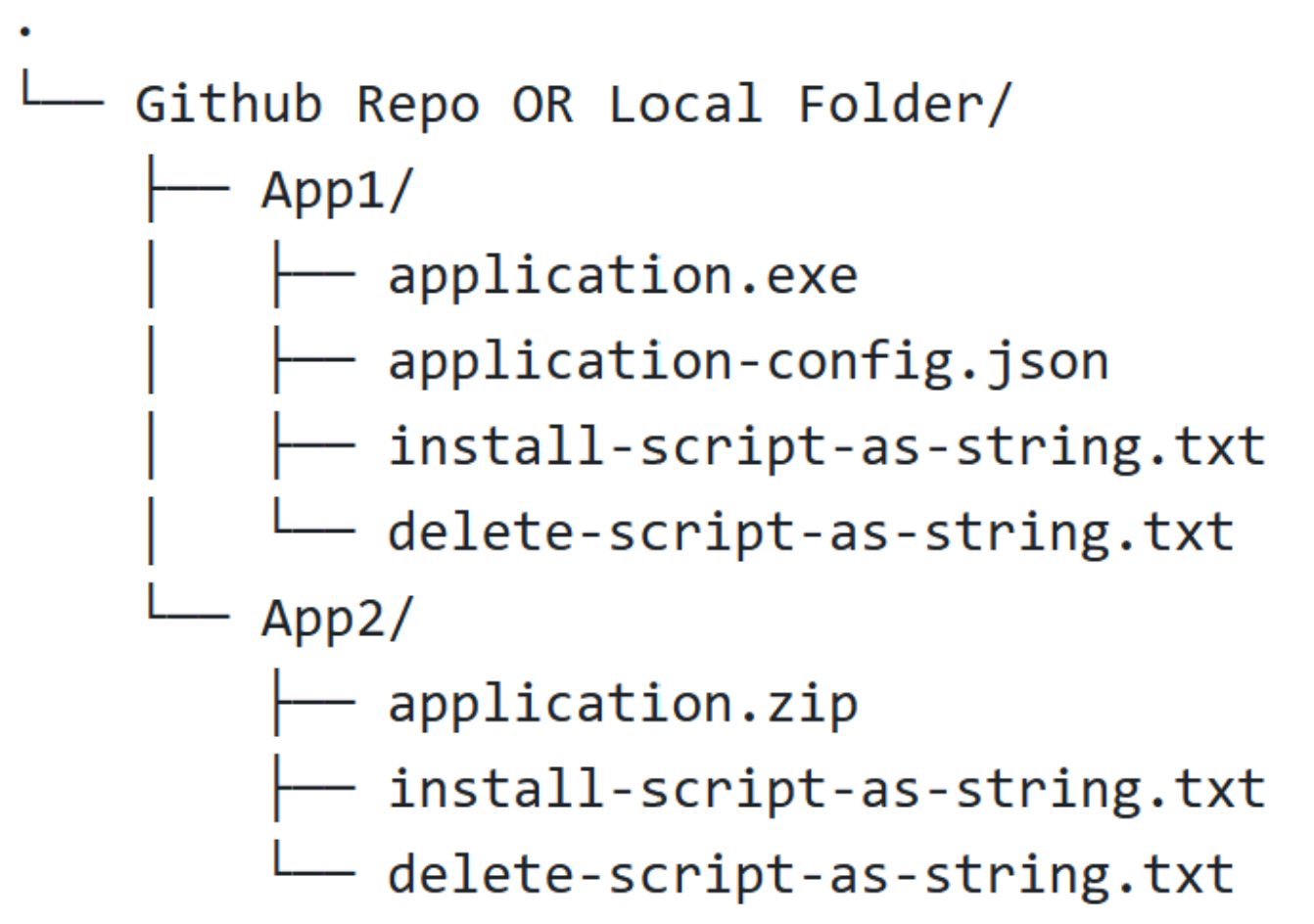
1. Create Azure storage account and storage container. This container is used to upload your application files.

2. Create Azure Compute Gallery for storing and sharing application resources.
3. Create user-assigned managed identity and assign to Azure Compute Gallery

> ⓘ Note
>
> Create your storage account with anonymous access disabled for improved security. Use a managed identity assigned to Azure Compute Gallery to publish your VM Applications securely without requiring SAS tokens.

Package the application

Before publishing the VM Application resource in Azure Compute Gallery, package your application files for upload to Azure Storage Account. The following diagram shows possible folder structures for organizing your application files:



1. Package the application files

VM Application requires a single file as the application package.

- If your application installation requires a single executable file (.exe, .msi, .sh, .ps, etc.) then you can use it as the application package.
- If your application requires multiple files (executables, dependencies, scripts, metadata, etc.), archive them into a single file (`.zip`, `.tar`, `.tar.gz`) and use this file as the application package.
- For microservice application, you can package and publish each microservice as a separate Azure VM Application. This modularization facilitates application reusability, cross-team development, and sequential installation of microservices using `order` property in the applicationProfile.

Learn more about creating application packages for Azure VM Applications.

2. (Optional) Package the application configuration file

- You can provide the configuration file separately using `defaultConfigurationLink` in the `publishingProfile` of the VM Application. This reduces the overhead of archiving and unarchiving application packages.
- Configuration files can also be passed during app deployment using `configurationReference` property in the `applicationProfile` of the VM enabling customized installation per VM.
- If both properties are set, `configurationReference` overrides the configuration passed using `defaultConfigurationLink`.

3. Create the install script

After the application and configuration blob is downloaded on the VM, Azure executes the provided install script to install the application. **The install script is provided as a string** and has a maximum character limit of 4,096 chars. The install commands should be written assuming the application package and the configuration file are in the current directory.

There may be few operations required to be performed in the install script

- **(Optional) Use the right command interpreter** The default command interpreter used by Azure are `/bin/bash` on Linux OS and `cmd.exe` on Windows OS. It's possible to use a different interpreter like Chocolatey or PowerShell, if its installed on the machine. Call the executable and pass the commands to it. For example, `powershell.exe -command '<powershell command>'`. If you're using PowerShell, you need to be using version 3.11.0 of the Az.Storage module.

- **(Optional) Rename application blob and configuration blob** Azure can't retain the original blob name and the file extensions. Azure downloads the application file with the

name of the VM application resource published on Azure Compute Gallery. So if name of VM App is "VMApp1", the file downloaded on VM has the name "VMApp1" without an extension. The configuration file has a default name as "VMApp1-config" without a file extension.

You can rename the file with the file extension using the install script or you can also pass the names in `packageFileName` and `configFileName` properties of the publishingProfile of VM Application version resource. Azure uses these names instead of default names while downloading the files.

- **Unarchive application blob** For archived application packages, it needs to be unarchived before installing the application. Use .zip or .tar since most OS has built-in support for unarchiving these formats. For other formats, make sure the Guest OS provides support.

- **Convert the script to string** The install script is passed as a string for the `install` property in the `publishingProfile` of Azure VM Application version resource.

- **Install quietly and synchronously** Install applications quietly or silently without any user interface prompts. Since the installation runs on a VM where customer input isn't available, you must use silent or unattended installation flags. Additionally, if the application installer spawns background or child processes or runs asynchronously, the parent script might return before the installation completes. To ensure the full installation finishes before Azure marks the operation as successful, use wait flags or blocking commands. This approach helps when installation involves multiple steps that run asynchronously or you need to verify installation completes before the subsequent application in the `order` sequence begins.

  Common examples include:
  - **Windows .msi**: `start /wait msiexec /i app.msi /quiet /qn`
  - **Windows .exe**: `start /wait app.exe /silent` or `start /wait app.exe /quiet` or `start /wait app.exe /S`
  - **PowerShell**: `Start-Process -FilePath '.\app.exe' -ArgumentList '/quiet' -Wait`
  - **Linux .sh**: `./install.sh --silent` (shell scripts run synchronously by default)
  - **Linux .deb**: `sudo DEBIAN_FRONTEND=noninteractive dpkg -i app.deb` (runs synchronously by default)
  - **Linux .rpm**: `sudo rpm -ivh app.rpm` (runs synchronously by default)

- **(Optional) Set right execution policy and permissions** After unarchiving, file permissions could be reset. It's a good practice to set the right permissions before executing the files.

- **(Optional) Set verbose execution and logging** VM Applications automatically captures any output from the install script and stores it in a stdout file on the VM. Use this

capability to log installation progress, debug failures, and verify successful deployments.
    - **Enable verbose mode** to capture application installer logs. Bash: `set -x`, PowerShell: `$VerbosePreference='Continue'`, Cmd: `@echo on`.
    - **Add custom messages** to track script progress. Bash: `echo "Install started"`, PowerShell: `Write-Output "Install started"`, Cmd: `echo Install started`

    You can retrieve these logs using Run Command or by connecting to the VM. The `stdout` and `stderr` files are located at the following path
    - Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>/`
    - Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.9\Downloads\<application name>\<application version>/`

- **(Optional) Move application and configuration blob to appropriate location** Azure downloads the application blob and configuration blob to following locations. The install script must move the files to appropriate locations when necessary.

    Linux: `/var/lib/waagent/Microsoft.CPlat.Core.VMApplicationManagerLinux/<application name>/<application version>`

    Windows: `C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\1.0.16\Downloads\<application name>\<application version>`

Here are few sample install scripts based on the file extension of the application blob. If `packageFileName` & `configFileName` properties are provided in the `publishingProfile` with file name and extension, remove the code for renaming files in following scripts.

.TAR

Replace `VMApp1` with your VM App name in application package (VMApp1) and configuration file (VMApp1-config). Replace `app.tar` with desired app name. Replace `config.yaml` with your desired config name.

Install using bash without config file:

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && bash ./install.sh
```

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && sudo
DEBIAN_FRONTEND=noninteractive dpkg -i ./app.deb
```

Install using bash with config file:

Bash

```
mv VMApp1 app.tar && mv VMApp1-config config.yaml && tar -xf app.tar && chmod -
R +xr . && bash ./install.sh --config config.yaml
```

If config file is packaged within the tar file

Bash

```
mv VMApp1 app.tar && tar -xf app.tar && chmod -R +xr . && sudo debconf-set-
selections < config.cfg && sudo DEBIAN_FRONTEND=noninteractive dpkg -i
./app.deb
```

4. Create the remove script

The `remove` script enables you to define the operations for removing the application. The `remove` script is provided as a string and has a maximum character limit of 4,096 characters. Write the commands assuming the application package and the configuration file are in the current directory. During uninstall operation, Azure runs the uninstall script and then deletes all files from the repository.

There are few operations that the `remove` script must perform.

- **Uninstall application:** Properly uninstall the application from the VM. For example:

CMD

CMD on Windows:

cli

```
start /wait uninstall.exe /quiet
```

cli

```
start /wait python.exe /uninstall /quiet
```

cli

```
start /wait msiexec /x app.msi /quiet /qn
```

- **Remove residual files:** If the application installation creates files in other parts of the file system, remove those files.

Upload the application files to Azure storage account

> ⓘ **Important**
>
> Before uploading files or generating SAS URLs, assign roles to the user, service principal, or managed identity that performs these operations on the target Storage account or container:
>
> - Storage Blob Data Contributor or Storage Blob Data Owner: required to upload, modify, or delete blobs.
> - For generating SAS with Azure AD (for example, using `--auth-mode` login in CLI/PowerShell), also assign Storage Blob Delegator at the storage account scope.

1. Upload your application and configuration files to a container in your Azure storage account.

Upload your packaged application and optional configuration file as blobs. Block blobs work for most scenarios.

Optionally, if you need to use page blobs, byte-align your files before uploading using one of the following scripts:

CLI

Azure CLI

```
inputFile="<the file you want to pad>"

# Get the file size
fileSize=$(stat -c %s "$inputFile")

# Calculate the remainder when divided by 512
```

```
remainder=$((fileSize % 512))

if [ "$remainder" -ne 0 ]; then
  # Calculate how many bytes to pad
  difference=$((512 - remainder))

  # Create padding (empty bytes)
  dd if=/dev/zero bs=1 count=$difference >> "$inputFile"
fi
```

2. Generate SAS or Blob URL for the application package and the configuration file

Once the application and configuration files are uploaded to the storage account, you need to generate a Blob URL or SAS URL with read privilege for these blobs. These URLs are then provided as reference while creating the VM Application version resource.

> ⓘ **Note**
>
> Use of blob URL with managed identity is strongly recommended for improved security and privacy.

Use blob URL:

- If managed identity is assigned to Azure Compute Gallery.
- If storage account is enabled for anonymous access. Use this method only for testing since its insecure.

Use SAS URL:

- If storage account is disabled for anonymous access and managed identity isn't assigned to Azure Compute Gallery.

Blob URL = https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${BLOB_NAME}
SAS URL = https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${BLOB_NAME}?${SAS_TOKEN}

CLI using Blob URL

Use the following script if managed identity is assigned to Azure Compute Gallery. This approach uses blob URLs without SAS tokens.

Azure CLI

```
#!/bin/bash
set -euo pipefail

# === CONFIGURATION ===
STORAGE_ACCOUNT="your-storage-account-name"
CONTAINER_NAME="your-container-name"
APP_FILE="./path/to/your-app-file"           # Path to your application payload
file
CONFIG_FILE="./path/to/your-config-file"     # Path to your configuration file
(optional)

# === LOGIN (if not already logged in) ===
az login --only-show-errors

# === CREATE CONTAINER IF NOT EXISTS ===
az storage container create \
  --name "$CONTAINER_NAME" \
  --account-name "$STORAGE_ACCOUNT" \
  --auth-mode login \
  --only-show-errors

# === UPLOAD APPLICATION FILE ===
APP_BLOB_NAME=$(basename "$APP_FILE")
az storage blob upload \
  --account-name "$STORAGE_ACCOUNT" \
  --container-name "$CONTAINER_NAME" \
  --name "$APP_BLOB_NAME" \
  --file "$APP_FILE" \
  --auth-mode login \
  --overwrite \
  --only-show-errors

# === UPLOAD CONFIG FILE (optional) ===
if [ -n "$CONFIG_FILE" ] && [ -f "$CONFIG_FILE" ]; then
  CONFIG_BLOB_NAME=$(basename "$CONFIG_FILE")
  az storage blob upload \
    --account-name "$STORAGE_ACCOUNT" \
    --container-name "$CONTAINER_NAME" \
    --name "$CONFIG_BLOB_NAME" \
    --file "$CONFIG_FILE" \
    --auth-mode login \
    --overwrite \
    --only-show-errors
fi

# === GENERATE BLOB URLs ===
echo "Generating Blob URLs..."

APP_BLOB_URL="https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME
}/${APP_BLOB_NAME}"
echo "Application file: $APP_BLOB_URL"
```

```
if [ -n "$CONFIG_FILE" ] && [ -f "$CONFIG_FILE" ]; then

CONFIG_BLOB_URL="https://${STORAGE_ACCOUNT}.blob.core.windows.net/${CONTAINER_NAME}/${CONFIG_BLOB_NAME}"
  echo "Configuration file: $CONFIG_BLOB_URL"
fi
```

Create the VM Application

To create the VM Application, first create the **VM Application** resource, which describes the application. Then create **VM Application Version** resource within it, which contains the VM application payload and scripts to install, update, and delete the application. Payload is supplied using Blob URL or SAS URL to the blob container in Azure Storage Account.

Refer schema for VM Application and VM Application version resource to learn more about each property.

Template

Use the following ARM template to create an Azure Compute Gallery, VM Application, and VM Application version. This template demonstrates the key properties and configuration options for publishing your application package.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2020-06-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "galleryName": {
      "type": "string"
    },
    "applicationName": {
      "type": "string"
    },
    "versionName": {
      "type": "string",
      "metadata": {
        "description": "Must follow the format: major.minor.patch (Example: 1.0.0)"
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "West US"
    },
```

```
    "supportedOSType": {
      "type": "string",
      "allowedValues": ["Windows", "Linux"]
    },
    "endOfLifeDate": {
      "type": "string",
      "metadata": {
        "description": "Optional. This property is for information only and
doesn't block app deployment."
      }
    },
    "description": {
      "type": "string",
      "defaultValue": "Description of the application"
    },
    "eula": {
      "type": "string",
      "defaultValue": ""
    },
    "privacyStatementUri": {
      "type": "string",
      "defaultValue": ""
    },
    "releaseNoteUri": {
      "type": "string",
      "defaultValue": ""
    },
    "mediaLink": {
      "type": "string"
    },
    "configLink": {
      "type": "string"
    },
    "appConfigFileName": {
      "type": "string"
    },
    "appPackageFileName": {
      "type": "string"
    },
    "replicaRegion1": {
      "type": "string",
      "defaultValue": "East US"
    },
    "replicaRegion2": {
      "type": "string",
      "defaultValue": "South Central US"
    },
    "installScript": {
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to install the application.
Example: echo 'Installing application...'"
      }
    },
    "updateScript": {
```

```
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to update the application.
Example: echo 'Updating application...'"
      }
    },
    "removeScript": {
      "type": "string",
      "metadata": {
        "description": "Optional. Script to run to delete the application.
Example: echo 'Deleting application...'"
      }
    },
    "storageAccountType": {
      "type": "string",
      "allowedValues": ["PremiumV2_LRS", "Premium_LRS", "Standard_LRS",
"Standard_ZRS"],
      "defaultValue": "Standard_LRS"
    }
  },
  "resources": [
    {
      "type": "Microsoft.Compute/galleries",
      "apiVersion": "2024-03-03",
      "name": "[parameters('galleryName')]",
      "location": "[parameters('location')]",
      "properties": {
        "identifier": {}
      }
    },
    {
      "type": "Microsoft.Compute/galleries/applications",
      "apiVersion": "2024-03-03",
      "name": "[format('{0}/{1}', parameters('galleryName'),
parameters('applicationName'))]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Compute/galleries',
parameters('galleryName'))]"
      ],
      "properties": {
        "supportedOSType": "[parameters('supportedOSType')]",
        "endOfLifeDate": "[parameters('endOfLifeDate')]",
        "description": "[parameters('description')]",
        "eula": "[if(equals(parameters('eula'), ''), json('null'),
parameters('eula'))]",
        "privacyStatementUri": "[if(equals(parameters('privacyStatementUri'),
''), json('null'), parameters('privacyStatementUri'))]",
        "releaseNoteUri": "[if(equals(parameters('releaseNoteUri'), ''),
json('null'), parameters('releaseNoteUri'))]"
      }
    },
    {
      "type": "Microsoft.Compute/galleries/applications/versions",
      "apiVersion": "2024-03-03",
```

```
      "name": "[format('{0}/{1}/{2}', parameters('galleryName'),
parameters('applicationName'), parameters('versionName'))]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Compute/galleries/applications',
parameters('galleryName'), parameters('applicationName'))]"
      ],
      "properties": {
        "publishingProfile": {
          "source": {
            "mediaLink": "[parameters('mediaLink')]",
            "defaultConfigurationLink": "[parameters('configLink')]"
          },
          "manageActions": {
            "install": "[parameters('installScript')]",
            "remove": "[parameters('removeScript')]",
            "update": "[parameters('updateScript')]"
          },
          "settings": {
            "scriptBehaviorAfterReboot": "Rerun",
            "configFileName": "[parameters('appConfigFileName')]",
            "packageFileName": "[parameters('appPackageFileName')]"
          },
          "targetRegions": [
            {
              "name": "[parameters('location')]",
              "regionalReplicaCount": 3,
              "storageAccountType": "[parameters('storageAccountType')]"
            },
            {
              "name": "[parameters('replicaRegion1')]",
              "regionalReplicaCount": 1,
              "storageAccountType": "[parameters('storageAccountType')]"
            },
            {
              "name": "[parameters('replicaRegion2')]"
            },
          ],
          "excludeFromLatest": false,
          "replicaCount": 2,
          "storageAccountType": "[parameters('storageAccountType')]"
        },
        "safetyProfile": {
          "allowDeletionOfReplicatedLocations": true
        },
        "endOfLifeDate": "[parameters('endOfLifeDate')]"
      }
    }
  ]
}
```

Deploy the VM Apps

One or more VM Applications can now be referenced in the `applicationProfile` of Azure VM or Azure Virtual Machine Scale Sets. Azure then pulls the payload of the VM Application and installs it on each VM using the provided install script. The `order` property defines the sequential order in which the VM Applications are installed on the VM.

Refer schema of applicationProfile of the VM / Virtual Machine Scale Sets to learn more about each property.

Template

Use the following ARM template to deploy VM Application on Azure VM or Azure Virtual Machine Scale Sets. This template demonstrates the key properties and configuration options for deploying your VM application.

Deploy VM Application on Virtual Machine Scale Sets

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmssName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "subscriptionId": {
      "type": "string"
    },
    "resourceGroupName": {
      "type": "string"
    },
    "galleryName": {
      "type": "string"
    },
    "applicationName1": {
      "type": "string"
    },
    "applicationVersion1": {
      "type": "string",
      "defaultValue": "latest"
    },
    "configurationReference1": {
      "type": "string",
```

```
      "metadata": {
        "description": "Optional path to configuration file from Storage
Account. Overrides default configuration file."
      }
    },
    "applicationName2": {
      "type": "string"
    },
    "applicationVersion2": {
      "type": "string",
      "defaultValue": "latest"
    }
  },
  "variables": {
    "packageReferenceId1": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName1'), parameters('applicationVersion1'))]",
    "packageReferenceId2": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName2'), parameters('applicationVersion2'))]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachineScaleSets",
      "apiVersion": "2024-03-03",
      "name": "[parameters('vmssName')]",
      "location": "[parameters('location')]",
      "properties": {
        "virtualMachineProfile": {
          "applicationProfile": {
            "galleryApplications": [
              {
                "order": 1,
                "packageReferenceId": "[variables('packageReferenceId1')]",
                "configurationReference": "
[parameters('configurationReference1')]",
                "treatFailureAsDeploymentFailure": true
              },
              {
                "order": 2,
                "packageReferenceId": "[variables('packageReferenceId2')]",
                "treatFailureAsDeploymentFailure": false
              }
            ]
          }
        }
      }
    }
  ]
}
```

Deploy VM Application on Azure VM

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "subscriptionId": {
      "type": "string"
    },
    "resourceGroupName": {
      "type": "string"
    },
    "galleryName": {
      "type": "string"
    },
    "applicationName1": {
      "type": "string"
    },
    "applicationVersion1": {
      "type": "string",
      "defaultValue": "latest"
    },
    "configurationReference1": {
      "type": "string",
      "metadata": {
        "description": "Optional path to configuration blob for application 1"
      }
    },
    "applicationName2": {
      "type": "string"
    },
    "applicationVersion2": {
      "type": "string",
      "defaultValue": "latest"
    }
  },
  "variables": {
    "packageReferenceId1": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName1'), parameters('applicationVersion1'))]",
    "packageReferenceId2": "
[format('/subscriptions/{0}/resourceGroups/{1}/providers/Microsoft.Compute/gall
eries/{2}/applications/{3}/versions/{4}', parameters('subscriptionId'),
```

```
parameters('resourceGroupName'), parameters('galleryName'),
parameters('applicationName2'), parameters('applicationVersion2'))]"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2024-07-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "applicationProfile": {
          "galleryApplications": [
            {
              "order": 1,
              "packageReferenceId": "[variables('packageReferenceId1')]",
              "configurationReference": "
[parameters('configurationReference1')]",
              "treatFailureAsDeploymentFailure": true
            },
            {
              "order": 2,
              "packageReferenceId": "[variables('packageReferenceId2')]",
              "treatFailureAsDeploymentFailure": false
            }
          ]
        }
      }
    }
  ]
}
```

Next steps

- Learn to manage, update, or delete Azure VM Applications.
- Learn more about Azure VM Applications.

Last updated on 03/06/2026

Manage Azure VM Applications

Summarize this article for me

This article talks about how to view, update, and delete published VM Application in Azure Compute Gallery. It then talks about how to view, monitor, update, and delete deployed VM application resource on Azure Virtual Machine (VM) or Virtual Machine Scale Sets.

Managed VM Application published in Azure Compute Gallery

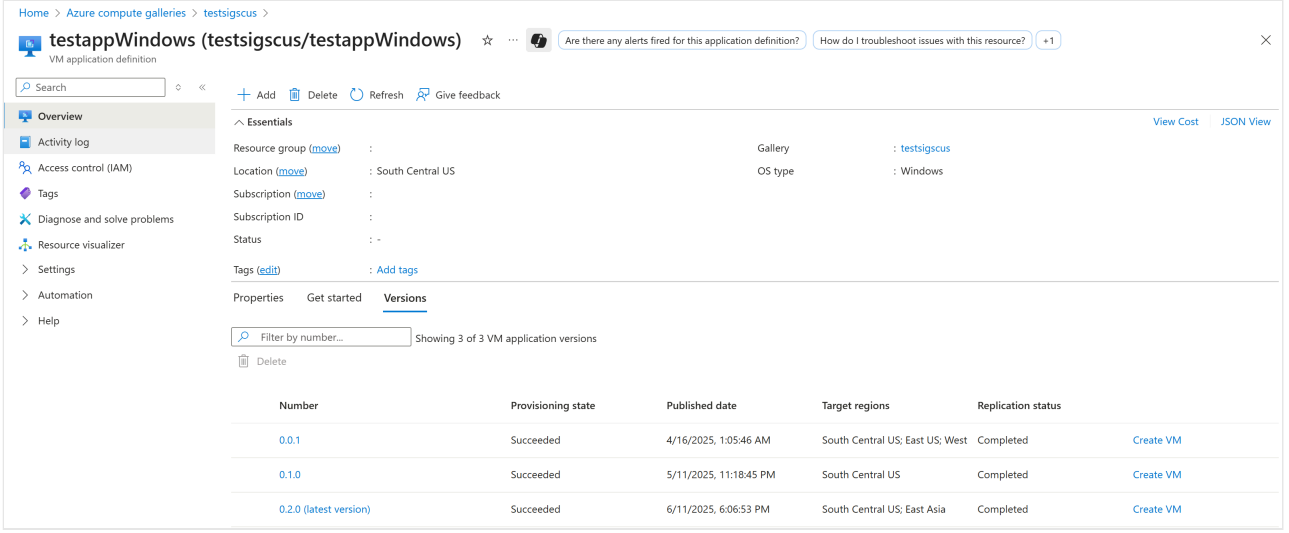
This section explains about how to view, update, and delete **published VM Applications** in Azure Compute Gallery.

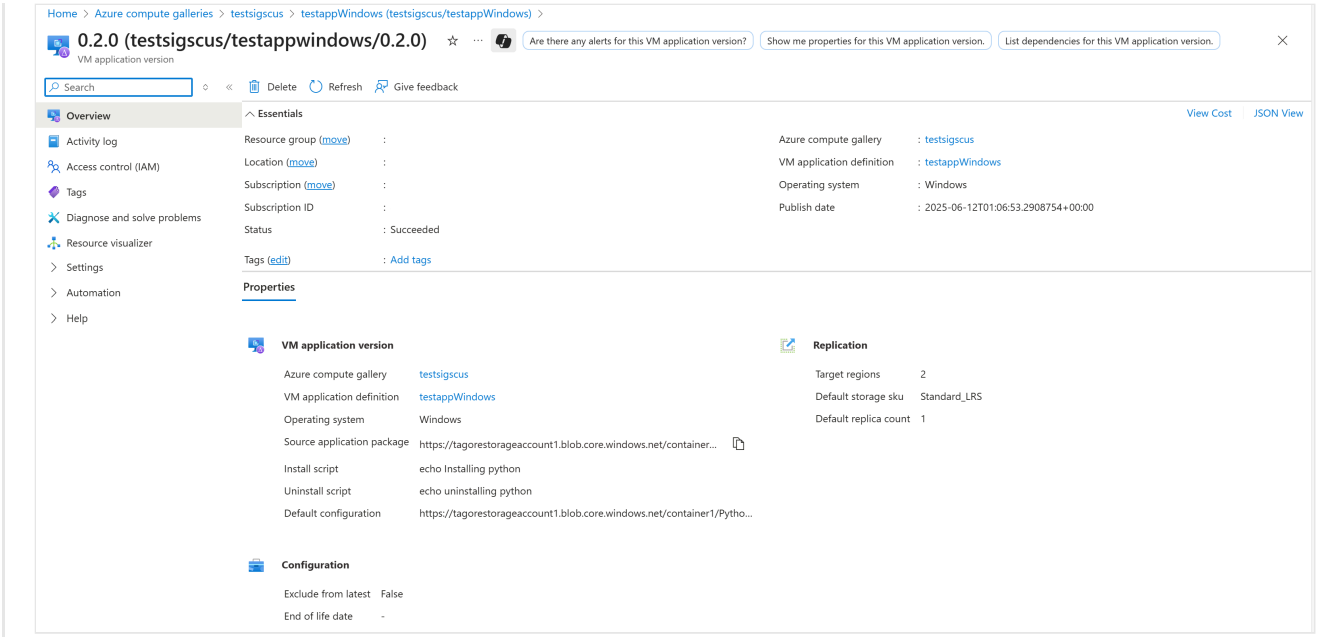
View published VM Applications

Portal

To view the properties of a published VM Application in the Azure portal:

1. Sign in to the Azure portal.
2. Search for **Azure Compute Gallery**.
3. Select the gallery that contains your VM Application.
4. Click the **VM Application Name** you want to view.
5. The **Overview/Properties** blade displays information about the VM Application.
6. The **Overview/Versions** blade displays all published versions and its basic properties like Target Regions, Provisioning state, and Replication state.
7. Select a **specific version** to view all its details.





View published VM applications using Azure Resource Graph

Azure resource graph query can be used to view all published VM applications and their properties across all compute galleries. It provides a programmatic way to view application inventory, and their properties at high scale. Use this method for integrating with dashboards and custom reports.

View list of all compute galleries

Kusto

```
resources
| where type == "microsoft.compute/galleries"
| where subscriptionId == "85236c53-92ad-4e66-97a4-8253a5246b99"
| extend provisioningState = properties["provisioningState"]
| extend permissions = properties["sharingProfile"]["permissions"]
| extend communityGalleryInfo = properties["sharingProfile"]
["communityGalleryInfo"]
| project subscriptionId, resourceGroup, location, name, provisioningState,
permissions, communityGalleryInfo
```

View list of all published VM Applications across all galleries

Kusto

```
resources
| where subscriptionId == "85236c53-92ad-4e66-97a4-8253a5246b99"
| where type == "microsoft.compute/galleries/applications"
| extend OSType = properties["supportedOSType"]
```

```
| extend description = properties["description"]
| extend endOfLifeDate = properties["endOfLifeDate"]
| parse id with  "/subscriptions/" SubId "/resourceGroups/" rgName
"/providers/Microsoft.Compute/galleries/" gallaryName "/applications/" appName
| project subscriptionId, resourceGroup, location, gallaryName, name, OSType,
description
```

View list of all published VM Application Versions across all applications and galleries

Kusto

```
resources
| where type == "microsoft.compute/galleries/applications/versions"
| project subscriptionId, resourceGroup, id, location, properties
| parse id with  "/subscriptions/" SubId "/resourceGroups/" rgName
"/providers/Microsoft.Compute/galleries/" gallaryName "/applications/" appName
"/versions/" versionNumber
| extend provisioningState = properties["provisioningState"]
| extend publishingProfile = properties["publishingProfile"]
| extend storageAccountType = publishingProfile["storageAccountType"]
| extend scriptBehaviorAfterReboot = publishingProfile["settings"]
["scriptBehaviorAfterReboot"]
| extend packageFileName = publishingProfile["settings"]["packageFileName"]
| extend configFileName = publishingProfile["settings"]["configFileName"]
| extend mediaLink = publishingProfile["source"]["mediaLink"]
| extend defaultConfigurationLink = publishingProfile["source"]
["defaultConfigurationLink"]
| extend excludeFromLatest = publishingProfile["excludeFromLatest"]
| extend targetRegions = publishingProfile["targetRegions"]
| extend replicaCount = publishingProfile["replicaCount"]
| extend publishedDate = publishingProfile["publishedDate"]
| extend installScript = publishingProfile["manageActions"]["install"]
| extend removeScript = publishingProfile["manageActions"]["remove"]
| extend safetyProfile = properties["safetyProfile"]
| extend allowDeletionOfReplicatedLocations =
safetyProfile["allowDeletionOfReplicatedLocations"]
| project-away safetyProfile, publishingProfile, SubId, rgName, id, properties
```

Update published VM Application

> **Note**
>
> Not all properties of a published VM Application or VM Application version can be updated. For a complete list of updatable properties, refer to the VM Application resource and VM Application version resource schema.

REST

Update VM Application resource:

rest

```
PATCH
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/galleries/{galleryName}/applications/{galleryApplicationName}?api-version=2024-03-03

Body
{
    "properties": {
        "description": "Updated description",
        "endOfLifeDate": "2026-12-31T00:00:00Z",
        "eula": "Updated EULA text",
        "privacyStatementUri": "https://contoso.com/privacy",
        "releaseNoteUri": "https://contoso.com/release-notes"
    }
}
```

Update VM Application version resource:

rest

```
PATCH
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/galleries/{galleryName}/applications/{galleryApplicationName}/versions/{galleryApplicationVersionName}?api-version=2024-03-03

Body
{
    "properties": {
        "publishingProfile": {
            "targetRegions": [
                {
                    "name": "eastus",
                    "regionalReplicaCount": 2,
                    "storageAccountType": "Standard_LRS"
                },
                {
                    "name": "westus",
                    "regionalReplicaCount": 1,
                    "storageAccountType": "Standard_LRS"
                }
            ],
            "excludeFromLatest": false
        }
```

```
    }
}
```

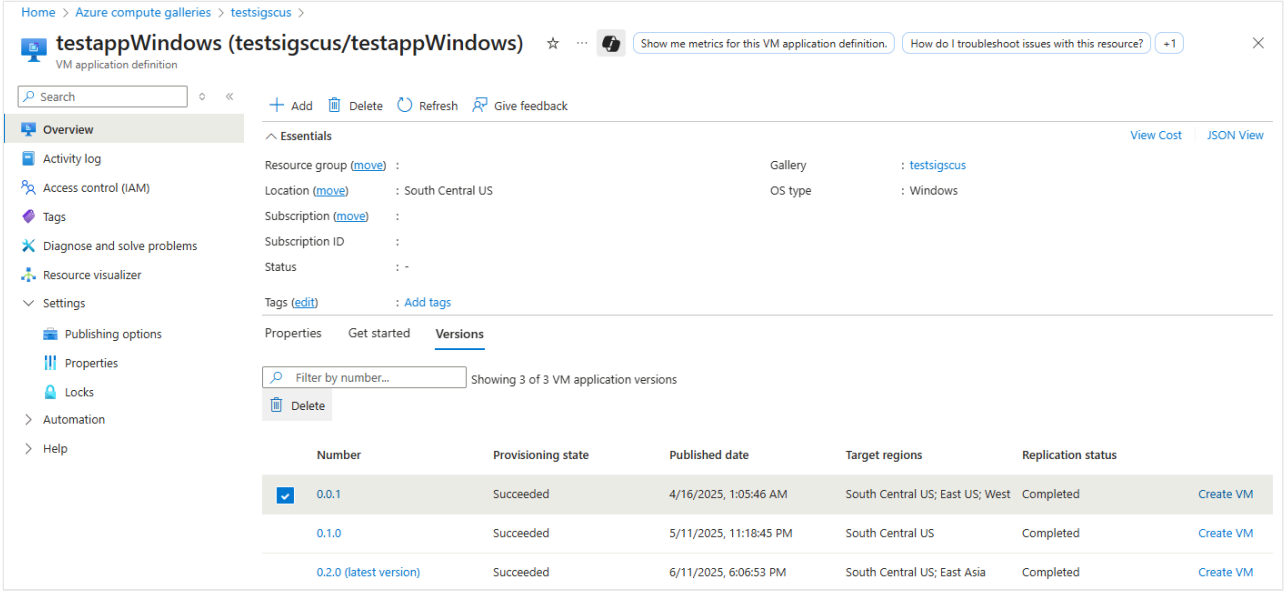
Delete published VM Application from Azure Compute Gallery

To delete the VM Application resource, you need to first delete all its versions. Deleting the application version causes deletion of the application version resource from Azure Compute Gallery and all its replicas. The application blob in Storage Account used to create the application version is unaffected.

> ⚠ **Warning**
>
> - Deleting the application version causes subsequent PUT operations on VMs using that version to fail. To prevent this failure, use `latest` keyword as the version number in the `applicationProfile` instead of hard coding the version number .
> - Deleting the VM application that is deployed on any VM or Virtual Machine Scale Sets causes subsequent PUT operations on these resources to fail (for example, update, scale, or reimage). Before deleting, ensure all VMs/Virtual Machine Scale Sets instances stop using the application by removing it from their applicationProfile.
> - To prevent accidental deletion, set `safetyProfile/allowDeletionOfReplicatedLocations` to `false` while publishing the version and apply an Azure Resource Manager lock (CanNotDelete or ReadOnly) on the VM application resource.

Portal

1. Sign in to the Azure portal.
2. Search for Azure Compute Gallery and open the target gallery.
3. Select the VM application you want to remove.
4. Select one or more versions, which you want to delete.
5. To delete the VM application, first delete all the versions. Then click delete (on top of the blade).
6. Monitor Notifications for completion. If deletion is blocked, remove any locks and ensure no VM or Virtual Machine Scale Sets references the application.



Manage deployed VM Application on Azure VM and Virtual Machine Scale Sets

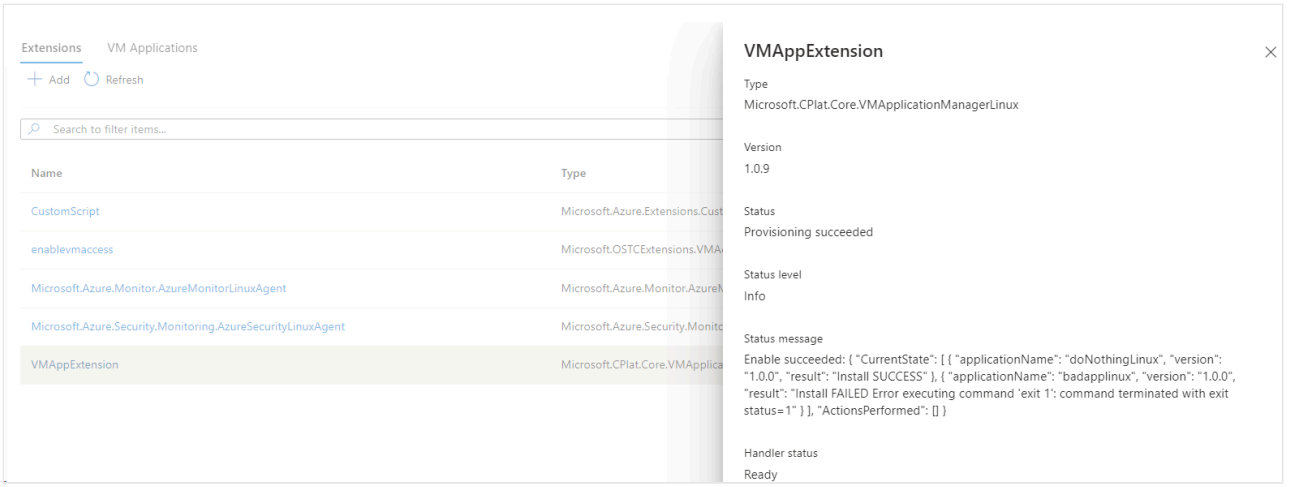
This section explains how to view deployed application details and monitor deployed applications across infrastructure. It also talks about how to update and delete **deployed VM Applications** on Azure VMs and Virtual Machine Scale Sets.

View deployed VM Applications & their state

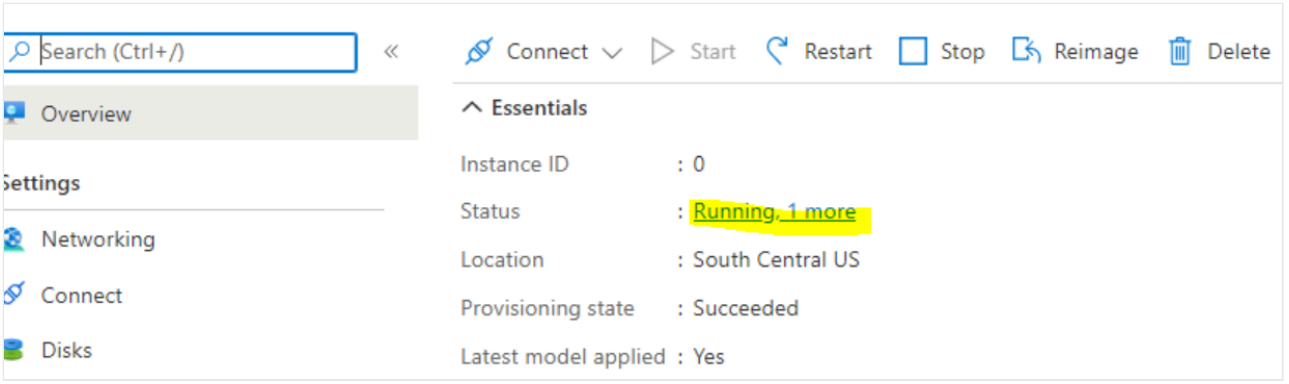
Azure uses VMAppExtension to deploy, monitor, and manage VM Applications on the VM. Therefore, provisioning state of the deployed VM Application is described in the status of the VMAppExtension.

Portal

To show the VM application status, go to the **Extensions + applications** tab under settings and check the status of the VMAppExtension:



To show the VM application status for a Virtual Machine Scale Sets, go to the Azure portal Virtual Machine Scale Sets page. In the Instances section, select one of the instances. Then go to **Extensions + Applications** tab under settings and check the status of the **VMAppExtension**:



View logs of application installation using Run command

When Azure VM Applications downloads and installs the application on Azure VM or Virtual Machine Scale Sets, it pipes all stdout results to `stdout` file within the application repository. Customers can enable verbose logging for the application installation and write custom logs using the `installScript`. Customers can then manually check the `stdout` and `stderr` file or use Runcommand to get the file content.

Use the following PowerShell script in your managed run command. Update the `appName` and `appVersion` variables for your application

Windows

azure-powershell

```
$appName = "vm-application-name"        # VM Application definition name
$appVersion = "1.0.0"                   # VM Application version name

$VMAppManagerVersion = "1.0.16"         # Version of the
VMApplicationManagerWindows extension on the VM
$StdoutFilePath =
"C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\$VMAppMan
agerVersion\Downloads\$appName\$appVersion\stdout"
$StderrFilePath =
"C:\Packages\Plugins\Microsoft.CPlat.Core.VMApplicationManagerWindows\$VMAppMan
agerVersion\Downloads\$appName\$appVersion\stderr"

Write-Host "`n=== Contents of stdout ==="
Get-Content -Path $StdoutFilePath

Write-Host "`n=== Contents of stderr ==="
Get-Content -Path $StderrFilePath
```

Execute the run command using the scripts and get the application install logs.

Portal

1. Open the Azure portal and navigate to your VM.
2. Under **Operations**, select **Run command**.
3. Choose **RunPowerShellScript** from the list.
4. Enter your PowerShell script in the editor.
5. Select **Run** to execute the script.
6. View the output in the **Output** section. If the script fails, check the **Error** section for details.

For more information, see Run PowerShell scripts in your Windows VM by using Run Command.

View all deployed VM applications using Azure Resource Graph

Azure resource graph query can be used to view all deployed VM applications and their properties across all VMs and Virtual Machine Scale Sets. It provides a programmatic way to view application inventory, state, and deployed versions at high scale. Use this method for integrating with dashboards and custom reports.

Kusto

```
resources
| where type == "microsoft.compute/virtualmachines" or type ==
"microsoft.compute/virtualmachinescalesets"
| where properties has "applicationProfile"
| extend resourceType = iff(type == "microsoft.compute/virtualmachines", "VM",
"VMSS")
| extend applications = iff(resourceType == "VM",
parse_json(properties["applicationProfile"]["galleryApplications"]),
parse_json(properties["virtualMachineProfile"]["applicationProfile"]
["galleryApplications"]))
| mv-expand applications
| extend enableAutomaticUpgrade = applications["enableAutomaticUpgrade"]
| extend packageReferenceId = applications["packageReferenceId"]
| extend treatFailureAsDeploymentFailure =
applications["treatFailureAsDeploymentFailure"]
| parse packageReferenceId with "/subscriptions/" publisherSubcriptionId
"/resourceGroups/" publisherResourceGroup "/providers/Microsoft.Compute/galleries/"
galleryName "/applications/" appName "/versions/" version
| project tenantId, subscriptionId, resourceGroup, resourceName = name, type,
location, appName, version, enableAutomaticUpgrade,
treatFailureAsDeploymentFailure, galleryName, publisherSubcriptionId,
publisherResourceGroup, properties
```

Audit required VM Application using Azure Policy

Azure Policy helps enforce governance by auditing whether required VM applications are deployed across your VMs and Virtual Machine Scale Sets. You can create and assign custom policies to check compliance and ensure that specific applications are present on your infrastructure.

For step-by-step instructions on how to audit VM application deployment using Azure Policy, see Audit required VM applications using Azure Policy.

Update the deployed VM Application

To update a deployed VM application, modify the `applicationProfile` to reference a newer version or change deployment settings such as `treatFailureAsDeploymentFailure` or `order`.

REST

Update the VM application version or settings on a single VM:

rest

```
PATCH
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{res
```

```
ourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmName}?api-
version=2024-03-03

Body
{
    "properties": {
        "applicationProfile": {
            "galleryApplications": [
                {
                    "packageReferenceId":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/M
icrosoft.Compute/galleries/{galleryName}/applications/{applicationName}/version
s/{newVersion}",
                    "treatFailureAsDeploymentFailure": true,
                    "enableAutomaticUpgrade": true
                }
            ]
        }
    }
}
```

Update VM application on a Virtual Machine Scale Sets (model):

rest

```
PATCH
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{res
ourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{vmssName}?
api-version=2024-03-03

Body
{
    "properties": {
        "virtualMachineProfile": {
            "applicationProfile": {
                "galleryApplications": [
                    {
                        "packageReferenceId":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/M
icrosoft.Compute/galleries/{galleryName}/applications/{applicationName}/version
s/{newVersion | latest}",
                        "treatFailureAsDeploymentFailure": true,
                        "order": 2
                    }
                ]
            }
        }
    }
}
```

Apply the change to existing Virtual Machine Scale Sets instances (required when upgradePolicy.mode is Manual):

rest

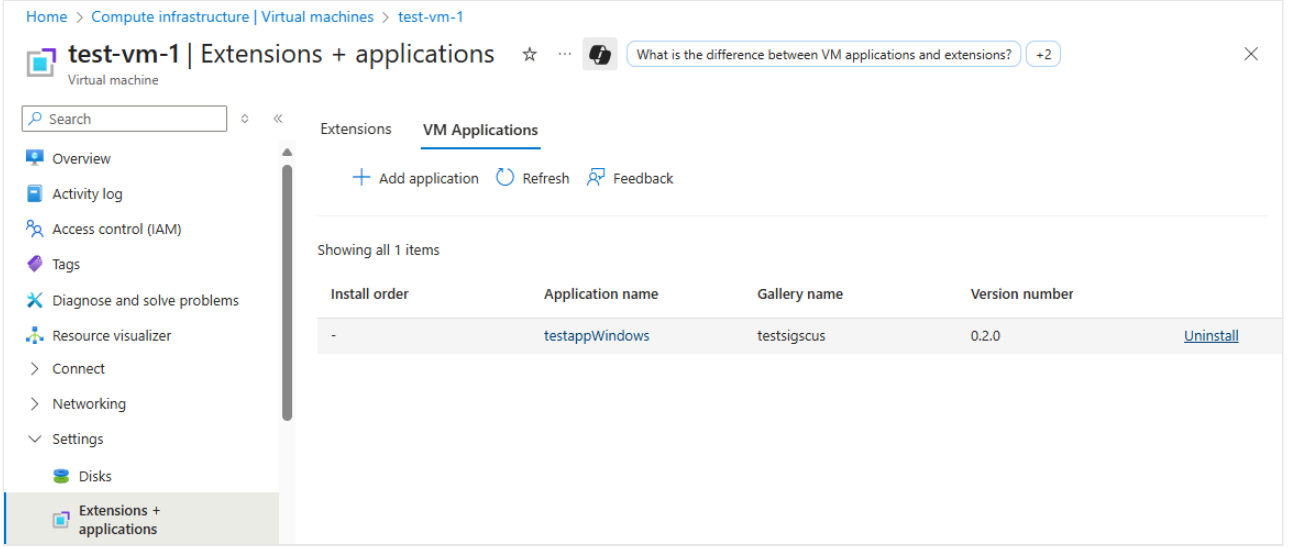
```
POST
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{res
ourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{vmssName}/
updateInstances?api-version=2024-03-03

Body
{
    "instanceIds": ["*"]
}
```

> **Tip**
>
> Use `latest` as the version identifier in `packageReferenceId` to automatically deploy the newest published version without manually updating deployments.

Remove the VM Application from Azure VM or Virtual Machine Scale Sets

Portal

1. Open the Azure portal and go to the target virtual machine (VM) or Virtual Machine Scale Sets.
2. In Settings, select **Extensions + applications**, then select the **VM Applications** tab.
3. Click uninstall button on the VM Application and Save.
4. Track progress in Notifications or check Instance view for the VMAppExtension status.



Next steps

- Learn more about Azure VM Applications.
- Learn to create Azure VM applications.

Last updated on 03/06/2026

Create packages for VM applications

Summarize this article for me

To create a VM Application, you need application package and scripts to properly install, update, and delete the application. This article explains how to create packages and scripts for different operating systems and formats that you can use as VM application payloads.

Package and Install VM Applications on Linux

Third party applications for Linux can be packaged in a few ways. Let's explore how to handle creating the install commands for some of the most common.

.tar and .gz files

These files are compressed archives and can be extracted to a desired location. Check the installation instructions for the original package in case they need to be extracted to a specific location. If .tar.gz file contains source code, see the instructions for the package for how to install from source.

Example to install command to install `golang` on a Linux machine:

Bash

```
sudo tar -C /usr/local -xzf go_linux
```

Example remove command:

Bash

```
sudo rm -rf /usr/local/go
```

Creating application packages using `.deb`, `.rpm`, and other platform specific packages for VMs with restricted internet access

You can download individual packages for platform specific package managers, but they usually don't contain all the dependencies. For these files, you must also include all dependencies in the application package, or have the system package manager download the dependencies through the repositories that are available to the VM. If you're working with a VM with restricted internet access, you must package all the dependencies yourself.

Figuring out the dependencies can be a bit tricky. There are third party tools that can show you the entire dependency tree.

Following process shows how to identify application dependencies, download them and package them together for different linux OS flavors.

Ubuntu

In Ubuntu, you can run `sudo apt show <package_name> | grep Depends` to show all the packages that are installed when executing the `sudo apt-get install <packge_name>` command. Then you can use that output to download all `.deb` files to create an archive that can be used as the application package.

To create a VM Application package for installing PowerShell on Ubuntu, perform the following steps:

1. Run the following commands to enable the repository to download PowerShell and to identify package dependencies on a new Ubuntu VM

Bash

```
# Download the Microsoft repository GPG keys
wget -q "https://packages.microsoft.com/config/ubuntu/$(lsb_release -
rs)/packages-microsoft-prod.deb"
# Register the Microsoft repository GPG keys
sudo dpkg -i packages-microsoft-prod.deb
sudo rm -rf packages-microsoft-prod.deb
sudo apt update
sudo apt show powershell | grep Depends
```

2. Check the output of the line **Depends** which lists the following packages:

Output

```
Depends: libc6, lib32gcc-s1, libgssapi-krb5-2, libstdc++6, zlib1g,
libicu72|libicu71|libicu70|libicu69|libicu68|libicu67|libicu66|libicu65|libicu6
3|libicu60|libicu57|libicu55|libicu52, libssl3|libssl1.1|libssl1.0.2|libssl1.
```

3. Download each of these files using `sudo apt-get download <package_name>` and create a tar compressed archive with all files.

- Ubuntu 18.04:

Bash

```
mkdir /tmp/powershell
cd /tmp/powershell
sudo apt-get download libc6
sudo apt-get download lib32gcc
sudo apt-get download libgssapi-krb5-2
sudo apt-get download libstdc++6
sudo apt-get download zlib1g
sudo apt-get download libssl1.1
sudo apt-get download libicu60
sudo apt-get download powershell
sudo tar -cvzf powershell.tar.gz *.deb
```

- Ubuntu 20.04:

Bash

```
mkdir /tmp/powershell
cd /tmp/powershell
sudo apt-get download libc6
sudo apt-get download lib32gcc-s1
sudo apt-get download libgssapi-krb5-2
sudo apt-get download libstdc++6
sudo apt-get download zlib1g
sudo apt-get download libssl1.1
sudo apt-get download libicu66
sudo apt-get download powershell
sudo tar -cvzf powershell.tar.gz *.deb
```

- Ubuntu 22.04:

Bash

```
mkdir /tmp/powershell
cd /tmp/powershell
sudo apt-get download libc6
sudo apt-get download lib32gcc-s1
sudo apt-get download libgssapi-krb5-2
sudo apt-get download libstdc++6
sudo apt-get download zlib1g
sudo apt-get download libssl3
sudo apt-get download libicu70
sudo apt-get download powershell
sudo tar -cvzf powershell.tar.gz *.deb
```

- Ubuntu 24.04:

Bash

```
mkdir /tmp/powershell
cd /tmp/powershell
```

```
sudo apt-get download libc6
sudo apt-get download lib32gcc-s1
sudo apt-get download libgssapi-krb5-2
sudo apt-get download libstdc++6
sudo apt-get download zlib1g
sudo apt-get download libssl3t64
sudo apt-get download libicu74
sudo apt-get download powershell
sudo tar -cvzf powershell.tar.gz *.deb
```

4. This tar archive is the application package file.

- The install command in this case is:

Bash

```
sudo tar -xvzf powershell.tar.gz && sudo dpkg -i *.deb
```

- And the remove command is:

Bash

```
sudo apt remove powershell
```

Use `sudo apt autoremove` instead of explicitly trying to remove all the dependencies. You may have installed other applications with overlapping dependencies, and in that case, an explicit remove command would fail.

In case you don't want to resolve the dependencies yourself, and `apt` is able to connect to the repositories, you can install an application with just one `.deb` file and let `apt` handle the dependencies.

Example install command:

Bash

```
dpkg -i <package_name> || apt --fix-broken install -y
```

Creating VM Applications on Windows

Most third party applications in Windows are available as .exe or .msi installers. Some are also available as extract and run zip files. Let us look at the best practices for each of them.

.exe installer

Installer executables typically launch a user interface (UI) and require someone to select through the UI. If the installer supports a silent mode parameter, it should be included in your installation string.

Cmd.exe also expects executable files to have the extension `.exe`, so you need to rename the file to have the `.exe` extension.

If I want to create a VM Application package for `myApp.exe`, which ships as an executable, my VM Application is called 'myApp', so I write the command assuming the application package is in the current directory:

terminal

```
"move .\\myApp .\\myApp.exe & myApp.exe /S -config myApp_config"
```

If the installer executable file doesn't support an uninstall parameter, you can sometimes look up the registry on a test machine to know where the uninstaller is located.

In the registry, the uninstall string is stored in `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\<installed application name>\UninstallString` so I would use the contents as my remove command:

terminal

```
'\"C:\\Program Files\\myApp\\uninstall\\helper.exe\" /S'
```

.msi installer

For command line execution of `.msi` installers, the commands to install or remove an application should use `msiexec`. Typically, `msiexec` runs as its own separate process and `cmd` doesn't wait for it to complete, which can lead to problems when installing more than one VM application. The `start` command can be used with `msiexec` to ensure that the installation completes before the command returns. For example:

terminal

```
start /wait %windir%\\system32\\msiexec.exe /i myapp /quiet /forcerestart /log
myapp_install.log
```

Example remove command:

```
terminal
start /wait %windir%\\system32\\msiexec.exe /x myapp /quiet /forcerestart /log
myapp_uninstall.log
```

Typically, the `start` command would be called within a batch script. If used with the `/wait` parameter, the calling script is paused until the called process terminates. Once complete, the batch script would check for the `errorlevel` variable set by the `start` command and exit as follows:

```
batch
start /wait %windir%\\system32\\msiexec.exe /i myapp /quiet /forcerestart /log
myapp_install.log
if %errorlevel% neq 0 exit /b %errorlevel%
...
```

Zipped files

For .zip or other zipped files, rename and unzip the contents of the application package to the desired destination.

Example install command:

```
terminal
rename myapp myapp.zip && mkdir C:\myapp && powershell.exe -Command "Expand-Archive
-path myapp.zip -destinationpath C:\myapp"
```

Example remove command:

```
terminal
rmdir /S /Q C:\\myapp
```

Next steps

- Learn more about Azure VM Applications.
- Learn how to create and deploy VM application packages.
- Learn how to manage and delete Azure VM Applications.

Last updated on 03/06/2026

Publish Azure VM applications using managed identity

Summarize this article for me

This article explains how to use managed identity with Azure Compute Gallery to securely publish Azure VM applications using blob URLs, eliminating the need for SAS URLs or publicly accessible storage accounts.

Overview

Deploying applications through Azure VM applications requires you to upload the application package to a storage account and provide a URL to the blob. Azure Compute Gallery uses this URL to access the blob and publish it as a VM application version.

Without managed identity, you have two options to provide blob access:

- **Blob URL** - Requires the storage account to enable anonymous access.
- **SAS URL** - The SAS token is stored as plain text in the application version resource, and you must manage token expiration and regeneration.

Both approaches are insecure as blob URLs expose your storage account to the public internet, and SAS URLs, which are secrets must be stored in plain text.

Managed identity for Azure Compute Gallery

A managed identity is a security feature that lets an Azure service prove its identity to other Azure services without storing passwords, keys, or tokens in your code or configuration. Instead of you managing credentials, Microsoft Entra ID handles authentication automatically.

By attaching a user-assigned managed identity to your Azure Compute Gallery, you give the gallery permission to access blobs in your storage account on your behalf. The gallery presents this identity when it needs to read your application package, and Microsoft Entra ID verifies the identity and grants access. Because Azure Compute Gallery is a trusted Microsoft service, this access works even when the storage account is behind a virtual network.

This approach enables you to:

- Restrict storage account access to specific virtual networks.
- Disable shared key access on the storage account and use plain blob URLs.
- Ensure access credentials like SAS URLs aren't stored as plain text.

How it works

At a high level, the authentication and access flow work as follows:

1. You upload your application package to a blob in your storage account.
2. You create a user-assigned managed identity and grant it the **Storage Blob Data Contributor** role on the storage account.
3. You attach the managed identity to your Azure Compute Gallery.
4. When you publish a VM application version, you provide a plain blob URL instead of a SAS URL.
5. Azure Compute Gallery attempts to access the blob using the blob URL directly.
6. Microsoft Entra ID verifies the identity and grants the gallery access to the blob.

Prerequisites

- An Azure Compute Gallery
- A storage account with your VM application package uploaded to a blob container.

Step 1: Create a user-assigned managed identity

Create a user-assigned managed identity that Azure Compute Gallery uses to authenticate with your storage account.

For detailed steps, see Create a user-assigned managed identity.

Step 2: Assign storage access permission to the identity

Assign the **Storage Blob Data Contributor** role to the managed identity so it can access the application package in your storage account.

For detailed steps, see Assign Azure roles using the Azure portal. When assigning the role:

1. Select **Storage Blob Data Contributor** as the role.
2. Assign access to **Managed identity**.
3. Select the user-assigned managed identity you created in the previous step.

Step 3: Attach managed identity to Azure Compute Gallery

Update your Azure Compute Gallery to use the user-assigned managed identity. Portal experience isn't currently available for this step.

Using REST API

Use the Galleries - Create Or Update API to attach the managed identity to your gallery.

HTTP

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/galleries/{galleryName}?api-version=2025-04-01

{
    "location": "<location>",
    "identity": {
        "type": "UserAssigned",
        "userAssignedIdentities": {
            "/subscriptions/<subscription-id>/resourceGroups/<resource-group>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<identity-name>": {}
        }
    }
}
```

Step 4: Get the blob URL for your application package

Retrieve the blob URL for the application package stored in your storage account. The URL format is:

```
https://<storage-account-name>.blob.core.windows.net/<container-name>/<blob-name>
```

Step 5: Publish a VM application version using the blob URL

Use the blob URL in the `mediaLink` property of the publishing profile when creating or updating your VM application version. Azure Compute Gallery uses the attached managed identity to authenticate and access the blob.

For detailed steps on creating a VM application version, see Create a VM application version.

Technical details

- **Authentication flow** - Azure Compute Gallery first attempts to access the blob using the provided URL. If direct access fails due to insufficient permissions, the gallery falls back to the attached managed identity for authentication.
- **Trusted service access** - Azure Compute Gallery is a trusted Microsoft service. The managed identity provides trusted access that bypasses storage account virtual network restrictions, so the gallery doesn't need to be inside a virtual network to reach a network-restricted storage account.

Limitations

- Managed identity access is supported only when **publishing** a VM application version to Azure Compute Gallery. It can't be used when deploying a VM application to a VM or Virtual Machine Scale Sets.
- Portal experience for attaching managed identity to the gallery isn't currently available. Use the REST API, Azure CLI, or Azure PowerShell.

Related content

- VM applications overview
- Create a VM application version
- Azure Compute Gallery overview
- What are managed identities for Azure resources?
- Grant access to trusted Azure services

Last updated on 03/06/2026

Govern and enforce compliance for Azure VM Applications using Azure Policy

Summarize this article for me

Azure VM Applications let you package, version, and deliver software to Azure Virtual Machines (VMs) and Virtual Machine Scale Sets from an Azure Compute Gallery.

Using Azure Policy with VM Applications enables customers and admin teams to:

- Monitor presence of required VM Applications across Virtual Machines & Virtual Machines Scale Sets.
- Inject required VM Applications across all Virtual Machines & Virtual Machine Scale Sets.
- Enforce consistent configuration and best practices across all Virtual Machines and Virtual Machine Scale Sets for improved reliability and security.

This guide shows how to:

- Govern required VM application for compliance monitoring.
- Enforce required VM Application for driving consistency and security.

Prerequisites

- An Azure Compute Gallery with the published VM Application and versions to enforce. See create and manage VM Applications.
  - Ensure the version is replicated to all regions where the application presence is required.
  - Ensure the gallery is shared to all subscriptions requiring access to the VM Application.
- Permissions:
  - Policy Contributor to create/assign policies. See policy definition structure and assignments.

Set up compliance monitor to govern required VM applications

Azure Policy with `audit` effect can be used to monitor presence of specific VM Applications across Azure Virtual Machine and Virtual Machine Scale Sets. Admin teams can use this policy to

- Setup compliance monitors for VM Applications (view compliant vs noncompliant Virtual Machines and Virtual Machine Scale Sets)
- Measuring rollout progress of software packaged as a VM Application.

1. Create a custom policy definition

Create a new custom policy using the following definition. This policy checks whether specific VM Applications are present on Virtual Machines and Virtual Machine Scale Sets, and reports compliance status.

Policy

This policy monitors compliance of all Virtual Machines, Virtual Machines Scale Sets across linux and windows. Linux apps on windows VMs and windows apps on linux VMs are considered as compliant resources.

JSON

```
{
    "mode": "Indexed",
    "policyRule": {
        "if": {
            "anyOf": [
                {
                    "allOf": [
                        { "field": "type", "equals":
"Microsoft.Compute/virtualMachines" },
                        {"field":
"Microsoft.Compute/virtualMachines/storageProfile.osDisk.osType", "equals": "
[parameters('osType')]" },
                        {
                            "count": {
                                "field":
"Microsoft.Compute/virtualMachines/applicationProfile.galleryApplications[*]",
                                "where": {
                                    "allOf": [
                                        {
                                            "field":
"Microsoft.Compute/virtualMachines/applicationProfile.galleryApplications[*].pa
ckageReferenceId",
                                            "contains": "[concat('/galleries/',
parameters('galleryName'), '/applications/', parameters('applicationName'),
'/')]"
                                        }
                                    ]
                                }
                            },
                            "equals": 0
                        }
```

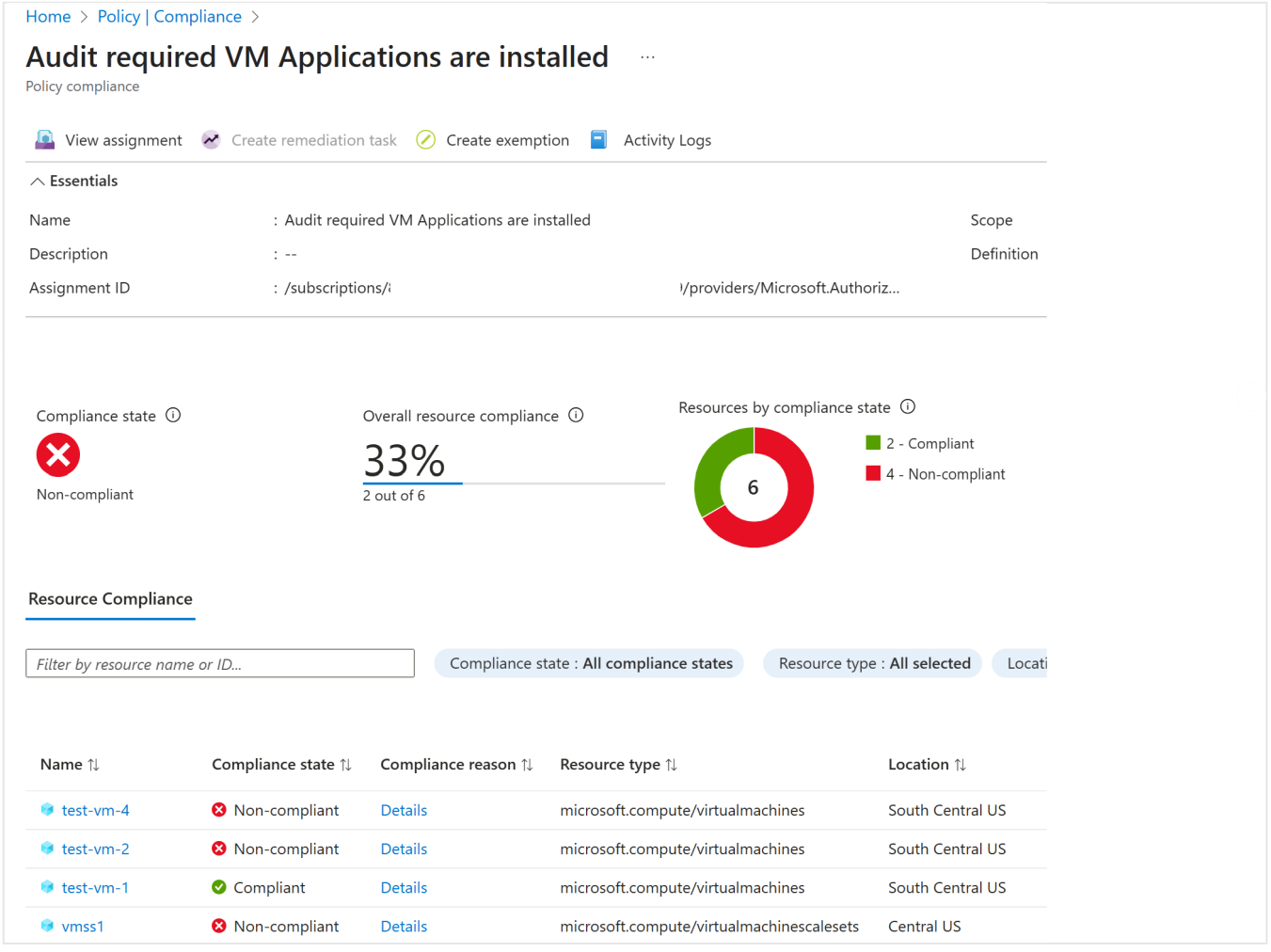
```
                    ]
                },
                {
                    "allOf": [
                        { "field": "type", "equals":
"Microsoft.Compute/virtualMachineScaleSets" },
                        {"field":
"Microsoft.Compute/virtualMachineScaleSets/virtualMachineProfile.storageProfile
.osDisk.osType", "equals": "[parameters('osType')]" },
                        {
                            "count": {
                                "field":
"Microsoft.Compute/virtualMachineScaleSets/virtualMachineProfile.applicationPro
file.galleryApplications[*]",
                                "where": {
                                    "allOf": [
                                        {
                                            "field":
"Microsoft.Compute/virtualMachineScaleSets/virtualMachineProfile.applicationPro
file.galleryApplications[*].packageReferenceId",
                                            "contains": "[concat('/galleries/',
parameters('galleryName'), '/applications/', parameters('applicationName'),
'/')]"
                                        }
                                    ]
                                }
                            },
                            "equals": 0
                        }
                    ]
                }
            ]
        },
        "then": { "effect": "audit" }
    },
    "parameters": {
        "galleryName": {
            "type": "String",
            "metadata": { "description": "Name of the Azure Compute Gallery
containing the VM Application." }
        },
        "applicationName": {
            "type": "String",
            "metadata": { "description": "Name of the VM Application to audit
for presence." }
        },
        "osType": {
            "type": "String",
            "allowedValues": [ "Windows", "Linux" ],
            "metadata": { "description": "OS type of the VM Application
(Windows or Linux)." }
        }
    }
}
```

2. Assign the policy and view compliance

Assign the newly created policy to start monitoring the VM application and generate compliance score. All Virtual Machines and Virtual Machine Scale Sets within the assigned scope are monitored for compliance.

Create separate assignments per VM application for granular and accurate monitoring.

Once the policy is assigned, all existing resources are evaluated and displayed on compliance monitor. Noncompliant resources are missing the VM Application defined in the policy. Resources without `applicationProfile` are also counted as noncompliant. Newly created or updated resources may take a few minutes to appear in evaluation cycles.



Common adjustments

- **Audit multiple required applications**: Create a separate policy (one per application) or update the policy by adding other applications under `allOf` parameter.

- **Block creation of Virtual Machines without the required applications**: Change effect to `deny` to prevent noncompliant creations.
- **Limit policy scope to Virtual Machines or Virtual Machine Scale Sets**: Remove the unused branch from `anyOf` within `policyRule`.
- **Limit policy scope by OS type**: Check `osType` of `storageProfile` within `policyRule` to filter based on Window / Linux OS:

  JSON

  ```
  { "field": "Microsoft.Compute/virtualMachines/storageProfile.osDisk.osType",
  "equals": "[parameters('osType')]" }
  ```

- **Limit policy scope by OS Image**: Check `offer` and `sku` within `policyRule` to filter based on image:

  JSON

  ```
  { "field":
  "Microsoft.Compute/virtualMachines/storageProfile.imageReference.offer",
  "equals": "[parameters('imageOffer')]" },
  { "field":
  "Microsoft.Compute/virtualMachines/storageProfile.imageReference.sku",
  "equals": "[parameters('imageSku')]" }
  ```

Inject VM Applications on Virtual Machine and Virtual Machine Scale Sets

Azure Policy with `modify` effect injects VM applications while creating new Virtual Machine and Virtual Machine Scale Sets. Remediation tasks can be used to modify existing resources and inject the VM Application.

Pre-requisite

- Managed identity on the policy assignment with **Virtual Machine Contributor** role, so remediation can modify resources. See remediation identity requirements.

1. Create a custom policy definition

Create a new custom policy using the policy definition. This policy checks for the presence of VM applications while creating Virtual Machines or Virtual Machine Scale Sets and appends the VM application if it doesn't exist. To modify multiple resource types (Virtual Machine and Virtual Machine Scale Sets), it's recommended to create different policies per resource type.

VM

JSON

```
{
    "displayName": "Inject VM Application into single instance VMs",
    "policyType": "Custom",
    "mode": "Indexed",
    "description": "Appends a VM Application reference to applicationProfile
for single instance VMs if the application is not already present. Filters by
OS type to prevent cross-platform injection.",
    "parameters": {
      "subscriptionId": {
        "type": "String",
        "metadata": {
          "description": "Subscription ID where the Compute Gallery resides."
        }
      },
      "resourceGroup": {
        "type": "String",
        "metadata": {
          "description": "Resource group of the Compute Gallery."
        }
      },
      "galleryName": {
        "type": "String",
        "metadata": {
          "description": "Name of the Compute Gallery."
        }
      },
      "applicationName": {
        "type": "String",
        "metadata": {
          "description": "VM Application name."
        }
      },
      "applicationVersion": {
        "type": "String",
        "defaultValue": "latest",
        "metadata": {
          "description": "VM Application version. Defaults to latest."
        }
      },
      "osType": {
        "type": "String",
        "allowedValues": [
```

```
          "Windows",
          "Linux"
        ],
        "metadata": {
          "description": "Target OS type. Ensures the application is only
injected onto VMs with a matching OS."
        }
      }
    },
    "policyRule": {
      "if": {
        "allOf": [
          {
            "field": "type",
            "equals": "Microsoft.Compute/virtualMachines"
          },
          {
            "field":
"Microsoft.Compute/virtualMachines/storageProfile.osDisk.osType",
            "equals": "[parameters('osType')]"
          },
          {
            "count": {
              "field":
"Microsoft.Compute/virtualMachines/applicationProfile.galleryApplications[*]",
              "where": {
                "field":
"Microsoft.Compute/virtualMachines/applicationProfile.galleryApplications[*].pa
ckageReferenceId",
                "contains": "[concat('/galleries/', parameters('galleryName'),
'/applications/', parameters('applicationName'), '/')]"
              }
            },
            "equals": 0
          }
        ]
      },
      "then": {
        "effect": "modify",
        "details": {
          "roleDefinitionIds": [
            "/providers/Microsoft.Authorization/roleDefinitions/9980e02c-c2be-
4d73-94e8-173b1dc7cf3c"
          ],
          "operations": [
            {
              "operation": "add",
              "field":
"Microsoft.Compute/virtualMachines/applicationProfile.galleryApplications",
              "value": {
                "packageReferenceId": "[concat('/subscriptions/',
parameters('subscriptionId'), '/resourceGroups/', parameters('resourceGroup'),
'/providers/Microsoft.Compute/galleries/', parameters('galleryName'),
'/applications/', parameters('applicationName'), '/versions/',
parameters('applicationVersion'))]",
```

```
                "order": 1,
                "treatFailureAsDeploymentFailure": true
              }
            }
          ]
        }
      }
    }
}
```

2. Assign the policy

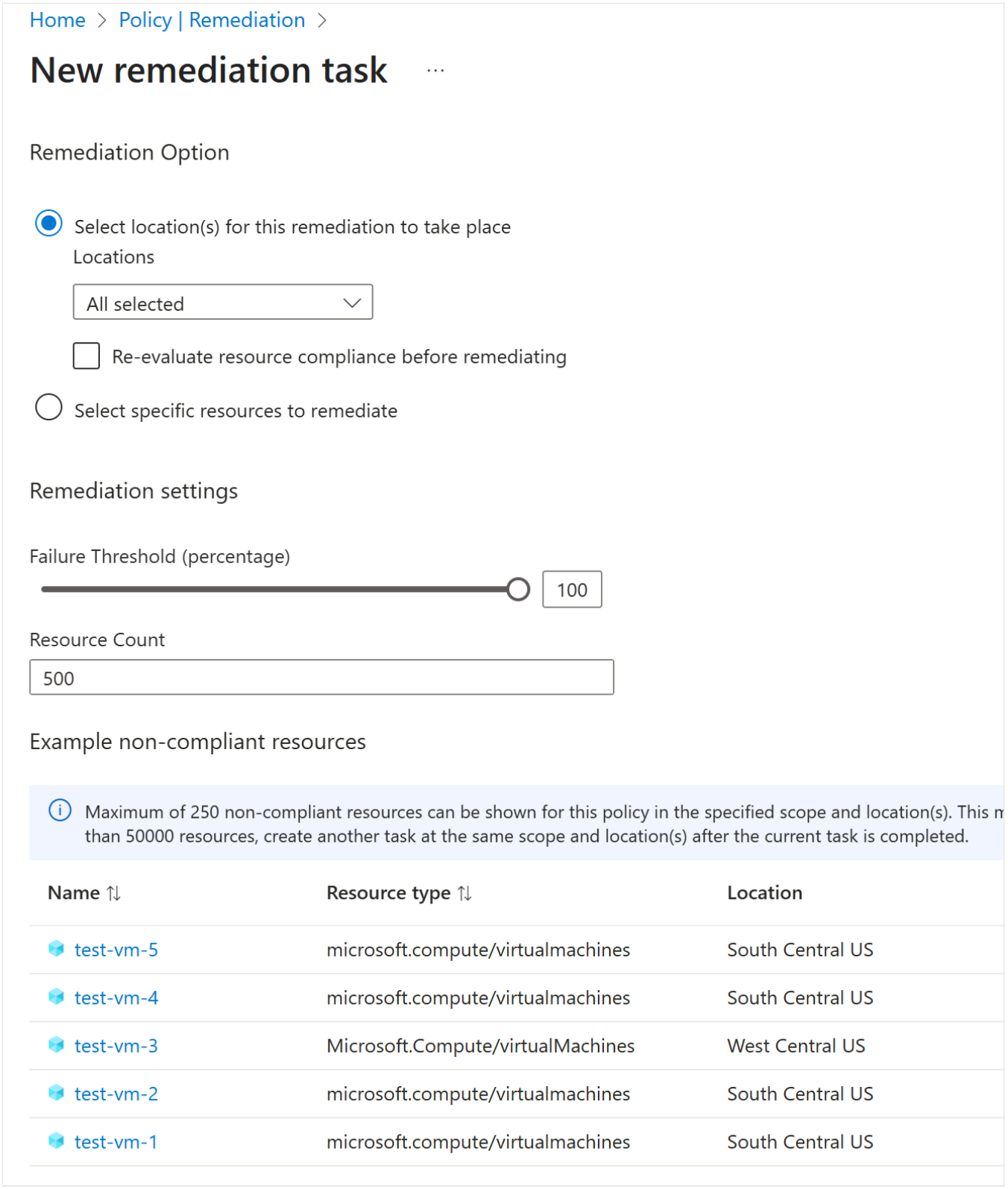
Assign the newly created policy to start generating compliance score. All Virtual Machines and Virtual Machine Scale Sets within the assigned scope are monitored for compliance. Once the policy is assigned, all existing resources are evaluated and displayed on compliance monitor.

Creating new Virtual Machines and Virtual Machine Scale Sets resource triggers this policy and modifies the applicationProfile of the resource to inject the application.

3. Create Remediation Task and modify existing resources

To modify existing resources, create a new Remediation tasks.

> **Note**
>
> Gradually remediate noncompliant resources for higher availability and failure resiliency. Create multiple remediation tasks, each scoped to one or more regions.



Common adjustments

The following examples show more conditions you can add to the `allOf` block within `policyRule` to narrow the policy scope.

- **Limit policy scope by OS image**: Filter based on a specific image offer and SKU:

  JSON

```
{ "field":
"Microsoft.Compute/virtualMachines/storageProfile.imageReference.offer",
"equals": "[parameters('imageOffer')]" },
{ "field":
"Microsoft.Compute/virtualMachines/storageProfile.imageReference.sku",
"equals": "[parameters('imageSku')]" }
```

- **Limit policy scope by region**: Target specific Azure regions:

  JSON

  ```
  { "field": "location", "in": "[parameters('allowedLocations')]" }
  ```

- **Limit policy scope by resource group or subscription**: To reduce the scope for remediation, assign the policy to a specific resource group or subscription. Use the assignment scope when assigning the policy rather than modifying the policy definition. For more granular control, use exclusions to omit specific resource groups from the assignment.

Next steps

- Learn more about Azure VM Applications.
- Learn how to create and deploy VM application packages.
- Learn how to manage and delete Azure VM Applications.
- Learn how to create application package for VM Applications

---

Last updated on 03/06/2026

Find Azure Marketplace image information using the Azure CLI

Article • 08/22/2024

> **⊗ Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This topic describes how to use the Azure CLI to find VM images in the Azure Marketplace. Use this information to specify a Marketplace image when you create a VM programmatically with the CLI, Resource Manager templates, or other tools.

You can also browse available images and offers using the Azure Marketplace or Azure PowerShell.

Terminology

A Marketplace image in Azure has the following attributes:

- **Publisher**: The organization that created the image. Examples: Canonical, RedHat, SUSE.
- **Offer**: The name of a group of related images created by a publisher. Examples: 0001-com-ubuntu-server-jammy, RHEL, sles-15-sp3.
- **SKU**: An instance of an offer, such as a major release of a distribution. Examples: 22_04-lts-gen2, 8-lvm-gen2, gen2.
- **Version**: The version number of an image SKU.

These values can be passed individually or as an image *URN*, combining the values separated by the colon (:). For example: *Publisher*:*Offer*:*Sku*:*Version*. You can replace the version number in the URN with `latest` to use the latest version of the image.

If the image publisher provides extra license and purchase terms, then you must accept those terms before you can use the image. For more information, see Check the purchase plan information.

List popular images

You can run the az vm image list --all to see all of the images available to you, but it can take several minutes to produce the entire list. A faster option is the use `az vm image list`, without the `--all` option, to see a list of popular VM images in the Azure Marketplace. For example, run the following command to display a cached list of popular images in table format:

Azure CLI

```
az vm image list --output table
```

The output includes the image URN. If you omit the `--all` option, you can see the *UrnAlias* for each image, if available. *UrnAlias* is a shortened version created for popular images like *Ubuntu2204*. The Linux image alias names and their details outputted by this command are:

Output

```
Architecture    Offer                         Publisher              Sku
Urn
UrnAlias                Version
--------------  ----------------------------  ---------------------  ------
--------------------------  ------------------------------------------------
------------------------------  ----------------------  ---------
x64             CentOS                        OpenLogic              8_5-
gen2                            OpenLogic:CentOS:8_5-gen2:latest
CentOS85Gen2            latest
x64             Debian11                      Debian                 11-
backports-gen2                  Debian:debian-11:11-backports-gen2:latest
Debian-11               latest
x64             flatcar-container-linux-free  kinvolk
stable-gen2                       kinvolk:flatcar-container-linux-
free:stable-gen2:latest                         FlatcarLinuxFreeGen2
latest
x64             opensuse-leap-15-4            SUSE                   gen2
SUSE:opensuse-leap-15-4:gen2:latest
OpenSuseLeap154Gen2     latest
x64             RHEL                          RedHat                 8-lvm-
gen2                        RedHat:RHEL:8-lvm-gen2:latest
RHELRaw8LVMGen2         latest
x64             sles-15-sp3                   SUSE                   gen2
SUSE:sles-15-sp3:gen2:latest
SLES                    latest
x64             0001-com-ubuntu-server-jammy  Canonical              22_04-
lts-gen2                    Canonical:0001-com-ubuntu-server-jammy:22_04-
lts-gen2:latest                 Ubuntu2204              latest
```

The Windows image alias names and their details outputted by this command are:

Output

```
Architecture    Offer                         Publisher               Sku
Urn
Alias                    Version
--------------  ---------------------------  ----------------------  ------
---------------------------  ---------------------------------------------
------------------------------ ----------------------  ---------
x64             WindowsServer                 MicrosoftWindowsServer  2022-
datacenter-azure-edition       MicrosoftWindowsServer:WindowsServer:2022-
datacenter-azure-edition:latest      Win2022AzureEdition       latest
x64             WindowsServer                 MicrosoftWindowsServer  2022-
datacenter-azure-edition-core  MicrosoftWindowsServer:WindowsServer:2022-
datacenter-azure-edition-core:latest Win2022AzureEditionCore   latest
x64             WindowsServer                 MicrosoftWindowsServer  2019-
Datacenter                     MicrosoftWindowsServer:WindowsServer:2019-
Datacenter:latest                    Win2019Datacenter         latest
x64             WindowsServer                 MicrosoftWindowsServer  2016-
Datacenter                     MicrosoftWindowsServer:WindowsServer:2016-
Datacenter:latest                    Win2016Datacenter         latest
x64             WindowsServer                 MicrosoftWindowsServer  2012-
R2-Datacenter                  MicrosoftWindowsServer:WindowsServer:2012-R2-
Datacenter:latest                 Win2012R2Datacenter       latest
x64             WindowsServer                 MicrosoftWindowsServer  2012-
Datacenter                     MicrosoftWindowsServer:WindowsServer:2012-
Datacenter:latest                    Win2012Datacenter         latest
```

Find specific images

You can filter the list of images by `--publisher` or another parameter to limit the results.

For example, the following command displays all Debian offers:

Azure CLI

```
az vm image list --offer Debian --all --output table
```

You can limit your results to a single architecture by adding the `--architecture` parameter. For example, to display all Arm64 images available from Canonical:

Azure CLI

```
az vm image list --architecture Arm64 --publisher Canonical --all --output
table
```

Look at all available images

Another way to find an image in a location is to run the az vm image list-publishers, az vm image list-offers, and az vm image list-skus commands in sequence. With these commands, you determine these values:

1. List the image publishers for a location. In this example, we're looking at the *West US* region.

   Azure CLI

   ```
   az vm image list-publishers --location westus --output table
   ```

2. For a given publisher, list their offers. In this example, we add *RedHat* as the publisher.

   Azure CLI

   ```
   az vm image list-offers --location westus --publisher RedHat --output
   table
   ```

3. For a given offer, list their SKUs. In this example, we add *RHEL* as the offer.

   Azure CLI

   ```
   az vm image list-skus --location westus --publisher RedHat --offer RHEL
   --output table
   ```

> **Note**
>
> Canonical has changed the **Offer** names they use for the most recent versions. Before Ubuntu 20.04, the **Offer** name is UbuntuServer. For Ubuntu 20.04 the **Offer** name is `0001-com-ubuntu-server-focal` and for Ubuntu 22.04 it's `0001-com-ubuntu-server-jammy`.

1. For a given publisher, offer, and SKU, show all of the versions of the image. In this example, we add *9_1* as the SKU.

   Azure CLI

   ```
   az vm image list \
       --location westus \
       --publisher RedHat \
   ```

```
    --offer RHEL \
    --sku 9_1 \
    --all --output table
```

Pass this value of the URN column with the `--image` parameter when you create a VM with the az vm create command. You can also replace the version number in the URN with "latest", to use the latest version of the image.

If you deploy a VM with a Resource Manager template, you set the image parameters individually in the `imageReference` properties. See the template reference.

Check the purchase plan information

Some VM images in the Azure Marketplace have extra license and purchase terms that you must accept before you can deploy them programmatically.

To deploy a VM from such an image, you'll need to accept the image's terms the first time you use it, once per subscription. You'll also need to specify *purchase plan* parameters to deploy a VM from that image

To view an image's purchase plan information, run the az vm image show command with the URN of the image. If the `plan` property in the output isn't `null`, the image has terms you need to accept before programmatic deployment.

For example, the Canonical Ubuntu Server 18.04 LTS image doesn't have extra terms, because the `plan` information is `null`:

Azure CLI

```
az vm image show --location westus --urn Canonical:UbuntuServer:18.04-
LTS:latest
```

Output:

Output

```
{
  "dataDiskImages": [],
  "id": "/Subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/Providers/Microsoft.Compute/Locations/westus/Publishers/Canonic
al/ArtifactTypes/VMImage/Offers/UbuntuServer/Skus/18.04-
LTS/Versions/18.04.201901220",
  "location": "westus",
  "name": "18.04.201901220",
  "osDiskImage": {
    "operatingSystem": "Linux"
```

```
  },
  "plan": null,
  "tags": null
}
```

Running a similar command for the RabbitMQ Certified by Bitnami image shows the following `plan` properties: `name`, `product`, and `publisher`. (Some images also have a `promotion code` property.)

Azure CLI

```
az vm image show --location westus --urn bitnami:rabbitmq:rabbitmq:latest
```

Output:

Output

```
{
  "dataDiskImages": [],
  "id": "/Subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/Providers/Microsoft.Compute/Locations/westus/Publishers/bitnami
/ArtifactTypes/VMImage/Offers/rabbitmq/Skus/rabbitmq/Versions/3.7.1901151016
",
  "location": "westus",
  "name": "3.7.1901151016",
  "osDiskImage": {
    "operatingSystem": "Linux"
  },
  "plan": {
    "name": "rabbitmq",
    "product": "rabbitmq",
    "publisher": "bitnami"
  },
  "tags": null
}
```

To deploy this image, you need to accept the terms and provide the purchase plan parameters when you deploy a VM using that image.

Accept the terms

To view and accept the license terms, use the az vm image terms command. When you accept the terms, you enable programmatic deployment in your subscription. You only need to accept terms once per subscription for the image. For example:

Azure CLI

```
az vm image terms show --urn bitnami:rabbitmq:rabbitmq:latest
```

The output includes a `licenseTextLink` to the license terms, and indicates that the value of `accepted` is `true`:

Output


```
{
  "accepted": true,
  "additionalProperties": {},
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/providers/Microsoft.MarketplaceOrdering/offertypes/bitnami/offe
rs/rabbitmq/plans/rabbitmq",
  "licenseTextLink":
"https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_B
ITNAMI%253a24RABBITMQ%253a24RABBITMQ%253a24IGRT7HHPIFOBV3IQYJHEN2O2FGUVXXZ3W
UYIMEIVF3KCUNJ7GTVXNNM23I567GBMNDWRFOY4WXJPN5PUYXNKB2QLAKCHP4IE5GO3B2I.txt",
  "name": "rabbitmq",
  "plan": "rabbitmq",
  "privacyPolicyLink": "https://bitnami.com/privacy",
  "product": "rabbitmq",
  "publisher": "bitnami",
  "retrieveDatetime": "2019-01-25T20:37:49.937096Z",
  "signature":
"XXXXXXLAZIK7ZL2YRV5JYQXONPV76NQJW3FKMKDZYCRGXZYVDGX6BVY45JO3BXVMNA2COBOEYG2
NO76ONORU7ITTRHGZDYNJNXXXXXX",
  "type": "Microsoft.MarketplaceOrdering/offertypes"
}
```


To accept the terms, type:

Azure CLI

```
az vm image terms accept --urn bitnami:rabbitmq:rabbitmq:latest
```

Deploy a new VM using the image parameters

With information about the image, you can deploy it using the `az vm create` command.

To deploy an image that doesn't have plan information, like the latest Ubuntu Server 18.04 image from Canonical, pass the URN for `--image`:

Azure CLI

```
az group create --name myURNVM --location westus
az vm create \
```

```
    --resource-group myURNVM \
    --name myVM \
    --admin-username azureuser \
    --generate-ssh-keys \
    --image Canonical:UbuntuServer:18.04-LTS:latest
```

For an image with purchase plan parameters, like the RabbitMQ Certified by Bitnami image, you pass the URN for `--image` and also provide the purchase plan parameters:

Azure CLI

```
az group create --name myPurchasePlanRG --location westus

az vm create \
   --resource-group myPurchasePlanRG \
   --name myVM \
   --admin-username azureuser \
   --generate-ssh-keys \
   --image bitnami:rabbitmq:rabbitmq:latest \
   --plan-name rabbitmq \
   --plan-product rabbitmq \
   --plan-publisher bitnami
```

If you get a message about accepting the terms of the image, review section Accept the terms. Make sure the output of `az vm image accept-terms` returns the value `"accepted": true,` showing that you've accepted the terms of the image.

Using an existing VHD with purchase plan information

If you have an existing VHD from a VM that was created using a paid Azure Marketplace image, you might need to give the purchase plan information when creating a new VM from that VHD.

If you still have the original VM, or another VM created using the same marketplace image, you can get the plan name, publisher, and product information from it using az vm get-instance-view. This example gets a VM named *myVM* in the *myResourceGroup* resource group and then displays the purchase plan information.

Azure CLI

```
az vm get-instance-view -g myResourceGroup -n myVM --query plan
```

If you didn't get the plan information before the original VM was deleted, you can file a support request. They'll need the VM name, subscription ID and the time stamp of the delete operation.

Once you have the plan information, you can create the new VM using the `--attach-os-disk` parameter to specify the VHD.

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name myNewVM \
  --nics myNic \
  --size Standard_DS1_v2 --os-type Linux \
  --attach-os-disk myVHD \
  --plan-name planName \
  --plan-publisher planPublisher \
  --plan-product planProduct
```

Next steps

To create a virtual machine quickly by using the image information, see Create and Manage Linux VMs with the Azure CLI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Find and use Azure Marketplace VM images with Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article describes how to use Azure PowerShell to find VM images in the Azure Marketplace. You can then specify a Marketplace image and plan information when you create a VM.

You can also browse available images and offers using the Azure Marketplace or the Azure CLI.

Terminology

A Marketplace image in Azure has the following attributes:

- **Publisher**: The organization that created the image. Examples: Canonical, MicrosoftWindowsServer
- **Offer**: The name of a group of related images created by a publisher. Examples: UbuntuServer, WindowsServer
- **SKU**: An instance of an offer, such as a major release of a distribution. Examples: 18.04-LTS, 2019-Datacenter
- **Version**: The version number of an image SKU.

These values can be passed individually or as an image *URN*, combining the values separated by the colon (:). For example: *Publisher*:*Offer*:*Sku*:*Version*. You can replace the version number in the URN with `latest` to use the latest version of the image.

If the image publisher provides other license and purchase terms, then you must accept those before you can use the image. For more information, see Accept purchase plan terms.

Default Images

Powershell offers several pre-defined image aliases to make the resource creation process easier. There are different images for resources with either a Windows or Linux operating system. Several Powershell cmdlets, such as `New-AzVM` and `New-AzVmss`, allow you to input the alias name as a parameter. For example:

PowerShell

```
$rgname = <Resource Group Name>
$location = <Azure Region>
$vmName = "v" + $rgname
$domainNameLabel = "d" + $rgname
$securePassword = <Password> | ConvertTo-SecureString -AsPlainText -Force
$username = <Username>
$credential = New-Object System.Management.Automation.PSCredential
($username, $securePassword)
New-AzVM -ResourceGroupName $rgname -Location $location -Name $vmName -image
Ubuntu2204 -Credential $credential -DomainNameLabel $domainNameLabel
```

The Linux image alias names and their details are:

Output

```
Alias                     Architecture    Offer
Publisher               Sku                                 Urn
Version
-----------------------   -------------  ---------------------------  ----
-----------------  ---------------------------------  --------------------
------------------------------------------------------------ ---------
Debian11                    x64             Debian-11
Debian                  11-backports-gen2                   Debian:debian-
11:11-backports-gen2:latest                                      latest
FlatcarLinuxFreeGen2      x64             flatcar-container-linux-free
kinvolk                 stable                              kinvolk:flatcar-
container-linux-free:stable:latest                             latest
OpenSuseLeap154Gen2       x64             opensuse-leap-15-4            SUSE
gen2                                SUSE:opensuse-leap-15-4:gen2:latest
latest
RHELRaw8LVMGen2           x64             RHEL
RedHat                  8-lvm-gen2                          RedHat:RHEL:8-
lvm-gen2:latest                                                  latest
SLES                      x64             sles-15-sp3                   SUSE
gen2                                SUSE:sles-15-sp3:gen2:latest
latest
Ubuntu2204                x64             0001-com-ubuntu-server-jammy
Canonical               22_04-lts-gen2                      Canonical:0001-
com-ubuntu-server-jammy:22_04-lts-gen2:latest                   latest
```

The Windows image alias names and their details are:

Output

```
Alias                   Architecture    Offer
Publisher               Sku                                 Urn
Version
----------------------- -------------  ---------------------------  ------
---------------  ---------------------------------  --------------------
```

```
---------------------------------------------------------   ---------
Win2022AzureEdition     x64             WindowsServer
MicrosoftWindowsServer  2022-datacenter-azure-edition
MicrosoftWindowsServer:WindowsServer:2022-datacenter-azure-edition:latest
latest
Win2022AzureEditionCore x64             WindowsServer
MicrosoftWindowsServer  2022-datacenter-azure-edition-core
MicrosoftWindowsServer:WindowsServer:2022-datacenter-azure-edition-
core:latest   latest
Win10                   x64             Windows
MicrosoftVisualStudio   Windows-10-N-x64
MicrosoftVisualStudio:Windows:Windows-10-N-x64:latest
latest
Win2019Datacenter       x64             WindowsServer
MicrosoftWindowsServer  2019-Datacenter
MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest
latest
Win2016Datacenter       x64             WindowsServer
MicrosoftWindowsServer  2016-Datacenter
MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest
latest
Win2012R2Datacenter     x64             WindowsServer
MicrosoftWindowsServer  2012-R2-Datacenter
MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest
latest
Win2012Datacenter       x64             WindowsServer
MicrosoftWindowsServer  2012-Datacenter
MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest
latest
```

List images

You can use PowerShell to narrow down a list of images if you want to use a specific image that is not provided by default. Replace the values of the below variables to meet your needs.

1. List the image publishers using Get-AzVMImagePublisher.

   PowerShell

   ```powershell
   $locName="<location>"
   Get-AzVMImagePublisher -Location $locName | Select PublisherName
   ```

2. For a given publisher, list their offers using Get-AzVMImageOffer.

   PowerShell

   ```powershell
   $pubName="<publisher>"
   Get-AzVMImageOffer -Location $locName -PublisherName $pubName | Select
   ```

```
Offer
```

3. For a given publisher and offer, list the SKUs available using Get-AzVMImageSku.

```PowerShell
$offerName="<offer>"
Get-AzVMImageSku -Location $locName -PublisherName $pubName -Offer
$offerName | Select Skus
```

4. For a SKU, list the versions of the image using Get-AzVMImage.

```PowerShell
$skuName="<SKU>"
Get-AzVMImage -Location $locName -PublisherName $pubName -Offer
$offerName -Sku $skuName | Select Version
```

   You can also use `latest` if you want to use the latest image and not a specific older version.

Now you can combine the selected publisher, offer, SKU, and version into a URN (values separated by :). Pass this URN with the `-Image` parameter when you create a VM with the New-AzVM cmdlet. You can also replace the version number in the URN with `latest` to get the latest version of the image.

If you deploy a VM with a Resource Manager template, then you must set the image parameters individually in the `imageReference` properties. See the template reference.

View purchase plan properties

Some VM images in the Azure Marketplace have other license and purchase terms that you must accept before you can deploy them programmatically. You need to accept the image's terms once per subscription.

To view an image's purchase plan information, run the `Get-AzVMImage` cmdlet. If the `PurchasePlan` property in the output is not `null`, the image has terms you need to accept before programmatic deployment.

For example, the *Windows Server 2016 Datacenter* image doesn't have additional terms, so the `PurchasePlan` information is `null`:

```PowerShell
```

```
$version = "2016.127.20170406"
Get-AzVMImage -Location $locName -PublisherName $pubName -Offer $offerName -
Skus $skuName -Version $version
```

The output looks similar to the following output:

Output

```
Id                : /Subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/Providers/Microsoft.Compute/Locations/westus/Publishers/Microso
ftWindowsServer/ArtifactTypes/VMImage/Offers/WindowsServer/Skus/2016-
Datacenter/Versions/2019.0.20190115
Location          : westus
PublisherName     : MicrosoftWindowsServer
Offer             : WindowsServer
Skus              : 2019-Datacenter
Version           : 2019.0.20190115
FilterExpression  :
Name              : 2019.0.20190115
OSDiskImage       : {
                      "operatingSystem": "Windows"
                    }
PurchasePlan      : null
DataDiskImages    : []
```

The example below shows a similar command for the *Data Science Virtual Machine - Windows 2016* image, which has the following `PurchasePlan` properties: `name`, `product`, and `publisher`. Some images also have a `promotion code` property. To deploy this image, see the following sections to accept the terms and to enable programmatic deployment.

PowerShell

```
Get-AzVMImage -Location "westus" -PublisherName "microsoft-ads" -Offer
"windows-data-science-vm" -Skus "windows2016" -Version "0.2.02"
```

The output looks similar to the following output:

```
Id                : /Subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/Providers/Microsoft.Compute/Locations/westus/Publishers/microso
ft-ads/ArtifactTypes/VMImage/Offers/windows-data-science-
vm/Skus/windows2016/Versions/19.01.14
Location          : westus
PublisherName     : microsoft-ads
Offer             : windows-data-science-vm
```

```
Skus             : windows2016
Version          : 19.01.14
FilterExpression :
Name             : 19.01.14
OSDiskImage      : {
                     "operatingSystem": "Windows"
                   }
PurchasePlan     : {
                     "publisher": "microsoft-ads",
                     "name": "windows2016",
                     "product": "windows-data-science-vm"
                   }
DataDiskImages   : []
```

To view the license terms, use the Get-AzMarketplaceterms cmdlet and pass in the purchase plan parameters. The output provides a link to the terms for the Marketplace image and shows whether you previously accepted the terms. Be sure to use all lowercase letters in the parameter values.

PowerShell

```
Get-AzMarketplaceterms -Publisher "microsoft-ads" -Product "windows-data-
science-vm" -Name "windows2016"
```

The output will look similar to the following:

Output

```
Publisher         : microsoft-ads
Product           : windows-data-science-vm
Plan              : windows2016
LicenseTextLink   :
https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_MI
CROSOFT%253a2DADS%253a24WINDOWS%253a2DDATA%253a2DSCIENCE%253a2DVM%253a24WIND
OWS2016%253a24OC5SKMQOXSED66BBSNTF4XRCS4XLOHP7QMPV54DQU7JCBZWYFP35IDPOWTUKXU
C7ZAG7W6ZMDD6NHWNKUIVSYBZUTZ245F44SU5AD7Q.txt
PrivacyPolicyLink : https://www.microsoft.com/EN-
US/privacystatement/OnlineServices/Default.aspx
Signature         :
2UMWH6PHSAIM4U22HXPXW25AL2NHUJ7Y7GRV27EBL6SUIDURGMYG6IIDO3P47FFIBBDFHZHSQTR7
PNK6VIIRYJRQ3WXSE6BTNUNENXA
Accepted          : False
Signdate          : 1/25/2019 7:43:00 PM
```

Accept purchase plan terms

Use the Set-AzMarketplaceterms cmdlet to accept or reject the terms. You only need to accept terms once per subscription for the image. Be sure to use all lowercase letters in the parameter values.

PowerShell

```
$agreementTerms=Get-AzMarketplaceterms -Publisher "microsoft-ads" -Product
"windows-data-science-vm" -Name "windows2016"

Set-AzMarketplaceTerms -Publisher "microsoft-ads" -Product "windows-data-
science-vm" -Name "windows2016" -Terms $agreementTerms -Accept
```

Output

```
Publisher         : microsoft-ads
Product           : windows-data-science-vm
Plan              : windows2016
LicenseTextLink   :
https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_MI
CROSOFT%253a2DADS%253a24WINDOWS%253a2DDATA%253a2DSCIENCE%253a2DV

M%253a24WINDOWS2016%253a24OC5SKMQOXSED66BBSNTF4XRCS4XLOHP7QMPV54DQU7JCBZWYFP
35IDPOWTUKXUC7ZAG7W6ZMDD6NHWNKUIVSYBZUTZ245F44SU5AD7Q.txt
PrivacyPolicyLink : https://www.microsoft.com/EN-
US/privacystatement/OnlineServices/Default.aspx
Signature         :
XXXXXXK3MNJ5SROEG2BYDA2YGECU33GXTD3UFPLPC4BAVKAUL3PDYL3KBKBLG4ZCDJZVNSA7KJWT
GMDSYDD6KRLV3LV274DLBXXXXXX
Accepted          : True
Signdate          : 2/23/2018 7:49:31 PM
```

Create a new VM from a marketplace image

If you already have the information about what image you want to use, you can pass that information into Set-AzVMSourceImage cmdlet to add image information to the VM configuration. See the next sections for searching and listing the images available in the marketplace.

Some paid images also require that you provide purchase plan information using the Set-AzVMPlan.

PowerShell

```
...

$vmConfig = New-AzVMConfig -VMName "myVM" -VMSize Standard_D1
```

```
# Set the Marketplace image
$offerName = "windows-data-science-vm"
$skuName = "windows2016"
$version = "19.01.14"
$vmConfig = Set-AzVMSourceImage -VM $vmConfig -PublisherName $publisherName
-Offer $offerName -Skus $skuName -Version $version

# Set the Marketplace plan information, if needed
$publisherName = "microsoft-ads"
$productName = "windows-data-science-vm"
$planName = "windows2016"
$vmConfig = Set-AzVMPlan -VM $vmConfig -Publisher $publisherName -Product
$productName -Name $planName

...
```

You'll then pass the VM configuration along with the other configuration objects to the `New-AzVM` cmdlet. For a detailed example of using a VM configuration with PowerShell, see this script.

If you get a message about accepting the terms of the image, see the earlier section Accept purchase plan terms.

Create a new VM from a VHD with purchase plan information

If you have an existing VHD that was created using an Azure Marketplace image, you might need to supply the purchase plan information when you create a new VM from that VHD.

If you still have the original VM, or another VM created from the same image, you can get the plan name, publisher, and product information from it using Get-AzVM. This example gets a VM named *myVM* in the *myResourceGroup* resource group and then displays the purchase plan information.

Azure PowerShell

```
$vm = Get-azvm `
   -ResourceGroupName myResourceGroup `
   -Name myVM
$vm.Plan
```

If you didn't get the plan information before the original VM was deleted, you can file a support request. The support request needs at minimum the VM name, subscription ID and the time stamp of the delete operation.

To create a VM using a VHD, refer to this article Create a VM from a specialized VHD and add in a line to add the plan information to the VM configuration using Set-AzVMPlan similar to the following:

Azure PowerShell

```
$vmConfig = Set-AzVMPlan `
    -VM $vmConfig `
    -Publisher "publisherName" `
    -Product "productName" `
    -Name "planName"
```

Next steps

To create a virtual machine quickly with the `New-AzVM` cmdlet by using basic image information, see Create a Windows virtual machine with PowerShell.

For more information on using Azure Marketplace images to create custom images in an Azure Compute Gallery (formerly known as Shared Image Gallery), see Supply Azure Marketplace purchase plan information when creating images.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use Windows client in Azure for dev/test scenarios

05/22/2025

Applies to: ✔ Windows VMs

You can use Windows 10 Enterprise (x64) or Windows 11 Enterprise (x64) in Azure for dev/test scenarios provided you have an appropriate Visual Studio (formerly MSDN) subscription.

To run Windows 10 or 11 in a production environment see, How to deploy Windows 10 or 11 on Azure with Multitenant Hosting Rights.

Subscription eligibility

Active Visual Studio subscribers (people who have acquired a Visual Studio subscription license) can use Windows client images for development and testing purposes. Windows client images can be used on your own hardware or on Azure virtual machines.

Certain Windows client images are available from the Azure Marketplace. Visual Studio subscribers within any type of offer can also prepare and create 64-bit Windows 10 or 11 images and then upload to Azure.

Eligible offers and client images

The following table details the offer IDs that are eligible to deploy Windows client images through the Azure Marketplace. The Windows client images are only visible to the following offers.

> **Note**
>
> Image offers are under **Windows Client** in the Azure Marketplace. Use **Windows Client** when searching for client images available to Visual Studio subscribers. If you need to purchase a Visual Studio subscription, see the various options at Buy Visual Studio

Expand table

Offer Name	Offer Number	Available client images
Pay-As-You-Go Dev/Test	0023P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)

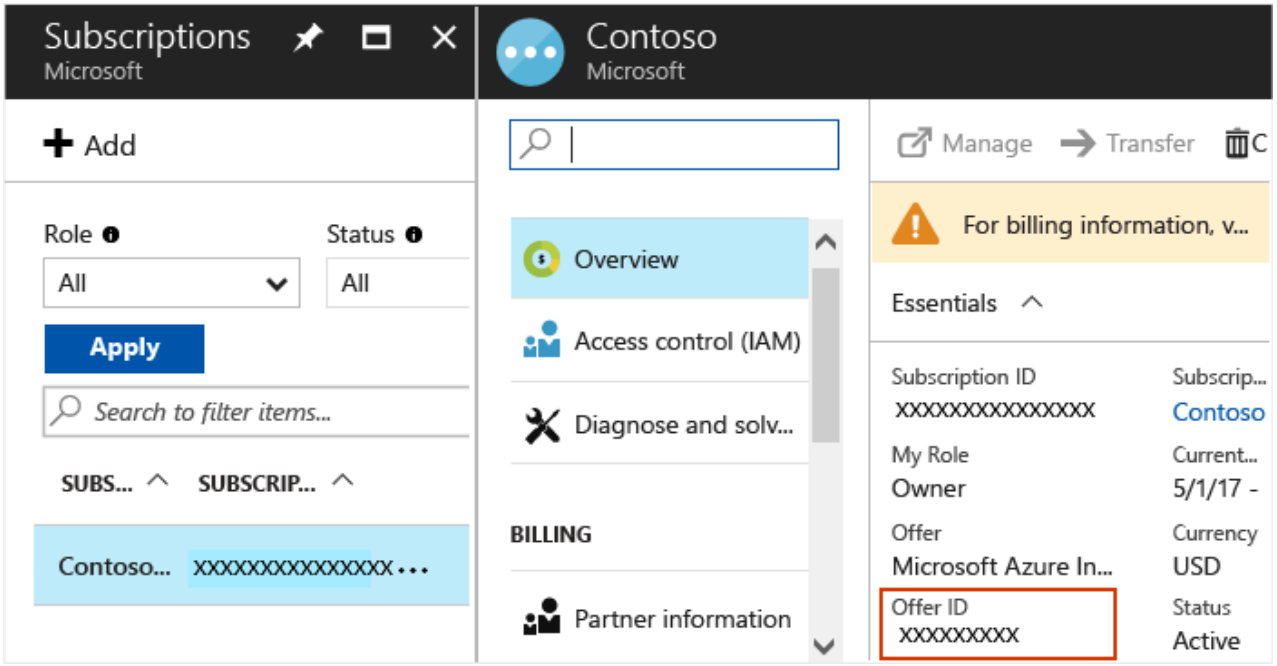
Offer Name	Offer Number	Available client images
Visual Studio Enterprise (MPN) subscribers	0029P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)
Visual Studio Professional subscribers	0059P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)
Visual Studio Test Professional subscribers	0060P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)
Visual Studio Enterprise subscribers	0063P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)
Visual Studio Enterprise (BizSpark) subscribers	0064P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)
Enterprise Dev/Test	0148P	Windows 10 Enterprise N (x64) Windows 11 Enterprise (x64)

For more information, see Understand Microsoft offer types

Check your Azure subscription

If you do not know your offer ID, you can obtain it through the Azure portal.

- On the *Subscriptions* window:



- Or, click **Billing** and then click your subscription ID. The offer ID appears in the *Billing* window.

Next steps

You can now deploy your VMs using PowerShell, Resource Manager templates, or Visual Studio.

How to deploy Windows 11 on Azure

05/22/2025

Applies to: ✔ Windows VMs ✔ Flexible scale sets

For customers with Windows 11 Enterprise E3/E5 per user or Azure Virtual Desktop Access per user (User Subscription Licenses or Add-on User Subscription Licenses), Multitenant Hosting Rights for Windows 11 allows you to bring your Windows 11 Licenses to the cloud and run Windows 11 Virtual Machines on Azure without paying for another license.

For more information, see Multitenant Hosting for Windows 11.

> **Note**
>
> - To use Windows 10 or 11 images for development or testing see Windows client in Azure for dev/test scenarios
> - Student & Free Trial accounts are enabled to deploy Windows 11 images for development or testing purposes.
> - For Windows Server licensing benefits, please refer to Azure Hybrid use benefits for Windows Server images.
> - When selecting a VM size, ensure the size meets the Windows 11 hardware minimum requirements

Subscription Licenses that qualify for Multitenant Hosting Rights

For more details about subscription licenses that qualify to run Windows 11 on Azure, download the Windows 11 licensing brief for Virtual Desktops

> **Important**
>
> Users **must** have one of the below subscription licenses in order to use Windows 11 images in Azure for any production workload. If you do not have one of these subscription licenses, they can be purchased through your Cloud Service Partner or directly through Microsoft.

Operating systems and licenses

You have a choice of operating systems that you can use for session hosts to provide virtual desktops and remote apps. You can use different operating systems with different host pools to provide flexibility to your users. Supported dates are inline with the Microsoft Lifecycle Policy. We support the following 64-bit versions of these operating systems:

Operating system licenses

- Windows 11 Enterprise multi-session
- Windows 11 Enterprise
- Windows 10 Enterprise, version 1909 and later (For Windows 10 deployments)

License entitlement

- Microsoft 365 E3, E5, A3, A5, F3, Business Premium, Student Use Benefit
- Windows Enterprise E3, E5
- Windows VDA E3, E5
- Windows Education A3, A5

External users can use per-user access pricing instead of license entitlement.

Deploying Windows 11 Image from Azure Marketplace

For PowerShell, CLI and Azure Resource Manager template deployments, Windows 11 images can be found using the `PublisherName: MicrosoftWindowsDesktop` and `Offer: Windows-11`.

PowerShell

```
Get-AzVmImageSku -Location 'West US' -PublisherName 'MicrosoftWindowsDesktop' -
Offer 'Windows-11'

Skus                Offer      PublisherName           Location
----                -----      -------------           --------
win11-21h2-avd      Windows-11 MicrosoftWindowsDesktop westus
win11-21h2-ent      Windows-11 MicrosoftWindowsDesktop westus
win11-21h2-entn     Windows-11 MicrosoftWindowsDesktop westus
win11-21h2-pro      Windows-11 MicrosoftWindowsDesktop westus
win11-21h2-pron     Windows-11 MicrosoftWindowsDesktop westus
win11-22h2-avd      Windows-11 MicrosoftWindowsDesktop westus
win11-22h2-ent      Windows-11 MicrosoftWindowsDesktop westus
win11-22h2-entn     Windows-11 MicrosoftWindowsDesktop westus
win11-22h2-pro      Windows-11 MicrosoftWindowsDesktop westus
win11-22h2-pron     Windows-11 MicrosoftWindowsDesktop westus
```

For more information on available images, see Find and use Azure Marketplace VM images with Azure PowerShell

> **Note**
>
> If you are upgrading to a newer version of Windows 11 with Trusted launch enabled and you are currently on a Windows 11 version without Trusted Launch enabled, the VM needs to be deallocated before proceeding with the upgrade. For more information, see Enabling Trusted Launch on existing Azure VMs

Uploading Windows 11 VHD to Azure

If you're uploading a generalized Windows 11 VHD, note Windows 11 doesn't have built-in administrator account enabled by default. To enable the built-in administrator account, include the following command as part of the Custom Script extension.

PowerShell

```
Net user <username> /active:yes
```

The following PowerShell snippet is to mark all administrator accounts as active, including the built-in administrator. This example is useful if the built-in administrator username is unknown.

PowerShell

```
$adminAccount = Get-WmiObject Win32_UserAccount -filter "LocalAccount=True" | ?
{$_.SID -Like "S-1-5-21-*-500"}
if($adminAccount.Disabled)
{
    $adminAccount.Disabled = $false
    $adminAccount.Put()
}
```

For more information:

- How to upload VHD to Azure
- How to prepare a Windows VHD to upload to Azure

Deploying Windows 11 with Multitenant Hosting Rights

Make sure you've installed and configured the latest Azure PowerShell. Once you've prepared your VHD, upload the VHD to your Azure Storage account using the `Add-AzVhd` cmdlet as follows:

PowerShell

```
Add-AzVhd -ResourceGroupName "myResourceGroup" -LocalFilePath
"C:\Path\To\myvhd.vhd" `
    -Destination "https://mystorageaccount.blob.core.windows.net/vhds/myvhd.vhd"
```

Deploy using Azure Resource Manager Template Deployment Within your Resource Manager templates, an additional parameter for `licenseType` can be specified. You can read more about authoring Azure Resource Manager templates. Once you've your VHD uploaded to Azure, edit your Resource Manager template to include the license type as part of the compute provider and deploy your template as normal:

JSON

```
"properties": {
    "licenseType": "Windows_Client",
    "hardwareProfile": {
        "vmSize": "[variables('vmSize')]"
    }
```

Deploy via PowerShell When deploying your Windows Server VM via PowerShell, you need to add another parameter for `-LicenseType`. Once you have your VHD uploaded to Azure, you can create a VM using `New-AzVM` and specify the licensing type as follows:

PowerShell

```
New-AzVM -ResourceGroupName "myResourceGroup" -Location "West US" -VM $vm -
LicenseType "Windows_Client"
```

Verify your VM is utilizing the licensing benefit

Once you've deployed your VM through either the PowerShell or Resource Manager deployment method, verify the license type with `Get-AzVM`:

PowerShell

```
Get-AzVM -ResourceGroup "myResourceGroup" -Name "myVM"
```

The output is similar to the following example for Windows 10 with correct license type:

PowerShell

```
Type                    : Microsoft.Compute/virtualMachines
Location                : westus
LicenseType             : Windows_Client
```

This output contrasts with the following VM deployed without Azure Hybrid Use Benefit licensing, such as a VM deployed straight from the Azure Gallery:

PowerShell

```
Type                    : Microsoft.Compute/virtualMachines
Location                : westus
LicenseType             :
```

Additional Information about joining Microsoft Entra ID

Azure provisions all Windows VMs with built-in administrator account, which can't be used to join Microsoft Entra ID. For example, *Settings* > *Account* > *Access Work or School* > + *Connect* won't work. You must create and log on as a second administrator account to join Microsoft Entra ID manually. You can also configure Microsoft Entra ID using a provisioning package, use the link in the *Next Steps* section to learn more.

Next Steps

- Learn more about Configuring VDA for Windows 11
- Learn more about Multitenant Hosting for Windows 11

Remove machine specific information by deprovisioning or generalizing a VM before creating an image

Summarize this article for me

Generalizing removes machine specific information so the image can be used to create multiple VMs. Once the VM has been generalized or deprovisioned, you need to let the platform know so that the boot sequence can be set correctly.

Generalizing or deprovisioning a VM isn't necessary for creating an image in an Azure Compute Gallery unless you want to create an image that has no machine specific information, like user accounts. Azure Compute Gallery supports *specialized* images that retain machine specific information. For more information, see Generalized and specialized images.

Generalizing is still required when creating a legacy image outside of a gallery.

> **Important**
>
> Once you mark a VM as `generalized` in Azure, you can't restart the VM.

Deprovisioning a Linux VM before creating an image

Use the Azure Linux Agent (waagent) to remove machine-specific information from a Linux VM to prepare it for imaging. Waagent deletes the Azure VM agent's provisioning data, machine-specific files, and the last provisioned user account, making the VM ready to be captured as a reusable image template.

Prerequisites for Linux VMs

Before creating a deprovisioning a VM to create an image, we recommend you follow the distro specific instructions for production workloads. Distribution specific instructions for preparing Linux images for Azure are available here:

- Generic steps
- CentOS
- Debian
- Flatcar

- FreeBSD
- Oracle Linux
- OpenBSD
- Red Hat
- SUSE
- Ubuntu

Deprovision a Linux VM

First you deprovision the VM using the Azure Linux Agent (waagent) to delete machine-specific files and data. Use the `waagent` command with the `-deprovision+user` parameter on your source Linux VM. For more information, see the Azure Linux Agent user guide. This process can't be reversed.

1. Connect to your Linux VM with an SSH client.
2. In the SSH window, enter the following command:

   Bash

   ```
   sudo waagent -deprovision+user
   ```

   > ⓘ **Note**
   >
   > Only run this command on a VM that you'll capture as an image. This command doesn't guarantee that the image is cleared of all sensitive information or is suitable for redistribution. The `+user` parameter also removes the last provisioned user account. To keep user account credentials in the VM, use only `-deprovision`.

3. Type **y** to continue. You can add the `-force` parameter to avoid this confirmation step.
4. After the command completes, type **exit** to close the SSH client. The VM will still be running at this point.

   Deallocate the VM that you deprovisioned with `az vm deallocate` so that it can be generalized.

   Azure CLI

   ```
   az vm deallocate \
      --resource-group myResourceGroup \
      --name myVM
   ```

Then mark the VM as generalized on the platform.

Azure CLI

```
az vm generalize \
   --resource-group myResourceGroup \
   --name myVM
```

Generalizing a Windows VM before creating an image

Sysprep removes all your personal account and security information, and then prepares the machine to be used as an image. For information about Sysprep, see Sysprep overview.

> **Important**
>
> After you have run Sysprep on a VM, that VM is considered *generalized* and can't be restarted. The process of generalizing a VM isn't reversible. If you need to keep the original VM functioning, you should create a snapshot of the OS disk, create a VM from the snapshot, and then generalize that copy of the VM.
>
> Custom answer files aren't supported in the sysprep step, so you can't use the `/unattend:_answerfile_` switch with your sysprep command.

Prerequisites for Windows VMs

- Verify the CD/DVD-ROM is enabled. If it is disabled, the Windows VM will be stuck at out-of-box experience (OOBE). At a command prompt running as an administrator, run the following command:

  Windows Command Prompt

  ```
  REM Enable CD/DVD-ROM
  reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cdrom /v start /t
  REG_DWORD /d 1 /f
  ```

  The Azure platform mounts an ISO file to the DVD-ROM when a Windows VM is created from a generalized image. For this reason, the **DVD-ROM must be enabled in the OS in the generalized image**. If it is disabled, the Windows VM will be stuck at out-of-box experience (OOBE).

- Verify there are no policies applied restricting removable storage access. For example: Computer configuration\Administrative Templates\System\Removable Storage Access\All Removable Storage classes: Deny all access
- Make sure the server roles running on the machine are supported by Sysprep. For more information, see Sysprep support for server roles and Unsupported scenarios.
- Disable encryption. Sysprep requires the drives to be fully decrypted. If encryption is enabled on your VM, disable encryption before you run Sysprep.
- If you plan to run Sysprep on a local machine before uploading your virtual hard disk (VHD) to Azure for the first time, make sure you have prepared your VM before starting.
- Before generalizing a Windows virtual machine, ensure that all installed applications support Sysprep (preinstalled in an OS image). Some applications (Security related, Store applications, Virtualization agents, etc.) either do not support Sysprep or require a specific preparation process to function correctly in a generalized image. Always review and follow the application's official documentation to confirm the required steps before including it in a reusable image.
- For more information about using Sysprep with SQL, see Install SQL Server with SysPrep.
- For information about using creating images with Microsoft Defender for Endpoint (MDE), see MDE for Non-Persistent VDI — Implementation Guide & Best Practices.

Generalize a Windows VM

To generalize your Windows VM, follow these steps:

1. Sign in to your Windows VM.
2. Open a Command Prompt window as an administrator.
3. Delete the panther directory (C:\Windows\Panther) if it exists. This directory contains logs from previous Sysprep operations that can cause Sysprep to fail.

   Windows Command Prompt

   ```
   rd /s /q C:\Windows\Panther
   ```

4. Change to the %windir%\system32\sysprep directory, and then run:

   Windows Command Prompt

```
sysprep.exe /generalize /shutdown
```

5. The VM will shut down when Sysprep is finished generalizing the VM. Do not restart the VM.

   Once Sysprep has finished, set the status of the virtual machine to **Generalized**.

   Azure PowerShell

   ```
   Set-AzVm -ResourceGroupName $rgName -Name $vmName -Generalized
   ```

Related content

- Azure Compute Gallery

Last updated on 03/13/2026

Bringing and creating Linux images in Azure

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

This overview covers the basic concepts around imaging and how to successfully build and use Linux images in Azure. Before you bring a custom image to Azure, you need to be aware of the types and options available to you.

This article talks through the image decision points and requirements as well as explain key concepts so that you can follow this and be able to create your own custom images to your specification.

VM scale set lets you deploy and manage a group of identical (or similar) VMs automatically. Azure handles:

- Scaling up/down (adding/removing VMs)
- Load balancing
- Health monitoring
- Zone/region distribution

There are two main types of scale sets:

- Uniform scale set: Use when you want simplicity and speed. All VMs are identical — same size, image, and configuration.
- Flexible scale set: Use when you want flexibility and high availability. VMs can be different sizes or configurations, and you can mix Spot + Regular VMs.

Difference between managed disks and images

Azure allows you to bring a VHD to the platform to use as a Managed Disk or as a source for an image.

An Azure managed disk is a single VHD. You can either take an existing VHD and create a managed disk from it, or create an empty managed disk from scratch. You can create VMs from managed disks by attaching the disk to the VM, but you can only use a VHD with one VM. You won't be able to modify any OS properties as Azure will just try to turn on the VM and start up using that disk.

Azure images can be made up of multiple OS disks and data disks. When you use a managed image to create a VM, the platform makes a copy of the image and uses that to create the VM. This allows managed images to support reusing the same image for multiple VMs. Azure also

provides advanced management capabilities for images, like global replication and versioning through Azure Compute Gallery (formerly known as Shared Image Gallery).

Generalized and specialized

Azure offers two main image types, generalized and specialized. The terms generalized and specialized are originally Windows terms which migrated in to Azure. These types define how the platform handles the VM when it turns it on. Both types have advantages, disadvantages, and prerequisites. Before you get started, you need to know what image type you need. Below summarizes the scenarios and type you would need to choose:

Expand table

Scenario	Image type	Storage options
Create an image that can be configured for use by multiple VMs. You can set the hostname, add an admin user, and perform other tasks during first boot.	Generalized	Azure Compute Gallery or stand-alone managed images
Create an image from a VM snapshot or a backup.	Specialized	Azure Compute Gallery or a managed disk
Quickly create an image that doesn't need any configuration for creating multiple VMs.	Specialized	Azure Compute Gallery

Generalized images

A generalized image is an image that requires setup to be completed on first boot. For example, on first boot you set the hostname, admin user, and other VM-specific configurations. This is useful when you want the image to be reused multiple times and when you want to pass in parameters during creation. If the generalized image contains the Azure agent, the agent processes the parameters and signal back to the platform that the initial configuration has completed. This process is called provisioning.

Provisioning requires that a provisioner is included in the image. There are two provisioners:

- Azure Linux Agent
- cloud-init

These are prerequisites for creating an image.

Specialized images

These are images that are completely configured and don't require VM or special parameters. The platform will just turn on the VM, and you need to handle uniqueness within the VM, like setting a hostname, to avoid DNS conflicts on the same VNET.

Provisioning agents aren't required for these images, however you may want to have extension handling capabilities. You can install the Linux Agent but disable the provisioning option. Even though you don't need a provisioning agent, the image must fulfill prerequisites for Azure Images.

Image storage options

When bringing your Linux image you have two options:

- Managed images for simple VM creation in a development and test environment.
- Azure Compute Gallery for creating and sharing images at-scale.

Managed images

Managed images can be used to create multiple VMs, but they have many limitations. Managed images can only be created from a generalized source (VM or VHD). They can only be used to create VMs in the same region and they can't be shared across subscriptions and tenants.

Managed images can be used for development and test environments, where you need a couple of simple generalized images to use within single region and subscription.

Azure Compute Gallery

Azure Compute Gallery (formerly known as Shared Image Gallery) is recommended for creating, managing, and sharing images at scale. Azure Compute Gallery helps you build structure and organization around your images.

- Support for both generalized and specialized images.
- Support for image both generation 1 and 2 images.
- Global replication of images.
- Versioning and grouping of images for easier management.
- Highly available images with Zone Redundant Storage (ZRS) in regions that support Availability Zones. ZRS offers better resilience against zonal failures.
- Sharing across subscriptions and even between Active Directory (AD) tenants using Azure RBAC.
- Scaling your deployments with image replicas in each region.

At a high level, you create a gallery and it is made up of:

- Image Definitions - These are containers that hold groups of images.
- Image Versions - These are the actual images.

Hyper-V generation

Azure supports Hyper-V Generation 1 (Gen1) and Generation 2 (Gen2). Gen2 is the latest generation and offers additional functionality over Gen1. For example: increased memory, Intel Software Guard Extensions (Intel SGX), and virtualized persistent memory (vPMEM). Generation 2 VMs running on-premises have some features that aren't supported in Azure yet. For more information, see the Features and capabilities section in this article. Create Gen2 images if you require the additional functionality.

If you still need to create your own image, ensure it meets the image prerequisites and upload to Azure. Distribution specific requirements:

- Debian Linux
- Flatcar Container Linux
- FreeBSD
- Oracle Linux
- OpenBSD
- Red Hat Enterprise Linux
- SLES & openSUSE
- Ubuntu

Next steps

Learn how to create an Azure Compute Gallery.

Last updated on 01/27/2026

Azure Linux VM provisioning

09/02/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When you create a VM from a generalized image (Azure Compute Gallery or Managed Image), the control plane will allow you to create a VM, and pass parameters and settings to the VM. This is called VM *provisioning*. During provisioning, the platform makes required VM Create parameter values (hostname, username, password, SSH keys, customData) available to the VM as it boots.

A provisioning agent baked inside the image will interface with the platform, connecting up to multiple independent provisioning interfaces), set the properties and signal to the platform it has completed.

The provisioning agents can be the Azure Linux Agent, or cloud-init. These are prerequisites of creating generalized images.

The provisioning agents, provide support for all endorsed Azure Linux distributions, and you will find the endorsed distro images in many cases will ship with both cloud-init and the Linux Agent. This gives you the option to have cloud-init to handle the provisioning, then the Linux Agent will provide support to handle Azure Extensions. Providing support for extensions means the VM will then be eligible to support additional Azure services, such as VM Password Reset, Azure Monitoring, Azure Backup, Azure Disk encryption etc.

After provisioning completes, cloud-init will run on each boot. Cloud-init will monitor for changes to the VM, like networking changes, mounting, and formatting the ephemeral disk, and starting the Linux Agent. The Linux Agent continually runs on the server, seeking a 'goal state' (new configuration) from the Azure platform, so whenever you install extensions, the agent will be able to process them.

Whilst there are currently two provisioning agents, cloud-init should be the provisioning agent you choose, and the Linux Agent should be installed for extension support. This allows you to take advantage of platform optimizations, and allows you to disable/remove the Linux Agent, for more details on how to create images without the agent, and how to remove it, please review this documentation.

If you have a Linux kernel that cannot support running either agent, but wish to set some of the VM Create properties, such as hostname, customData, userName, password, ssh keys, then in this document discusses how you can create generalized images without an agent, and meet platform requirements.

Provisioning agent responsibilities

Image provisioning

- Creation of a user account
- Configuring SSH authentication types
- Deployment of SSH public keys and key pairs
- Setting the host name
- Publishing the host name to the platform DNS
- Reporting SSH host key fingerprint to the platform
- Resource Disk Management
- Formatting and mounting the resource disk
- Consuming and processing `customData`

Networking

- Manages routes to improve compatibility with platform DHCP servers
- Ensures the stability of the network interface name

Kernel

- Configures virtual NUMA (disable for kernel < `2.6.37`)
- Consumes Hyper-V entropy for `/dev/random`
- Configures SCSI timeouts for the root device (which could be remote)

Diagnostics

- Console redirection to the serial port

Communication

The information flow from the platform to the agent occurs via two channels:

- A boot-time attached DVD for IaaS deployments. The DVD includes an OVF-compliant configuration file that includes all provisioning information, other than the actual SSH key pairs.
- A TCP endpoint exposing a REST API used to obtain deployment, and topology configuration.

Azure provisioning agent requirements

The Linux Agent, and cloud-init, depend on some system packages in order to function properly:

- Python 2.6+
- OpenSSL 1.0+
- OpenSSH 5.3+
- Filesystem utilities: `sfdisk`, `fdisk`, `mkfs`, `parted`
- Password tools: chpasswd, sudo
- Text processing tools: sed, grep
- Network tools: ip-route
- Kernel support for mounting UDF filesystems.

Next steps

If you need to, you can disable provisioning and remove the Linux agent.

Endorsed Linux distributions on Azure

Article • 02/21/2025

> **⊗ Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

In this article, we'll cover the following:

- Types of Images
- Partners
- Image Update Cadence
- Azure-tuned Kernels

There are several different sources of Linux Virtual Machine (VM) images for Azure. Each source provides a different expectation for quality, utility, and support. This document summarizes each source (marketplace images, platform images, custom images, and community gallery images) and gives you more details about platform images, which are images provided in partnership between Microsoft and several mainstream Linux publishers such as Red Hat, Canonical, and SUSE.

Microsoft's Linux distribution partners provide a multitude of Linux images in the Azure Marketplace. For distributions that aren't available from the Marketplace, you can always provide a custom built Linux image by following the guidelines found in Create and upload a virtual hard disk that contains the Linux operating system. For older versions, see Linux Kernel Requirements.

The Azure Linux Agent is already preinstalled on Azure Marketplace images and is typically available from the distribution package repository. Source code can be found on GitHub.

For more information on support by distribution, see Support for Linux images in Microsoft Azure.

For more information about running Azure workloads on Linux, see the following video: https://www.youtube-nocookie.com/embed/K3ybTXorG5I?si=Bedb40scj_A0oD3U

Types of Images

Azure Linux images can be grouped into three categories:

Marketplace Images

Images published and maintained by either Microsoft or partners. There are a large variety of images from multiple publishers for various use cases (security hardened, full database / application stack, etc.), and can be available free, pay-as-you-go, or BYOL (bring your own license/subscription).

Platform Images are a type of Marketplace images for which Microsoft has partnered with several mainstream publishers (see table below about Partners) to create a set of "platform images" that undergo additional testing and receive predictable updates (see section below on Image Update Cadence). These platform images can be used for building your own custom images and solution stacks. These images are published by the endorsed Linux distribution partners such as Canonical (Ubuntu), Red Hat (RHEL), and Credativ (Debian).

Microsoft provides commercially reasonable customer support for these images. Additionally, Red Hat, Canonical, and SUSE offer integrated vendor support capabilities for their platform images.

Custom Images

These images are created and maintained by the customer, often based on platform images. These images can also be created from scratch and uploaded to Azure - learn how to create custom images. Customers can host these images in Azure Compute Gallery and they can share these images with others in their organization.

Microsoft provides commercially reasonable customer support for custom images.

Community Gallery Images

These images are created and provided by open source projects, communities, and teams. These images are provided using licensing terms set out by the publisher, often under an open source license. They don't appear as traditional marketplace listings, however, they do appear in the portal and via command line tools. More information on community galleries can be found here: Azure Compute Gallery.

Microsoft provides commercially reasonable support for Community Gallery images.

Platform Image Partners

Expand table

Linux Publisher / Distribution	Images (Offers)	Microsoft Support Policy	Description
AlmaLinux	AlmaLinux OS (x86_64/AMD64) AlmaLinux OS (AArch64/ARM64) AlmaLinux OS (x86_64/AMD64) HPC	Microsoft CSS provides commercially reasonable support for these images.	AlmaLinux OS is a 100% community owned and governed, open source and forever free enterprise-grade Linux distribution. With long-term stability and providing a robust production-grade platform, AlmaLinux OS is 1:1 compatible with RHEL/CentOS.
Canonical / Ubuntu	Ubuntu Server 20.04 LTS Ubuntu Pro 20.4 LTS Ubuntu Server 22.04 LTS Ubuntu Pro 22.04 LTS Ubuntu 22.04 LTS (including Pro)	Microsoft CSS provides commercially reasonable support these images.	Canonical produces official Ubuntu images for Microsoft Azure and continuously tracks and delivers updates to these, ensuring security and stability are built from the moment your virtual machines launch. Canonical works closely with Microsoft to optimize Ubuntu images on Azure and ensure Ubuntu supports the latest cloud features as they're released. Ubuntu powers more mission-critical workloads on Azure than any other operating system. https://ubuntu.com/azure
Credativ / Debian	Debian 11 "Bullseye" Debian 12 "Bookworm"	Microsoft CSS provides support for these images.	Credativ is an independent consulting and services company that specializes in the development and implementation of professional solutions by using free software. As leading open-source specialists, Credativ has international recognition with many IT departments that use their support. In conjunction with Microsoft, Credativ is preparing Debian images. The images are specially designed to run on Azure and can be easily managed via the platform. credativ will also support the long-term maintenance and updating of the Debian images for Azure through its Open Source Support Centers.

Linux Publisher / Distribution	Images (Offers)	Microsoft Support Policy	Description
			https://www.credativ.de/portfolio/support/open-source-support-center
Kinvolk / Flatcar	Flatcar Container Linux Flatcar Container Linux (BYOL) Flatcar Container Linux ARM64	Microsoft CSS provides commercially reasonable support these images.	Flatcar Container Linux is a minimal, immutable, and auto-updating operating system for containerized applications. Originally developed by Kinvolk, it's now 100% community governed as a Cloud Native Computing Foundation (CNCF) project. Flatcar is a minimal distro, containing just those packages required for deploying containers. Its immutable file system guarantees consistency and security, while its auto-update capabilities enable you to be always up-to-date with the latest security fixes. Kinvolk was acquired by Microsoft in April 2021 and, post-acquisition, continues its mission within Microsoft to support the Flatcar community. https://www.flatcar.org
Oracle Linux	Oracle Linux	Microsoft CSS provides commercially reasonable support these images.	Oracle's strategy is to offer a broad portfolio of solutions for public and private clouds. The strategy gives customers choice and flexibility in how they deploy Oracle software in Oracle clouds and other clouds. Oracle's partnership with Microsoft enables customers to deploy Oracle software to Microsoft public and private clouds with the confidence of certification and support from Oracle. Oracle's commitment and investment in Oracle public and private cloud solutions is unchanged. https://www.oracle.com/cloud/azure
Red Hat / Red Hat Enterprise Linux (RHEL)	Red Hat Enterprise Linux Red Hat Enterprise Linux RAW Red Hat Enterprise Linux ARM64 Red Hat Enterprise Linux for SAP Apps Red Hat Enterprise	Microsoft CSS provides commercially reasonable support these images.	The world's leading provider of open-source solutions, Red Hat helps more than 90% of Fortune 500 companies solve business challenges, align their IT and business strategies, and prepare for the future of technology. Red Hat achieves this by providing secure solutions through an open business model and an affordable, predictable subscription model. https://www.redhat.com/partners/microsoft

Linux Publisher / Distribution	Images (Offers)	Microsoft Support Policy	Description
	Linux for SAP, HA, Updated Services Red Hat Enterprise Linux with HA add-on		
CIQ / Rocky Linux	Rocky Linux from CIQ	Microsoft CSS provides commercially reasonable support for these images.	Rocky Linux, developed by the Rocky Enterprise Software Foundation (RESF), is an open-source Enterprise Linux operating system designed for the community. As the founding sponsor of RESF, CIQ enhances and supports Rocky Linux on Microsoft Azure, delivering a secure, compliant, high-performance, and user-friendly platform for both development and production workloads. https://ciq.com/partners/cloud/azure/
SUSE / SUSE Linux Enterprise Server (SLES)	SUSE 12 SP5 - +24x7 Support SUSE 12 SP5 - Patching only SUSE 12 SP5 - SAP +24x7 Support SUSE 12 SP5 - BYOS SUSE 12 SP5 - SAP BYOS SUSE 15 SP6 - +24x7 Support SUSE 15 SP6 - Patching only SUSE 15 SP6 - SAP +24x7 Support SUSE 15 SP6 - BYOS	Microsoft CSS provides commercially reasonable support these images.	SUSE Linux Enterprise Server on Azure is a proven platform that provides superior reliability and security for cloud computing. SUSE's versatile Linux platform seamlessly integrates with Azure cloud services to deliver an easily manageable cloud environment. With more than 9,200 certified applications from more than 1,800 independent software vendors for SUSE Linux Enterprise Server, SUSE ensures that workloads running supported in the data center can be confidently deployed on Azure. https://www.suse.com/partners/alliance/microsoft

Linux Publisher / Distribution	Images (Offers)	Microsoft Support Policy	Description
	SUSE 15 SP6 - SAP BYOS SUSE 15 SP6 - arm64 +24x7 Support		

Image Update Cadence

Azure requires that the publishers of the endorsed Linux distributions regularly update their platform images in Azure Marketplace with the latest patches and security fixes, at a quarterly or faster cadence. Updated images in the Marketplace are available automatically to customers as new versions of an image SKU. More information about how to find Linux images: Find Azure Marketplace image information using the Azure CLI.

Azure-tuned Kernels

Azure works closely with various endorsed Linux distributions to optimize the images that they published to Azure Marketplace. One aspect of this collaboration is the development of "tuned" Linux kernels that are optimized for the Azure platform and delivered as fully supported components of the Linux distribution. The Azure-Tuned kernels incorporate new features and performance improvements, and at a faster (typically quarterly) cadence compared to the default or generic kernels that are available from the distribution.

In most cases, you'll find these kernels preinstalled on the default images in Azure Marketplace so customers will immediately get the benefit of these optimized kernels. More information about these Azure-Tuned kernels can be found in the following links:

- CentOS Azure-Tuned Kernel - Available via the CentOS Virtualization SIG
- Debian Cloud Kernel - Available with the Debian 10 and Debian 9 "backports" image on Azure
- SLES Azure-Tuned Kernel
- Ubuntu Azure-Tuned Kernel
- Flatcar Container Linux

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Prepare Linux for imaging in Azure

Article • 10/08/2024

> **⊗ Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the **CentOS End Of Life guidance**.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The Azure platform service-level agreement (SLA) applies to virtual machines (VMs) running the Linux operating system only when you're using one of the endorsed distributions. For endorsed distributions, Azure Marketplace provides preconfigured Linux images. For more information, see:

- Endorsed Linux distributions on Azure
- Support for Linux and open-source technology in Azure

All other distributions running on Azure, including community-supported and non-endorsed distributions, have some prerequisites.

This article focuses on general guidance for running your Linux distribution on Azure. This article can't be comprehensive, because every distribution is different. Even if you meet all the criteria that this article describes, you might need to significantly tweak your Linux system for it to run properly.

General Linux installation notes

- Azure doesn't support the Hyper-V virtual hard disk (VHDX) format. Azure supports only *fixed VHD*. You can convert the disk to VHD format by using Hyper-V Manager or the Convert-VHD cmdlet. If you're using VirtualBox, select **Fixed size** rather than the default (**Dynamically allocated**) when you're creating the disk.
- Azure supports Gen1 (BIOS boot) and Gen2 (UEFI boot) virtual machines.
- The virtual file allocation table (VFAT) kernel module must be enabled in the kernel.
- The maximum size allowed for the VHD is 1,023 GB.
- When you're installing the Linux system, we recommend that you use standard partitions rather than Logical Volume Manager (LVM). LVM is the default for many

installations.

Using standard partitions will avoid LVM name conflicts with cloned VMs, particularly if an OS disk is ever attached to another identical VM for troubleshooting. You can use LVM or RAID on data disks.

- Kernel support for mounting user-defined function (UDF) file systems is necessary. At first boot on Azure, the provisioning configuration is passed to the Linux VM via UDF-formatted media that are attached to the guest. The Azure Linux agent must mount the UDF file system to read its configuration and provision the VM.

- Linux kernel versions earlier than 2.6.37 don't support Non-Uniform Memory Access (NUMA) on Hyper-V with larger VM sizes. This issue primarily affects older distributions that use the upstream Red Hat 2.6.32 kernel. It was fixed in Red Hat Enterprise Linux (RHEL) 6.6 (kernel-2.6.32-504).

  Systems running custom kernels older than 2.6.37, or RHEL-based kernels older than 2.6.32-504, must set the boot parameter `numa=off` on the kernel command line in *grub.conf*. For more information, see Red Hat KB 436883.

- Don't configure a swap partition on the OS disk. You can configure the Linux agent to create a swap file on the temporary resource disk, as described later in this article.

- All VHDs on Azure must have a virtual size aligned to 1 MB (1024 x 1024 bytes). When you're converting from a raw disk to VHD, ensure that the raw disk size is a multiple of 1 MB before conversion, as described later in this article.

- Use the most up-to-date distribution version, packages, and software.

- Remove users and system accounts, public keys, sensitive data, unnecessary software, and applications.

> **Note**
>
> Cloud-init version 21.2 or later removes the UDF requirement. But without the `udf` module enabled, the CD-ROM won't mount during provisioning, which prevents the custom data from being applied. A workaround is to apply user data. However, unlike custom data, user data isn't encrypted. For more information, see User data formats in the cloud-init documentation.

Install kernel modules without Hyper-V

Azure runs on the Hyper-V hypervisor, so Linux requires certain kernel modules to run in Azure. If you have a VM that was created outside Hyper-V, the Linux installers might not include the drivers for Hyper-V in the initial RAM disk (initrd or initramfs), unless the VM detects that it's running in a Hyper-V environment.

When you're using a different virtualization system (such as VirtualBox or KVM) to prepare your Linux image, you might need to rebuild initrd so that at least the `hv_vmbus` and `hv_storvsc` kernel modules are available on the initial RAM disk. This known issue is for systems based on the upstream Red Hat distribution, and possibly others.

The mechanism for rebuilding the initrd or initramfs image can vary, depending on the distribution. Consult your distribution's documentation or support for the proper procedure. Here's one example for rebuilding initrd by using the `mkinitrd` utility:

1. Back up the existing initrd image:

   Bash

   ```
   cd /boot
   sudo cp initrd-`uname -r`.img  initrd-`uname -r`.img.bak
   ```

2. Rebuild initrd by using the `hv_vmbus` and `hv_storvsc` kernel modules:

   Bash

   ```
   sudo mkinitrd --preload=hv_storvsc --preload=hv_vmbus -v -f initrd-
   `uname -r`.img `uname -r`
   ```

Resize VHDs

VHD images on Azure must have a virtual size aligned to 1 MB. Typically, VHDs created through Hyper-V are aligned correctly. If the VHD isn't aligned correctly, you might get an error message similar to the following example when you try to create an image from your VHD:

config

```
The VHD http://<mystorageaccount>.blob.core.windows.net/vhds/MyLinuxVM.vhd
has an unsupported virtual size of 21475270656 bytes. The size must be a
whole number (in MBs).
```

In this case, resize the VM by using either the Hyper-V Manager console or the Resize-VHD PowerShell cmdlet. If you aren't running in a Windows environment, we

recommend using `qemu-img` to convert (if needed) and resize the VHD.

> **Note**
>
> There's a known bug in qemu-img for QEMU version 2.2.1 and some later versions that results in an improperly formatted VHD. The issue was fixed in QEMU 2.6. We recommend using version 2.2.0 or earlier, or using version 2.6 or later.

1. Resizing the VHD directly by using tools such as `qemu-img` or `vbox-manage` might result in an unbootable VHD. We recommend first converting the VHD to a raw disk image by using the following code.

   If the VM image was created as a raw disk image, you can skip this step. Creating the VM image as a raw disk image is the default in some hypervisors, such as KVM.

   Bash

   ```
   sudo qemu-img convert -f vpc -O raw MyLinuxVM.vhd MyLinuxVM.raw
   ```

2. Calculate the required size of the disk image so that the virtual size is aligned to 1 MB. The following Bash shell script uses `qemu-img info` to determine the virtual size of the disk image, and then calculates the size to the next 1 MB:

   Bash

   ```
   rawdisk="MyLinuxVM.raw"
   vhddisk="MyLinuxVM.vhd"

   MB=$((1024*1024))
   size=$(qemu-img info -f raw --output json "$rawdisk" | \
   gawk 'match($0, /"virtual-size": ([0-9]+),/, val) {print val[1]}')

   rounded_size=$((($size+$MB-1)/$MB)*$MB))

   echo "Rounded Size = $rounded_size"
   ```

3. Resize the raw disk by using `$rounded_size`:

   Bash

   ```
   sudo qemu-img resize MyLinuxVM.raw $rounded_size
   ```

4. Convert the raw disk back to a fixed-size VHD:

Bash

```
sudo qemu-img convert -f raw -o subformat=fixed,force_size -O vpc
MyLinuxVM.raw MyLinuxVM.vhd
```

Or, with QEMU versions before 2.6, remove the `force_size` option:

Bash

```
sudo qemu-img convert -f raw -o subformat=fixed -O vpc MyLinuxVM.raw
MyLinuxVM.vhd
```

Linux kernel requirements

The Linux Integration Services (LIS) drivers for Hyper-V and Azure are contributed directly to the upstream Linux kernel. Many distributions that include a recent Linux kernel version (such as 3.x) have these drivers available already, or otherwise provide backported versions of these drivers with their kernels.

LIS drivers are constantly being updated in the upstream kernel with new fixes and features. When possible, we recommend running an endorsed distribution that includes these fixes and updates.

If you're running a variant of RHEL versions 6.0 to 6.3, you need to install the latest LIS drivers for Hyper-V. Beginning with RHEL 6.4+ (and derivatives), the LIS drivers are already included with the kernel, so you don't need additional installation packages.

If a custom kernel is required, we recommend a recent kernel version (such as 3.8+). For distributions or vendors that maintain their own kernel, you need to regularly backport the LIS drivers from the upstream kernel to your custom kernel.

Even if you're already running a relatively recent kernel version, we highly recommend keeping track of any upstream fixes in the LIS drivers and backporting them as needed. The locations of the LIS driver source files are specified in the MAINTAINERS file in the Linux kernel source tree:

config

```
F:    arch/x86/include/asm/mshyperv.h
F:    arch/x86/include/uapi/asm/hyperv.h
F:    arch/x86/kernel/cpu/mshyperv.c
F:    drivers/hid/hid-hyperv.c
F:    drivers/hv/
F:    drivers/input/serio/hyperv-keyboard.c
F:    drivers/net/hyperv/
```

```
F:    drivers/scsi/storvsc_drv.c
F:    drivers/video/fbdev/hyperv_fb.c
F:    include/linux/hyperv.h
F:    tools/hv/
```

The VM's active kernel must include the following patches. This list can't be complete for all distributions.

- ata_piix: defer disks to the Hyper-V drivers by default
- storvsc: Account for in-transit packets in the RESET path
- storvsc: avoid usage of WRITE_SAME
- storvsc: Disable WRITE SAME for RAID and virtual host adapter drivers
- storvsc: NULL pointer dereference fix
- storvsc: ring buffer failures may result in I/O freeze
- scsi_sysfs: protect against double execution of __scsi_remove_device

Azure Linux Agent

The Azure Linux Agent (`waagent`) provisions a Linux virtual machine in Azure. You can get the latest version, report problems, or submit pull requests at the Linux Agent GitHub repo.

Here are some considerations for using the Azure Linux Agent:

- The Linux agent is released under the Apache 2.0 license. Many distributions already provide .rpm or .deb packages for the agent. You can easily install and update these packages.
- The Azure Linux Agent requires Python v2.6+.
- The agent also requires the `python-pyasn1` module. Most distributions provide this module as a separate package to be installed.
- In some cases, the Azure Linux Agent might not be compatible with NetworkManager. Many of the packages (.rpm or .deb) provided by distributions configure NetworkManager as a conflict to the `waagent` package. In these cases, the agent will uninstall NetworkManager when you install the Linux agent package.
- The Azure Linux Agent must be at or above the minimum supported version.

> **Note**
>
> Make sure the `udf` and `vfat` modules are enabled. Disabling the `udf` module will cause a provisioning failure. Disabling the `vfat` module will cause both provisioning and boot failures. Cloud-init version 21.2 or later can provision VMs without requiring UDF if both of these conditions exist:

- You created the VM by using SSH public keys and not passwords.
- You didn't provide any custom data.

General Linux system requirements

1. Modify the kernel boot line in GRUB or GRUB2 to include the following parameters, so that all console messages are sent to the first serial port. These messages can assist Azure support with debugging any issues.

   ```config
   GRUB_CMDLINE_LINUX="rootdelay=300 console=ttyS0 earlyprintk=ttyS0
   net.ifnames=0"
   ```

   We also recommend *removing* the following parameters if they exist:

   ```config
   rhgb quiet crashkernel=auto
   ```

   Graphical and quiet boot aren't useful in a cloud environment, where you want all logs sent to the serial port. You can leave the `crashkernel` option configured if needed, but this parameter reduces the amount of available memory in the VM by at least 128 MB. Reducing available memory might be problematic for smaller VM sizes.

2. After you finish editing */etc/default/grub*, run the following command to rebuild the GRUB configuration:

   ```Bash
   sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

3. Add the Hyper-V module for initramfs by using `dracut`:

   ```Bash
   cd /boot
   sudo cp initramfs-<kernel-version>.img <kernel-version>.img.bak
   sudo dracut -f -v initramfs-<kernel-version>.img <kernel-version> --
   add-drivers "hv_vmbus hv_netvsc hv_storvsc"
   ```

```
sudo grub-mkconfig -o /boot/grub/grub.cfg
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Add the Hyper-V module for initrd by using `mkinitramfs`:

```
Bash

cd /boot
sudo cp initrd.img-<kernel-version>  initrd.img-<kernel-version>.bak
sudo mkinitramfs -o initrd.img-<kernel-version> <kernel-version>  --
with=hv_vmbus,hv_netvsc,hv_storvsc
sudo update-grub
```

4. Ensure that the SSH server is installed and configured to start at boot time. This configuration is usually the default.

5. Install the Azure Linux Agent.

   The Azure Linux Agent is required for provisioning a Linux image on Azure. Many distributions provide the agent as an .rpm or .deb package. The package is typically called `WALinuxAgent` or `walinuxagent`. You can also install the agent manually by following the steps in the Azure Linux Agent guide.

   > **Note**
   >
   > Make sure the `udf` and `vfat` modules are enabled. Removing or disabling them will cause a provisioning or boot failure. Cloud-init version 21.2 or later removes the UDF requirement.

   Install the Azure Linux Agent, cloud-init, and other necessary utilities by running one of the following commands.

   Use this command for Red Hat or CentOS:

   ```
   Bash

   sudo yum install -y WALinuxAgent cloud-init cloud-utils-growpart gdisk
   hyperv-daemons
   ```

   Use this command for Ubuntu/Debian:

   ```
   Bash
   ```

```
sudo apt install walinuxagent cloud-init cloud-utils-growpart gdisk
hyperv-daemons
```

Use this command for SUSE:

```Bash
sudo zypper install python-azure-agent cloud-init cloud-utils-growpart
gdisk hyperv-daemons
```

Then enable the agent and cloud-init on all distributions:

```Bash
sudo systemctl enable waagent.service
sudo systemctl enable cloud-init.service
```

6. Don't create swap space on the OS disk.

   You can use the Azure Linux Agent or cloud-init to configure swap space via the local resource disk. This resource disk is attached to the VM after provisioning on Azure. The local resource disk is a temporary disk and might be emptied when the VM is deprovisioned. The following blocks show how to configure this swap.

   If you choose Azure Linux Agent, modify the following parameters in */etc/waagent.conf*:

```config
ResourceDisk.Format=y
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt/resource
ResourceDisk.EnableSwap=y
ResourceDisk.SwapSizeMB=2048    ## NOTE: Set this to your desired size.
```

   If you choose cloud-init, configure cloud-init to handle the provisioning:

```Bash
sudo sed -i 's/Provisioning.Agent=auto/Provisioning.Agent=cloud-init/g'
/etc/waagent.conf
sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g'
/etc/waagent.conf
sudo sed -i 's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
/etc/waagent.conf
```

To configure cloud-init to format and create swap space, you have two options:

- Pass in a cloud-init configuration every time you create a VM through `customdata`. We recommend this method.
- Use a cloud-init directive in the image to configure swap space every time the VM is created.

Create a .cfg file to configure swap space by using cloud-init:

Bash

```
echo 'DefaultEnvironment="CLOUD_CFG=/etc/cloud/cloud.cfg.d/00-azure-swap.cfg"' | sudo tee -a /etc/systemd/system.conf
cat << EOF | sudo tee /etc/cloud/cloud.cfg.d/00-azure-swap.cfg
#cloud-config
# Generated by Azure cloud image build
disk_setup:
  ephemeral0:
    table_type: mbr
    layout: [66, [33, 82]]
    overwrite: True
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
  - ["ephemeral0.1", "/mnt/resource"]
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-systemd.requires=cloud-init.service,x-systemd.device-timeout=2", "0", "0"]
EOF
```

7. Configure cloud-init to handle the provisioning:

   a. Configure `waagent` for cloud-init:

   Bash

   ```
   sudo sed -i 's/Provisioning.Agent=auto/Provisioning.Agent=cloud-init/g' /etc/waagent.conf
   sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g' /etc/waagent.conf
   sudo sed -i 's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g' /etc/waagent.conf
   ```

   If you're migrating a specific virtual machine and don't want to create a generalized image, set `Provisioning.Agent=disabled` in the */etc/waagent.conf*

configuration.

b. Configure mounts:

```
Bash

echo "Adding mounts and disk_setup to init stage"
sudo sed -i '/ - mounts/d' /etc/cloud/cloud.cfg
sudo sed -i '/ - disk_setup/d' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - mounts' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - disk_setup'
/etc/cloud/cloud.cfg
```

c. Configure the Azure data source:

```
Bash

echo "Allow only Azure datasource, disable fetching network setting
via IMDS"
cat << EOF | sudo tee /etc/cloud/cloud.cfg.d/91-azure_datasource.cfg
datasource_list: [ Azure ]
datasource:
   Azure:
     apply_network_config: False
EOF
```

d. Remove the existing swap file if you configured one:

```
Bash

if [[ -f /mnt/resource/swapfile ]]; then
echo "Removing swapfile" #RHEL uses a swap file by default
swapoff /mnt/resource/swapfile
rm /mnt/resource/swapfile -f
fi
```

e. Configure cloud-init logging:

```
Bash

echo "Add console log file"
cat << EOF | sudo tee -a /etc/cloud/cloud.cfg.d/05_logging.cfg

# This tells cloud-init to redirect its stdout and stderr to
# 'tee -a /var/log/cloud-init-output.log' so the user can see output
# there without needing to look on the console.
```

```
output: {all: '| tee -a /var/log/cloud-init-output.log'}
EOF
```

8. Run the following commands to deprovision the virtual machine.

   > **⊗ Caution**
   >
   > If you're migrating a specific virtual machine and don't want to create a generalized image, skip the deprovisioning step. Running the command `waagent -force -deprovision+user` will render the source machine unusable. This step is intended only to create a generalized image.

   Bash

   ```
   sudo rm -f /var/log/waagent.log
   sudo cloud-init clean
   sudo waagent -force -deprovision+user
   sudo rm -f ~/.bash_history
   sudo export HISTSIZE=0
   ```

   On VirtualBox, you might see an error message after you run `waagent -force -deprovision` that says `[Errno 5] Input/output error`. This error message is not critical, and you can ignore it.

9. Shut down the virtual machine and upload the VHD to Azure.

Next steps

Create a Linux VM from a custom disk by using the Azure CLI

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Prepare a CentOS-based virtual machine for Azure

Article • 08/22/2024

> **⊗ Caution**
>
> This article references CentOS, a Linux distribution that's nearing end-of-life (EOL) status. Consider your use and plan accordingly. For more information, see the CentOS end-of-life guidance.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Learn to create and upload an Azure virtual hard disk (VHD) that contains a CentOS-based Linux operating system (OS). For more information, see:

- Prepare a CentOS 6.x virtual machine (VM) for Azure
- Prepare a CentOS 7.0+ VM for Azure

Prerequisites

This article assumes that you've already installed a CentOS (or similar derivative) Linux OS to a VHD. Multiple tools exist to create .vhd files. An example is a virtualization solution such as Hyper-V. For instructions, see Install the Hyper-V role and configure a VM.

CentOS installation notes

- For more tips on preparing Linux for Azure, see General Linux installation notes.
- The VHDX format isn't supported in Azure, only *fixed VHD*. You can convert the disk to VHD format by using Hyper-V Manager or the `convert-vhd` cmdlet. If you're using VirtualBox, you select **Fixed size** as opposed to the default that's dynamically allocated when you create the disk.
- The vfat kernel module must be enabled in the kernel.
- When you install the Linux system, we recommend that you use standard partitions rather than Logical Volume Manager (LVM), which is often the default for many installations. Using partitions avoids LVM name conflicts with cloned VMs, particularly if an OS disk ever needs to be attached to another identical VM for troubleshooting. LVM or RAID can also be used on data disks.

- Kernel support for mounting user-defined functions (UDF) file systems is necessary. At first boot on Azure, the provisioning configuration is passed to the Linux VM by using UDF-formatted media that's attached to the guest. The Azure Linux agent or `cloud-init` must mount the UDF file system to read its configuration and provision the VM.
- Linux kernel versions below 2.6.37 don't support NUMA on Hyper-V with larger VM sizes. This issue primarily affects older distributions using the upstream Centos 2.6.32 kernel and was fixed in Centos 6.6 (kernel-2.6.32-504). Systems running custom kernels older than 2.6.37 or Red Hat Enterprise Linux (RHEL)-based kernels older than 2.6.32-504 must set the boot parameter `numa=off` on the kernel command line in *grub.conf*. For more information, see Red Hat KB 436883.
- Don't configure a swap partition on the OS disk.
- All VHDs on Azure must have a virtual size aligned to 1 MB. When you convert from a raw disk to VHD, you must ensure that the raw disk size is a multiple of 1 MB before conversion. For more information, see Linux installation notes.

> **Note**
>
> *Cloud-init >= 21.2 removes the UDF requirement*. But without the UDF module enabled, the CD-ROM won't mount during provisioning, which prevents custom data from being applied. A workaround for this situation is to apply custom data by using user data. Unlike custom data, user data isn't encrypted. For more information, see User data formats.

CentOS 6.x

> **Important**
>
> CentOS 6 has reached its EOL and is no longer supported by the CentOS community. No further updates or security patches will be released for this version, leaving it vulnerable to potential security risks. We strongly recommend that you upgrade to a more recent version of CentOS to ensure the safety and stability of your system. For further assistance, consult with your IT department or system administrator.

1. In Hyper-V Manager, select the VM.
2. Select **Connect** to open a console window for the VM.

3. In CentOS 6, `NetworkManager` can interfere with the Azure Linux agent. Uninstall this package:

```
Bash

sudo rpm -e --nodeps NetworkManager
```

4. Create or edit the file `/etc/sysconfig/network` and add the following text:

```
config

NETWORKING=yes
HOSTNAME=localhost.localdomain
```

5. Create or edit the file `/etc/sysconfig/network-scripts/ifcfg-eth0` and add the following text:

```
config

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
```

6. Modify udev rules to avoid generating static rules for the Ethernet interfaces. These rules can cause problems when you clone a VM in Azure or Hyper-V:

```
Bash

sudo ln -s /dev/null /etc/udev/rules.d/75-persistent-net-
generator.rules
sudo rm -f /etc/udev/rules.d/70-persistent-net.rules
```

7. Ensure that the network service starts at boot time:

```
Bash

sudo chkconfig network on
```

8. If you want to use the OpenLogic mirrors that are hosted within the Azure datacenters, replace the `/etc/yum.repos.d/CentOS-Base.repo` file with the following

repositories. This action also adds the [openlogic] repository that includes extra packages, such as the Azure Linux agent:

config

```
[openlogic]
name=CentOS-$releasever - openlogic packages for $basearch
baseurl=http://olcentgbl.trafficmanager.net/openlogic/$releasever/openl
ogic/$basearch/
enabled=1
gpgcheck=0

[base]
name=CentOS-$releasever - Base
#mirrorlist=http://mirrorlist.centos.org/?
release=$releasever&arch=$basearch&repo=os&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/os/$base
arch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#released updates
[updates]
name=CentOS-$releasever - Updates
#mirrorlist=http://mirrorlist.centos.org/?
release=$releasever&arch=$basearch&repo=updates&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/updates/
$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#additional packages that might be useful
[extras]
name=CentOS-$releasever - Extras
#mirrorlist=http://mirrorlist.centos.org/?
release=$releasever&arch=$basearch&repo=extras&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/extras/$
basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
#mirrorlist=http://mirrorlist.centos.org/?
release=$releasever&arch=$basearch&repo=centosplus&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/centospl
us/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#contrib - packages by Centos Users
[contrib]
```

```
name=CentOS-$releasever - Contrib
#mirrorlist=http://mirrorlist.centos.org/?
release=$releasever&arch=$basearch&repo=contrib&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/contrib/
$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

> **Note**
>
> The rest of this article assumes that you're using at least the `[openlogic]` repo, which is used to install the Azure Linux agent.

9. Add the following line to `/etc/yum.conf`:

```config
http_caching=packages
```

10. Clear the current yum metadata and update the system with the latest packages:

```Bash
sudo yum clean all
```

Unless you're creating an image for an older version of CentOS, we recommend that you update all the packages to the latest:

```Bash
sudo yum -y update
```

A reboot might be required after you run this command.

11. Optional: Install the drivers for the Linux Integration Services (LIS).

> **Important**
>
> The step is *required* for CentOS 6.3 and earlier and is optional for later releases.

```Bash
```

```
sudo rpm -e hypervkvpd  ## (might return an error if not installed,
that's OK)
sudo yum install microsoft-hyper-v
```

Alternatively, you can follow the manual installation instructions on the LIS download page to install the RPM onto your VM.

12. Install the Azure Linux agent and dependencies. Start and enable the `waagent` service:

    ```
    Bash

    sudo yum install python-pyasn1 WALinuxAgent
    sudo service waagent start
    sudo chkconfig waagent on
    ```

    The WALinuxAgent package removes the `NetworkManager` and `NetworkManager-gnome` packages if they weren't already removed, as described in step 3.

13. Modify the kernel boot line in your grub configuration to include other kernel parameters for Azure. To do this step, open `/boot/grub/menu.lst` in a text editor and ensure that the default kernel includes the following parameters:

    ```
    config

    console=ttyS0 earlyprintk=ttyS0 rootdelay=300
    ```

    This modification also ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues.

    We also recommend that you *remove* the following parameters:

    ```
    config

    rhgb quiet crashkernel=auto
    ```

    Graphical and quiet boot aren't useful in a cloud environment where you want all the logs to be sent to the serial port. The `crashkernel` option can be left configured if you want. But this parameter reduces the amount of available memory in the VM by 128 MB or more, which might be a problem for smaller VM sizes.

> **Important**
>
> CentOS 6.5 and earlier must also set the kernel parameter `numa=off`. For more information, see Red Hat KB 436883.

14. Ensure that the Secure Shell server is installed and configured to start at boot time. This setting is usually the default.

15. Don't create swap space on the OS disk.

    The Azure Linux agent can automatically configure swap space by using the local resource disk that's attached to the VM after the VM is provisioned on Azure. The local resource disk is a *temporary* disk and might be emptied when the VM is deprovisioned. After you install the Azure Linux agent (see the previous step), modify the following parameters in `/etc/waagent.conf` appropriately:

    ```config
    ResourceDisk.Format=y
    ResourceDisk.Filesystem=ext4
    ResourceDisk.MountPoint=/mnt/resource
    ResourceDisk.EnableSwap=y
    ResourceDisk.SwapSizeMB=2048 ## NOTE: set this to whatever you need it
    to be.
    ```

16. Deprovision the VM and prepare it for provisioning on Azure:

    ```Bash
    sudo waagent -force -deprovision+user
    sudo export HISTSIZE=0
    ```

    > **Note**
    >
    > If you're migrating a specific VM and don't want to create a generalized image, skip the deprovision step.

17. Select **Action** > **Shut Down** in Hyper-V Manager. Your Linux VHD is now ready to be uploaded to Azure.

CentOS 7.0+

Follow the steps in the next sections if you're using CentOS 7.0+.

Changes in CentOS 7 (and similar derivatives)

Preparing a CentOS 7 VM for Azure is similar to CentOS 6. Several significant differences are worth noting:

- The `NetworkManager` package no longer conflicts with the Azure Linux agent. This package is installed by default and we recommend that you don't remove it.
- GRUB2 is now used as the default bootloader, so the procedure for editing kernel parameters has changed. (See the "Configuration steps" section.)
- XFS is now the default file system. The ext4 file system can still be used if you want.
- Because CentOS 8 Stream and newer no longer include `network.service` by default, you need to install it manually:

  Bash

  ```
  sudo yum install network-scripts
  sudo systemctl enable network.service
  ```

Configuration steps

1. In Hyper-V Manager, select the VM.
2. Select **Connect** to open a console window for the VM.
3. Create or edit the file `/etc/sysconfig/network` and add the following text:

   config

   ```
   NETWORKING=yes
   HOSTNAME=localhost.localdomain
   ```

4. Create or edit the file `/etc/sysconfig/network-scripts/ifcfg-eth0` and add the following text:

   config

   ```
   DEVICE=eth0
   ONBOOT=yes
   BOOTPROTO=dhcp
   TYPE=Ethernet
   ```

```
USERCTL=no
PEERDNS=yes
IPV6INIT=no
NM_CONTROLLED=no
```

5. Modify udev rules to avoid generating static rules for the Ethernet interfaces. These rules can cause problems when you clone a VM in Azure or Hyper-V:

```Bash
sudo ln -s /dev/null /etc/udev/rules.d/75-persistent-net-generator.rules
```

6. If you want to use the `OpenLogic` mirrors that are hosted within the Azure datacenters, replace the */etc/yum.repos.d/CentOS-Base.repo* file with the following repositories. This action also adds the [openlogic] repository that includes packages for the Azure Linux agent:

```confg
[openlogic]
name=CentOS-$releasever - openlogic packages for $basearch
baseurl=http://olcentgbl.trafficmanager.net/openlogic/$releasever/openlogic/$basearch/
enabled=1
gpgcheck=0

[base]
name=CentOS-$releasever - Base
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS-$releasever - Updates
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that might be useful
[extras]
name=CentOS-$releasever - Extras
#mirrorlist=http://mirrorlist.centos.org/?
```

```
release=$releasever&arch=$basearch&repo=extras&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/extras/$
basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
#mirrorlist=http://mirrorlist.centos.org/?
release=$releasever&arch=$basearch&repo=centosplus&infra=$infra
baseurl=http://olcentgbl.trafficmanager.net/centos/$releasever/centospl
us/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

> **Note**
>
> The rest of this article assumes that you're using at least the `[openlogic]` repo, which is used to install the Azure Linux agent.

7. Clear the current yum metadata and install any updates:

   Bash

   ```
   sudo yum clean all
   ```

   Unless you're creating an image for an older version of CentOS, we recommend that you update all the packages to the latest:

   Bash

   ```
   sudo yum -y update
   ```

   A reboot might be required after you run this command.

8. Modify the kernel boot line in your grub configuration to include other kernel parameters for Azure. To do this step, open `/etc/default/grub` in a text editor and edit the `GRUB_CMDLINE_LINUX` parameter. For example:

   config

   ```
   GRUB_CMDLINE_LINUX="rootdelay=300 console=ttyS0 earlyprintk=ttyS0
   net.ifnames=0"
   ```

This modification also ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues. It also turns off the new CentOS 7 naming conventions for network interface cards. We also recommend that you *remove* the following parameters:

```config
rhgb quiet crashkernel=auto
```

Graphical and quiet boot aren't useful in a cloud environment where you want all the logs to be sent to the serial port. The `crashkernel` option can be left configured if you want. But this parameter reduces the amount of available memory in the VM by 128 MB or more, which might be a problem for smaller VM sizes.

9. After you're finished editing `/etc/default/grub`, rebuild the grub configuration:

```Bash
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

> **Note**
>
> If you're uploading a UEFI-enabled VM, the command to update grub is `grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg`. Also, the vfat kernel module must be enabled in the kernel. Otherwise, provisioning fails.
>
> Make sure the udf module is enabled. Removing or disabling it will cause a provisioning or boot failure. *(_Cloud-init >= 21.2 removes the udf requirement. For more information, read the top of the document.)*

10. If you're building the image from VMware, VirtualBox, or KVM, ensure that the Hyper-V drivers are included in the initramfs:

    a. Edit `/etc/dracut.conf` and add content:

```config
add_drivers+=" hv_vmbus hv_netvsc hv_storvsc "
```

    b. Rebuild the initramfs:

Bash

```
sudo dracut -f -v
```

11. Install the Azure Linux agent and dependencies for Azure VM extensions:

Bash

```
sudo yum install python-pyasn1 WALinuxAgent
sudo systemctl enable waagent
```

12. Install `cloud-init` to handle the provisioning:

Bash

```
sudo yum install -y cloud-init cloud-utils-growpart gdisk hyperv-
daemons
```

- Configure `waagent` for `cloud-init`:

Bash

```
sudo sed -i 's/Provisioning.Agent=auto/Provisioning.Agent=auto/g'
/etc/waagent.conf
sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g'
/etc/waagent.conf
sudo sed -i 's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
/etc/waagent.conf
```

Bash

```
sudo echo "Adding mounts and disk_setup to init stage"
sudo sed -i '/ - mounts/d' /etc/cloud/cloud.cfg
sudo sed -i '/ - disk_setup/d' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - mounts' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - disk_setup' /etc/cloud/cloud.cfg
```

Bash

```
sudo echo "Allow only Azure datasource, disable fetching network
setting via IMDS"
sudo cat > /etc/cloud/cloud.cfg.d/91-azure_datasource.cfg <<EOF
datasource_list: [ Azure ]
datasource:
    Azure:
        apply_network_config: False
EOF
```

```
if [[ -f /mnt/swapfile ]]; then
echo Removing swapfile - RHEL uses a swapfile by default
swapoff /mnt/swapfile
rm /mnt/swapfile -f
fi

echo "Add console log file"
cat >> /etc/cloud/cloud.cfg.d/05_logging.cfg <<EOF

# This tells cloud-init to redirect its stdout and stderr to
# 'tee -a /var/log/cloud-init-output.log' so the user can see output
# there without needing to look on the console.
output: {all: '| tee -a /var/log/cloud-init-output.log'}
EOF
```

13. Swap configuration:

    a. Don't create swap space on the OS disk.

    Previously, the Azure Linux agent was used to automatically configure swap space by using the local resource disk that's attached to the VM after the VM is provisioned on Azure. However, `cloud-init` now handles this step. You *must not* use the Linux agent to format the resource disk to create the swap file. Modify the following parameters in `/etc/waagent.conf` appropriately:

    Bash

    ```
    sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g'
    /etc/waagent.conf
    sudo sed -i
    's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
    /etc/waagent.conf
    ```

    b. If you want to mount, format, and create the swap file, you can either:

    - Pass this command in as a `cloud-init` configuration every time you create a VM.
    - Use a `cloud-init` directive baked into the image to do this step every time the VM is created:

      Bash

      ```
      sudo echo
      'DefaultEnvironment="CLOUD_CFG=/etc/cloud/cloud.cfg.d/00-azure-
      swap.cfg"' >> /etc/systemd/system.conf
      sudo cat > /etc/cloud/cloud.cfg.d/00-azure-swap.cfg << EOF
      ```

```
#cloud-config
# Generated by Azure cloud image build
disk_setup:
  ephemeral0:
    table_type: mbr
    layout: [66, [33, 82]]
    overwrite: True
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
  - ["ephemeral0.1", "/mnt"]
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-
systemd.requires=cloud-init.service,x-systemd.device-
timeout=2", "0", "0"]
EOF
```

14. Run the following commands to deprovision the VM and prepare it for provisioning on Azure.

    > **Note**
    >
    > If you're migrating a specific VM and don't want to create a generalized image, skip the deprovision step.

    Bash

    ```
    sudo rm -f /var/log/waagent.log
    sudo cloud-init clean
    sudo waagent -force -deprovision+user
    sudo rm -f ~/.bash_history
    sudo export HISTSIZE=0
    ```

15. Select **Action** > **Shut Down** in Hyper-V Manager. Your Linux VHD is now ready to be uploaded to Azure.

Related content

You're now ready to use your CentOS Linux VHD to create new VMs in Azure. If this is the first time that you're uploading the .vhd file to Azure, see Create a Linux VM from a custom disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Prepare a Debian VHD for Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Prerequisites

This section assumes that you've already installed a Debian Linux operating system from an .iso file downloaded from the Debian website to a virtual hard disk (VHD). Multiple tools exist to create .vhd files. Hyper-V is only one example. For instructions on using Hyper-V, see Install the Hyper-V role and configure a virtual machine (VM).

Installation notes

- For more tips on preparing Linux for Azure, see General Linux installation notes.
- The newer VHDX format isn't supported in Azure. You can convert the disk to VHD format by using Hyper-V Manager or the `convert-vhd` cmdlet.
- When you install the Linux system, we recommend that you use standard partitions rather than Logical Volume Manager (LVM), which is often the default for many installations. Using partitions avoids LVM name conflicts with cloned VMs, particularly if an OS disk ever needs to be attached to another VM for troubleshooting. LVM or RAID can also be used on data disks.
- Don't configure a swap partition on the OS disk. The Azure Linux agent can be configured to create a swap file on the temporary resource disk. More information is available in the following steps.
- All VHDs on Azure must have a virtual size aligned to 1 MB. When you convert from a raw disk to VHD, you must ensure that the raw disk size is a multiple of 1 MB before conversion. For more information, see Linux installation notes.

Prepare a Debian image for Azure

You can create the base Azure Debian cloud image with the fully automatic installation (FAI) cloud image builder. To prepare an image without FAI, check out the generic steps article.

The following git clone and apt installation commands were pulled from the Debian cloud images repo. Start by cloning the repo and installing dependencies:

```
```

```
$ git clone https://salsa.debian.org/cloud-team/debian-cloud-images.git
$ sudo apt install --no-install-recommends ca-certificates debsums
dosfstools \
    fai-server fai-setup-storage make python3 python3-libcloud python3-
marshmallow \
    python3-pytest python3-yaml qemu-utils udev
$ cd ./debian-cloud-images
```

Optional: Customize the build by adding scripts (for example, shell scripts) to `./config_space/scripts/AZURE`.

Script example to customize the image

```
$ mkdir -p ./config_space/scripts/AZURE
$ cat > ./config_space/scripts/AZURE/10-custom <<EOF
#!/bin/bash

\$ROOTCMD bash -c "echo test > /usr/local/share/testing"
EOF
$ sudo chmod 755 ./config_space/scripts/AZURE/10-custom
```

Prefix any commands you want to have customizing the image with `$ROOTCMD`. It's aliased as `chroot $target`.

Build the Azure Debian image

```
$ make image_[release]_azure_amd64
```

This command outputs a handful of files in the current directory, most notably the `image_[release]_azure_amd64.raw` image file.

Convert the raw image to VHD for Azure:

```
rawdisk="image_[release]_azure_amd64.raw"
vhddisk="image_[release]_azure_amd64.vhd"

MB=$((1024*1024))
size=$(qemu-img info -f raw --output json "$rawdisk" | \
```

```
gawk 'match($0, /"virtual-size": ([0-9]+),/, val) {print val[1]}')

rounded_size=$((($size+$MB-1)/$MB)*$MB))
rounded_size_adjusted=$(($rounded_size + 512))

echo "Rounded Size Adjusted = $rounded_size_adjusted"

sudo qemu-img resize "$rawdisk" $rounded_size
qemu-img convert -f raw -o subformat=fixed,force_size -O vpc "$rawdisk"
"$vhddisk"
```

This process creates a VHD `image_[release]_azure_amd64.vhd` with a rounded size so that it can be copied successfully to an Azure disk.

> ⓘ **Note**
>
> Rather than cloning the salsa repository and building images locally, current stable images can be built and downloaded from **FAI**.

After you create a stable Debian VHD image and before you upload, verify that the following packages are installed:

- apt-get install hyperv-daemons
- apt-get install waagent # *(Optional but recommended for password resets and the use of extensions)*
- apt-get install cloud-init

Then perform a full upgrade:

- apt-get full-upgrade

Now the Azure resources must be created for this image. This example uses the `$rounded_size_adjusted` variable, so it should be from within the same shell process from the preceding step.

```
az group create -l $LOCATION -n $RG

az disk create \
    -n $DISK \
    -g $RG \
    -l $LOCATION \
    --for-upload --upload-size-bytes "$rounded_size_adjusted" \
    --sku standard_lrs --hyper-v-generation V1

ACCESS=$(az disk grant-access \
```

```
    -n $DISK -g $RG \
    --access-level write \
    --duration-in-seconds 86400 \
    --query accessSas -o tsv)

azcopy copy "$vhddisk" "$ACCESS" --blob-type PageBlob

az disk revoke-access -n $DISK -g $RG
az image create \
    -g $RG \
    -n $IMAGE \
    --os-type linux \
    --source $(az disk show \
        -g $RG \
        -n $DISK \
        --query id -o tsv)
az vm create \
    -g $RG \
    -n $VM \
    --ssh-key-value $SSH_KEY_VALUE \
    --public-ip-address-dns-name $VM \
    --image $(az image show \
        -g $RG \
        -n $IMAGE \
        --query id -o tsv)
```

If the bandwidth from your local machine to the Azure disk is causing a long time to process the upload with `azcopy`, you can use an Azure VM jumpbox to speed up the process. Here's how this process can be done:

1. Create a tarball of the VHD on your local machine: `tar -czvf ./image_buster_azure_amd64.vhd.tar.gz ./image_[release]_azure_amd64.vhd`.
2. Create an Azure Linux VM (distribution of your choice). Make sure that you create it with a large-enough disk to hold the extracted VHD.
3. Download the `azcopy` utility to the Azure Linux VM. You can retrieve it from Get started with AzCopy.
4. Copy the tarball to the VM: `scp ./image_buster_azure_amd64.vhd.tar.gz <vm>:~`.
5. On the VM, extract the VHD: `tar -xf ./image_buster_azure_amd64.vhd.tar.gz`. This step takes a bit of time based on the size of the file.
6. Finally, on the VM, copy the VHD to the Azure disk with `azcopy` (the preceding command).

Related content

You're now ready to use your Debian Linux VHD to create new VMs in Azure. If this is the first time that you're uploading the .vhd file to Azure, see Create a Linux VM from a

custom disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Using a prebuilt Flatcar image for Azure

Applies to: ✔ Linux VMs ✔ Flexible scale sets

You can download prebuilt Azure virtual hard disk images of Flatcar Container Linux for each of the Flatcar supported channels:

- stable
- beta
- alpha
- LTS

This image is already fully set up and optimized to run on Azure. You only need to decompress it.

Building your own Flatcar-based virtual machine for Azure

Alternatively, you can choose to build your own Flatcar Container Linux image.

On any linux based machine, follow the instructions detailed in the Flatcar Container Linux developer SDK guide. When running the `image_to_vm.sh` script, make sure you pass `--format=azure` to create an Azure virtual hard disk.

Next steps

Once you have the .vhd file, you can use the resulting file to create new virtual machines in Azure. If this is the first time that you're uploading a .vhd file to Azure, see Create a Linux VM from a custom disk.

Last updated on 01/22/2026

Introduction to FreeBSD on Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article provides an overview of running a FreeBSD virtual machine (VM) in Azure.

Overview

FreeBSD for Azure is an advanced computer operating system used to power modern servers, desktops, and embedded platforms.

Microsoft is making images of FreeBSD available on Azure with the Azure VM Guest Agent preconfigured. Currently, the following FreeBSD versions are offered as images by Microsoft:

- FreeBSD 10.4 in Azure Marketplace
- FreeBSD 11.2 in Azure Marketplace
- FreeBSD 11.3 in Azure Marketplace
- FreeBSD 12.0 in Azure Marketplace

The following FreeBSD versions also include the Azure VM Guest Agent. They're offered as images by the FreeBSD Foundation:

- FreeBSD 11.4 in Azure Marketplace
- FreeBSD 12.2 in Azure Marketplace
- FreeBSD 13.0 in Azure Marketplace

The agent is responsible for communication between the FreeBSD VM and the Azure fabric for operations such as provisioning the VM on first use (user name, password or SSH key, and host name) and enabling functionality for selective VM extensions.

As for future versions of FreeBSD, the strategy is to stay current and make the latest releases available shortly after they're published by the FreeBSD release engineering team.

Create a FreeBSD VM through the Azure CLI on FreeBSD

First, you need to install the Azure CLI through the following command on a FreeBSD machine:

```
bash
```

```
curl -L https://aka.ms/InstallAzureCli | bash
```

If bash isn't installed on your FreeBSD machine, run the following command before the installation:

Bash

```
sudo pkg install bash
```

If Python isn't installed on your FreeBSD machine, run the following commands before the installation:

Bash

```
sudo pkg install python38
cd /usr/local/bin
sudo rm /usr/local/bin/python
sudo ln -s /usr/local/bin/python3.8 /usr/local/bin/python
```

During the installation, you're asked to `Modify profile to update your $PATH and enable shell/tab completion now? (Y/n)`. If you answer `y` and enter `/etc/rc.conf` as `a path to an rc file to update`, you might see `ERROR: [Errno 13] Permission denied`. To resolve this problem, you should grant the write permission to the current user against the file `etc/rc.conf`.

Now you can sign in to Azure and create your FreeBSD VM. The following example shows how to create a FreeBSD 11.0 VM. You can also add the parameter `--public-ip-address-dns-name` with a globally unique DNS name for a newly created public IP.

Azure CLI

```
az login
az group create --name myResourceGroup --location eastus
az vm create --name myFreeBSD11 \
    --resource-group myResourceGroup \
    --image MicrosoftOSTC:FreeBSD:11.0:latest \
    --admin-username azureuser \
    --generate-ssh-keys
```

Then you can sign in to your FreeBSD VM through the IP address that printed in the output of the preceding deployment.

Bash

```
ssh azureuser@xx.xx.xx.xx -i /etc/ssh/ssh_host_rsa_key
```

VM extensions for FreeBSD

The following VM extensions are supported in FreeBSD.

VMAccess

The VMAccess extension can:

- Reset the password of the original sudo user.
- Create a new sudo user with the password specified.
- Set the public host key with the key given.
- Reset the public host key provided during VM provisioning if the host key isn't provided.
- Open the SSH port (22) and restore the `sshd_config` if `reset_ssh` is set to `true`.
- Remove the existing user.
- Check disks.
- Repair an added disk.

CustomScript

The CustomScript extension can:

- If provided, download the customized scripts from Azure Storage or external public storage (for example, GitHub).
- Run the entry point script.
- Support inline commands.
- Convert Windows-style newline in shell and Python scripts automatically.
- Remove BOM in shell and Python scripts automatically.
- Protect sensitive data in `CommandToExecute`.

> **Note**
>
> FreeBSD VM only supports CustomScript version 1.x by now.

Authentication: User names, passwords, and SSH keys

When you're creating a FreeBSD VM by using the Azure portal, you must provide a user name, password, or SSH public key.

User names for deploying a FreeBSD VM on Azure must not match the names of system accounts (UID <100) already present in the VM ("root," for example).

Currently, only the RSA SSH key is supported. A multiline SSH key must begin with `---- BEGIN SSH2 PUBLIC KEY ----` and end with `---- END SSH2 PUBLIC KEY ----`.

Obtain superuser privileges

The user account that's specified during VM instance deployment on Azure is a privileged account. The package of sudo was installed in the published FreeBSD image.

After you're logged in through this user account, you can run commands as root by using the command syntax.

```
$ sudo <COMMAND>
```

You can optionally obtain a root shell by using `sudo -s`.

Known issues

The Azure VM Guest Agent version 2.2.2 has a known issue that causes the provision failure for FreeBSD VM on Azure. The fix was captured by Azure VM Guest Agent version 2.2.3 and later releases.

Related content

- Go to Azure Marketplace to create a FreeBSD VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Prepare an Oracle Linux virtual machine for Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article assumes that you've already installed an Oracle Linux operating system (OS) to a virtual hard disk (VHD). Multiple tools exist to create .vhd files. An example is a virtualization solution such as Hyper-V. For instructions, see Install the Hyper-V role and configure a virtual machine (VM).

Oracle Linux installation notes

- For more tips on preparing Linux for Azure, see General Linux installation notes.
- Hyper-V and Azure support Oracle Linux with either the Unbreakable Enterprise Kernel (UEK) or the Red Hat Compatible Kernel.
- Oracle's UEK2 isn't supported on Hyper-V and Azure because it doesn't include the required drivers.
- The VHDX format isn't supported in Azure, only *fixed VHD*. You can convert the disk to VHD format by using Hyper-V Manager or the `convert-vhd` cmdlet.
- *Kernel support for mounting user-defined functions (UDF) file systems is required.* At first boot on Azure, the provisioning configuration is passed to the Linux VM via UDF-formatted media that's attached to the guest. The Azure Linux agent must be able to mount the UDF file system to read its configuration and provision the VM.
- When you install the Linux system, we recommend that you use standard partitions rather than Logical Volume Manager (LVM), which is often the default for many installations. These standard partitions avoid LVM name conflicts with cloned VMs, particularly if an OS disk ever needs to be attached to another VM for troubleshooting. LVM or RAID can also be used on data disks.
- Linux kernel versions earlier than 2.6.37 don't support NUMA on Hyper-V with larger VM sizes. This issue primarily affects older distributions that use the upstream Red Hat 2.6.32 kernel and was fixed in Oracle Linux 6.6 and later.
- Don't configure a swap partition on the OS disk.
- All VHDs on Azure must have a virtual size aligned to 1 MB. When you convert from a raw disk to VHD, you must ensure that the raw disk size is a multiple of 1 MB before conversion. For more information, see Linux installation notes.
- Make sure that the `Addons` repository is enabled. Edit the file `/etc/yum.repos.d/public-yum-ol6.repo` (Oracle Linux 6) or

`/etc/yum.repos.d/public-yum-ol7.repo` (Oracle Linux 7). Change the line `enabled=0` to `enabled=1` under [ol6_addons] or [ol7_addons] in this file.

Oracle Linux 6.X

> **ⓘ Important**
>
> Remember that Oracle Linux 6.x is already at end of life. Oracle Linux version 6.10 has available Extended Lifecycle Support, which ends July 2024.

You must complete specific configuration steps in the OS for the VM to run in Azure.

1. In the center pane of Hyper-V Manager, select the VM.
2. Select **Connect** to open the window for the VM.
3. Uninstall `NetworkManager`:

   ```
   Bash

   sudo rpm -e --nodeps NetworkManager
   ```

   > **ⓘ Note**
   >
   > If the package isn't already installed, this command fails with an error message. This message is expected.

4. Create a file named **network** in the `/etc/sysconfig/` directory that contains the following text:

   ```
   config

   NETWORKING=yes
   HOSTNAME=localhost.localdomain
   ```

5. Create a file named **ifcfg-eth0** in the `/etc/sysconfig/network-scripts/` directory that contains the following text:

   ```
   config

   DEVICE=eth0
   ONBOOT=yes
   ```

```
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
```

6. Modify udev rules to avoid generating static rules for the Ethernet interfaces. These rules can cause problems when you clone a VM in Azure or Hyper-V:

```Bash
sudo ln -s /dev/null /etc/udev/rules.d/75-persistent-net-
generator.rules
sudo rm -f /etc/udev/rules.d/70-persistent-net.rules
```

7. Ensure that the network service starts at boot time:

```Bash
sudo chkconfig network on
```

8. Install `python-pyasn1`:

```Bash
sudo yum install python-pyasn1
```

9. Modify the kernel boot line in your grub configuration to include more kernel parameters for Azure. To do this step, open `/boot/grub/menu.lst` in a text editor and ensure that the kernel includes the following parameters:

```config-grub
console=ttyS0 earlyprintk=ttyS0
```

   This setting ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues.

   In addition to the preceding steps, we recommend that you *remove* the following parameters:

```config-grub
rhgb quiet crashkernel=auto
```

Graphical and quiet boot aren't useful in a cloud environment where you want all the logs to be sent to the serial port.

The `crashkernel` option can be left configured if you want. This parameter reduces the amount of available memory in the VM by 128 MB or more, which might be a problem for smaller VM sizes.

10. Ensure that the SSH server is installed and configured to start at boot time. This setting is usually the default.

11. Install the Azure Linux agent by running the following command. The latest version is 2.0.15.

```Bash
sudo yum install WALinuxAgent
```

Installing the `WALinuxAgent` package removes the `NetworkManager` and `NetworkManager-gnome` packages if they weren't already removed, as described in step 2.

12. Don't create swap space on the OS disk.

The Azure Linux agent can automatically configure swap space by using the local resource disk that's attached to the VM after provisioning on Azure. The local resource disk is a *temporary* disk and might be emptied when the VM is deprovisioned. After you install the Azure Linux agent (see the previous step), modify the following parameters in `/etc/waagent.conf` appropriately:

```config-conf
ResourceDisk.Format=y
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt
ResourceDisk.EnableSwap=y
ResourceDisk.SwapSizeMB=2048    ## NOTE: set this to whatever you need
it to be.
```

13. Deprovision the VM and prepare it for provisioning on Azure:

```Bash
sudo waagent -force -deprovision
sudo export HISTSIZE=0
sudo logout
```

14. Select **Action** > **Shut Down** in Hyper-V Manager. Your Linux VHD is now ready to be uploaded to Azure.

Oracle Linux 7.0 and later

Follow the steps in the next sections if you're using Oracle Linux 7.0 or later.

Changes in Oracle Linux 7

Preparing an Oracle Linux 7 VM for Azure is similar to Oracle Linux 6, but several differences are worth noting:

- Azure supports Oracle Linux with either the Unbreakable Enterprise Kernel (UEK) or the Red Hat Compatible Kernel. We recommend that you use Oracle Linux with UEK.
- The `NetworkManager` package no longer conflicts with the Azure Linux agent. This package is installed by default, and we recommend that you don't remove it.
- GRUB2 is now used as the default bootloader, so the procedure for editing kernel parameters has changed. (See the "Configuration steps" section.)
- XFS is now the default file system. The ext4 file system can still be used if you want.

Configuration steps

1. In Hyper-V Manager, select the VM.
2. Select **Connect** to open a console window for the VM.
3. Create a file named **network** in the `/etc/sysconfig/` directory that contains the following text:

   ```
   config

   NETWORKING=yes
   HOSTNAME=localhost.localdomain
   ```

4. Create a file named **ifcfg-eth0** in the `/etc/sysconfig/network-scripts/` directory that contains the following text:

   ```
   config

   DEVICE=eth0
   ONBOOT=yes
   ```

```
BOOTPROTO=dhcp
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=no
```

5. Modify udev rules to avoid generating static rules for the Ethernet interfaces. These rules can cause problems when you clone a VM in Azure or Hyper-V:

```
Bash

sudo ln -s /dev/null /etc/udev/rules.d/75-persistent-net-
generator.rules
```

6. Ensure that the network service starts at boot time:

```
Bash

sudo chkconfig network on
```

7. Install the `python-pyasn1` package:

```
Bash

sudo yum install python3-pyasn1
```

8. Clear the current yum metadata and install any updates:

```
Bash

sudo yum clean all
sudo yum -y update
```

9. Modify the kernel boot line in your grub configuration to include more kernel parameters for Azure. To do this step, open `/etc/default/grub` in a text editor and edit the `GRUB_CMDLINE_LINUX` parameter. For example:

```
config-grub

GRUB_CMDLINE_LINUX="console=ttyS0 earlyprintk=ttyS0 net.ifnames=0"
```

   This modification also ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues. It also turns off the

naming conventions for network interface cards in Oracle Linux 7 with the UEK. We also recommend that you *remove* the following parameters:

config-grub

```
rhgb quiet crashkernel=auto
```

Graphical and quiet boot aren't useful in a cloud environment where you want all the logs to be sent to the serial port.

The `crashkernel` option can be left configured if you want. This parameter reduces the amount of available memory in the VM by 128 MB or more, which might be a problem for smaller VM sizes.

10. After you're finished editing `/etc/default/grub`, rebuild the grub configuration:

    Bash

    ```
    sudo grub2-mkconfig -o /boot/grub2/grub.cfg
    ```

11. Ensure that the SSH server is installed and configured to start at boot time. This setting is usually the default.

12. Install the Azure Linux agent and dependencies:

    Bash

    ```
    sudo yum install WALinuxAgent
    sudo systemctl enable waagent
    ```

13. Install `cloud-init` to handle the provisioning:

    Bash

    ```
    sudo yum install -y cloud-init cloud-utils-growpart gdisk hyperv-
    daemons
    ```

14. Configure `waagent` for `cloud-init`:

    Bash

    ```
    sudo sed -i
    's/Provisioning.UseCloudInit=n/Provisioning.UseCloudInit=y/g'
    /etc/waagent.conf
    ```

```
sudo sed -i 's/Provisioning.Enabled=y/Provisioning.Enabled=n/g'
/etc/waagent.conf
```

Bash

```
sudo echo "Adding mounts and disk_setup to init stage"
sudo sed -i '/ - mounts/d' /etc/cloud/cloud.cfg
sudo sed -i '/ - disk_setup/d' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - mounts' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - disk_setup' /etc/cloud/cloud.cfg
```

Bash

```
echo "Allow only Azure datasource, disable fetching network setting via
IMDS"
```

Bash

```
sudo cat > /etc/cloud/cloud.cfg.d/91-azure_datasource.cfg <<EOF
datasource_list: [ Azure ]
datasource:
    Azure:
        apply_network_config: False
EOF


if [[ -f /mnt/resource/swapfile ]]; then
echo Removing swapfile - Oracle Linux uses a swapfile by default
swapoff /mnt/swapfile
rm /mnt/swapfile -f
fi

echo "Add console log file"
cat >> /etc/cloud/cloud.cfg.d/05_logging.cfg <<EOF

# This tells cloud-init to redirect its stdout and stderr to
# 'tee -a /var/log/cloud-init-output.log' so the user can see output
# there without needing to look on the console.
output: {all: '| tee -a /var/log/cloud-init-output.log'}
EOF
```

15. Swap configuration:

    a. Don't create swap space on the operating system disk.

    Previously, the Azure Linux agent was used automatically to configure swap space by using the local resource disk that's attached to the VM after the VM is provisioned on Azure. However, `cloud-init` now handles this step. You *must not*

use the Linux agent to format the resource disk to create the swap file. Modify the following parameters in `/etc/waagent.conf` appropriately:

Bash

```
sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g'
/etc/waagent.conf
sudo sed -i
's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
/etc/waagent.conf
```

b. If you want to mount, format, and create the swap, you can either:

- Pass this code in as a `cloud-init` configuration every time you create a VM.
- Use a `cloud-init` directive baked into the image to do this step every time the VM is created:

Bash

```
echo 'DefaultEnvironment="CLOUD_CFG=/etc/cloud/cloud.cfg.d/00-azure-
swap.cfg"' >> /etc/systemd/system.conf
cat > /etc/cloud/cloud.cfg.d/00-azure-swap.cfg << EOF
#cloud-config
# Generated by Azure cloud image build
disk_setup:
  ephemeral0:
    table_type: mbr
    layout: [66, [33, 82]]
    overwrite: True
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
  - ["ephemeral0.1", "/mnt/resource"]
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-
systemd.requires=cloud-init.service,x-systemd.device-timeout=2",
"0", "0"]
EOF
```

16. Deprovision the VM and prepare it for provisioning on Azure:

Bash

```
sudo cloud-init clean
sudo rm -f /var/log/waagent.log
sudo waagent -force -deprovision
```

```
sudo rm -f ~/.bash_history
sudo export HISTSIZE=0
```

> **Note**
>
> If you're migrating a specific VM and don't want to create a generalized image, skip the deprovision step.

17. Select **Action** > **Shut Down** in Hyper-V Manager. Your Linux VHD is now ready to be uploaded to Azure.

Related content

You're now ready to use your Oracle Linux .vhd to create new VMs in Azure. If this is the first time that you're uploading the .vhd file to Azure, see Create a Linux VM from a custom disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create and upload an OpenBSD disk image to Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to create and upload a virtual hard disk (VHD) that contains the OpenBSD operating system. After you upload it, you can use it as your own image to create a virtual machine (VM) in Azure through the Azure CLI.

Prerequisites

This article assumes that you have the following items:

- **An Azure subscription**: If you don't have an account, you can create one in a couple of minutes. If you have an MSDN subscription, see Monthly Azure credit for Visual Studio subscribers. Otherwise, learn how to create a free trial account.
- **Azure CLI**: Make sure you have the latest Azure CLI installed and logged in to your Azure account with az login.
- **OpenBSD operating system installed in a .vhd file**: A supported OpenBSD operating system (6.6 version AMD64) must be installed to a VHD. Multiple tools exist to create .vhd files. For example, you can use a virtualization solution such as Hyper-V to create the .vhd file and install the operating system. For instructions about how to install and use Hyper-V, see Install Hyper-V and create a VM.

Prepare an OpenBSD image for Azure

On the VM where you installed the OpenBSD operating system 6.1, which added Hyper-V support, complete the following procedures:

1. If DHCP isn't enabled during installation, enable the service:

   ```sh
   doas echo dhcp > /etc/hostname.hvn0
   ```

2. Set up a serial console:

   ```sh
   ```

```
doas echo "stty com0 115200" >> /etc/boot.conf
doas echo "set tty com0" >> /etc/boot.conf
```

3. Configure package installation:

```sh
doas echo "https://ftp.openbsd.org/pub/OpenBSD" > /etc/installurl
```

4. By default, the `root` user is disabled on VMs in Azure. Users can run commands with elevated privileges by using the `doas` command on OpenBSD VM. Doas is enabled by default.

5. Install and configure prerequisites for the Azure agent:

```sh
doas pkg_add py-setuptools openssl git
doas ln -sf /usr/local/bin/python2.7 /usr/local/bin/python
doas ln -sf /usr/local/bin/python2.7-2to3 /usr/local/bin/2to3
doas ln -sf /usr/local/bin/python2.7-config /usr/local/bin/python-
config
doas ln -sf /usr/local/bin/pydoc2.7  /usr/local/bin/pydoc
```

6. You can find the latest release of the Azure agent on GitHub. Install the agent:

```sh
doas git clone https://github.com/Azure/WALinuxAgent
doas cd WALinuxAgent
doas python setup.py install
doas waagent -register-service
```

> **Important**
>
> After you install the Azure agent, it's a good idea to verify that it's running:
>
> ```sh
> doas ps auxw | grep waagent
> root     79309  0.0  1.5  9184 15356 p1  S      4:11PM    0:00.46
> python /usr/local/sbin/waagent -daemon (python2.7)
> doas cat /var/log/waagent.log
> ```

7. Deprovision the system to clean it and make it suitable for deprovisioning. The following command also deletes the last provisioned user account and the associated data:

```sh
doas waagent -deprovision+user -force
```

> **Note**
>
> If you're migrating a specific VM and don't want to create a generalized image, skip the deprovision step.

Now you can shut down your VM.

Prepare the VHD

The VHDX format isn't supported in Azure, only *fixed VHD*. You can convert the disk to fixed VHD format by using Hyper-V Manager or the PowerShell convert-vhd cmdlet. Here's an example:

```PowerShell
Convert-VHD OpenBSD61.vhdx OpenBSD61.vhd -VHDType Fixed
```

Create storage resources and upload

First, create a resource group with az group create. The following example creates a resource group named `myResourceGroup` in the `eastus` location:

```Azure CLI
az group create --name myResourceGroup --location eastus
```

To upload your VHD, create a storage account with az storage account create. Storage account names must be unique, so provide your own name. The following example creates a storage account named `mystorageaccount`:

Azure CLI

```
az storage account create --resource-group myResourceGroup \
    --name mystorageaccount \
    --location eastus \
    --sku Premium_LRS
```

To control access to the storage account, obtain the storage key with the az storage account keys list:

Azure CLI

```
STORAGE_KEY=$(az storage account keys list \
    --resource-group myResourceGroup \
    --account-name mystorageaccount \
    --query "[?keyName=='key1']  | [0].value" -o tsv)
```

To logically separate the VHDs you upload, create a container within the storage account with az storage container create:

Azure CLI

```
az storage container create \
    --name vhds \
    --account-name mystorageaccount \
    --account-key ${STORAGE_KEY}
```

Finally, upload your VHD with az storage blob upload:

Azure CLI

```
az storage blob upload \
    --container-name vhds \
    --file ./OpenBSD61.vhd \
    --name OpenBSD61.vhd \
    --account-name mystorageaccount \
    --account-key ${STORAGE_KEY}
```

Create a VM from your VHD

You can create a VM with a sample script or directly with az vm create. To specify the OpenBSD VHD you uploaded, use the `--image` parameter:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
```

```
    --name myOpenBSD61 \
    --image
"https://mystorageaccount.blob.core.windows.net/vhds/OpenBSD61.vhd" \
    --os-type linux \
    --admin-username azureuser \
    --ssh-key-value ~/.ssh/id_rsa.pub
```

Obtain the IP address for your OpenBSD VM with az vm list-ip-addresses:

Azure CLI

```
az vm list-ip-addresses --resource-group myResourceGroup --name myOpenBSD61
```

Now you can SSH to your OpenBSD VM as normal:

Bash

```
ssh azureuser@<ip address>
```

Related content

- To learn more about Hyper-V support on OpenBSD 6.1, read OpenBSD 6.1.
- To create a VM from a managed disk, read az disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Prepare a Red Hat-based virtual machine for Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

In this article, you learn how to prepare a Red Hat Enterprise Linux (RHEL) virtual machine (VM) for use in Azure. The versions of RHEL that are covered in this article are 7.x, 8.X., 9.x. The hypervisors for preparation that are covered in this article are Hyper-V, kernel-based VM (KVM), VMware, and Kickstart.

For more information about eligibility requirements for participating in Red Hat's Cloud Access program, see the Red Hat Cloud Access website and Running RHEL on Azure. For ways to automate building RHEL images, see Azure Image Builder.

Prerequisites

> **Note**
>
> This document assumes you already deployed RHEL7 and have it fully licensed. Uploaded images that are at or beyond EOL are supported on a reasonable business-effort basis. For more information, see the Red Hat Product Life Cycles.

RHEL installation notes

- Azure doesn't support the VHDX format. Azure supports only *fixed VHD*. You can use Hyper-V Manager to convert the disk to VHD format, or you can use the `convert-vhd` cmdlet. If you use VirtualBox, select **Fixed size** as opposed to the default dynamically allocated option when you create the disk.
- Azure supports Gen1 (BIOS boot) and Gen2 (UEFI boot) VMs.
- The maximum allowed size for the VHD is 1,023 GB.
- The vfat kernel module must be enabled in the kernel.
- Logical Volume Manager (LVM) is supported and can be used on the OS disk or data disks in Azure VMs. In general, we recommend that you use standard partitions on the OS disk rather than LVM. This practice avoids LVM name conflicts with cloned VMs, particularly if you ever need to attach an operating system disk to another identical VM for troubleshooting. For more information, see the LVM and RAID documentation.
- *Kernel support for mounting Universal Disk Format (UDF) file systems is required*. At first boot on Azure, the attached UDF-formatted media passes the provisioning configuration

to the Linux VM. The Azure Linux agent must be able to mount the UDF file system to read its configuration and provision the VM. Without this step, provisioning fails.

- Don't configure a swap partition on the operating system disk. For more information, read the following steps.
- All VHDs on Azure must have a virtual size aligned to 1 MB. When you convert from a raw disk to VHD, you must ensure that the raw disk size is a multiple of 1 MB before conversion. For more information, read the following steps. See also Linux installation notes.

> **Note**
>
> *Cloud-init >= 21.2 removes the UDF requirement*. However, without the UDF module enabled, the provided CD-ROM fails to mount, preventing custom data from being applied. A workaround is to apply custom data by using user data. Unlike custom data, user data isn't encrypted. For more information, see User data formats.

RHEL 7 using Hyper-V Manager

1. In Hyper-V Manager, select the VM.
2. Select **Connect** to open a console window for the VM.
3. Create or edit the `/etc/sysconfig/network` file, and add the following text:

   ```config
   NETWORKING=yes
   HOSTNAME=localhost.localdomain
   ```

4. Create or edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file, and add the following text:

   ```config
   DEVICE=eth0
   ONBOOT=yes
   BOOTPROTO=dhcp
   TYPE=Ethernet
   USERCTL=no
   PEERDNS=yes
   IPV6INIT=no
   PERSISTENT_DHCLIENT=yes
   NM_CONTROLLED=yes
   ```

> **Note**
>
> When you use Accelerated Networking, the provisioned synthetic interface must be configured to be unmanaged by using a udev rule. This action prevents `NetworkManager` from assigning the same IP to it as the primary interface.

To apply it:

Bash

```
sudo tee <<EOF /etc/udev/rules.d/68-azure-sriov-nm-unmanaged.rules >
/dev/null
# Accelerated Networking on Azure exposes a new SRIOV interface to the VM.
# This interface is transparently bonded to the synthetic interface,
# so NetworkManager should just ignore any SRIOV interfaces.
SUBSYSTEM=="net", DRIVERS=="hv_pci", ACTION!="remove",
ENV{NM_UNMANAGED}="1"
EOF
```

5. Ensure that the network service starts at boot time:

Bash

```
sudo systemctl enable network
```

6. Register your Red Hat subscription to enable the installation of packages from the RHEL repository:

Bash

```
sudo subscription-manager register --auto-attach --username=XXX --
password=XXX
```

7. Modify the kernel boot line in your grub configuration to include more kernel parameters for Azure. To do this modification, open `/etc/default/grub` in a text editor and edit the `GRUB_CMDLINE_LINUX` parameter. For example:

config-grub

```
GRUB_TIMEOUT=10
GRUB_CMDLINE_LINUX="console=tty1 console=ttyS0,115200n8 earlyprintk=ttyS0
net.ifnames=0 nvme_core.io_timeout=240"
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no -
-stop=1"
```

These modifications turn off the new naming convention for NICs, ensure all console messages are sent to the first serial port, and enable interaction with the console, which can assist Aure support with debugging issues.

```config
rhgb quiet crashkernel=auto
```

Graphical and quiet boots aren't useful in a cloud environment where you want all the logs to be sent to the serial port. You can leave the `crashkernel` option configured if you want. This parameter reduces the amount of available memory in the VM by 128 MB or more, which might be a problem for smaller VM sizes.

8. After you're finished editing `/etc/default/grub`, run the following command to rebuild the grub configuration:

```Bash
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

> **Note**
>
> If you're uploading a UEFI-enabled VM, the command to update grub is `grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg`.

9. Ensure that the SSH server is installed and configured to start at boot time, which is usually the default. Modify `/etc/ssh/sshd_config` to include the following line:

```config
ClientAliveInterval 180
```

10. The WALinuxAgent package, `WALinuxAgent-<version>`, is available from the Extras repository.

```Bash
sudo subscription-manager repos --enable=rhel-7-server-extras-rpms
```

11. Install the Azure Linux agent, `cloud-init`, and other necessary utilities:

```Bash
```

```
sudo yum install -y WALinuxAgent cloud-init cloud-utils-growpart gdisk
hyperv-daemons
sudo systemctl enable waagent.service
sudo systemctl enable cloud-init.service
```

12. Configure `cloud-init` to handle the provisioning:

    a. Configure `waagent` for `cloud-init`:

    Bash

    ```
    sudo sed -i 's/Provisioning.Agent=auto/Provisioning.Agent=auto/g'
    /etc/waagent.conf
    sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g'
    /etc/waagent.conf
    sudo sed -i 's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
    /etc/waagent.conf
    ```

    > **Note**
    >
    > If you're migrating a specific VM and don't want to create a generalized image, set `Provisioning.Agent=disabled` on the `/etc/waagent.conf` configuration.

    b. Configure mounts:

    Bash

    ```
    sudo echo "Adding mounts and disk_setup to init stage"
    sudo sed -i '/ - mounts/d' /etc/cloud/cloud.cfg
    sudo sed -i '/ - disk_setup/d' /etc/cloud/cloud.cfg
    sudo sed -i '/cloud_init_modules/a\\ - mounts' /etc/cloud/cloud.cfg
    sudo sed -i '/cloud_init_modules/a\\ - disk_setup' /etc/cloud/cloud.cfg
    ```

    c. Configure the Azure data source:

    Bash

    ```
    sudo echo "Allow only Azure datasource, disable fetching network setting
    via IMDS"
    sudo tee /etc/cloud/cloud.cfg.d/91-azure_datasource.cfg <<EOF
    datasource_list: [ Azure ]
    datasource:
        Azure:
            apply_network_config: False
    EOF
    ```

d. If configured, remove the existing swap file:

```bash
if [[ -f /mnt/resource/swapfile ]]; then
echo "Removing swapfile" #RHEL uses a swapfile by default
swapoff /mnt/resource/swapfile
rm /mnt/resource/swapfile -f
fi
```

e. Configure `cloud-init` logging:

```bash
sudo echo "Add console log file"
sudo tee <<EOF /etc/cloud/cloud.cfg.d/05_logging.cfg > /dev/null

# This tells cloud-init to redirect its stdout and stderr to
# 'tee -a /var/log/cloud-init-output.log' so the user can see output
# there without needing to look on the console.
output: {all: '| tee -a /var/log/cloud-init-output.log'}
EOF
```

13. Swap configuration:

- Don't create swap space on the operating system disk.

  Previously, the Azure Linux agent was used to automatically configure swap space on the local resource disk attached to the VM after the VM is provisioned on Azure, `cloud-init` handles now this action. You *must not* use the Linux agent to format the resource disk to create the swap file. Modify the following parameters in `/etc/waagent.conf` appropriately:

  ```config
  ResourceDisk.Format=n
  ResourceDisk.EnableSwap=n
  ```

- If you want to mount, format, and create the swap, you can either:
  - Pass this code in as a `cloud-init` configuration every time you create a VM through custom data. We recommend this method.
  - Use a `cloud-init` directive baked into the image that does this step every time the VM is created.

Bash

```
sudo echo 'DefaultEnvironment="CLOUD_CFG=/etc/cloud/cloud.cfg.d/00-azure-swap.cfg"' >> /etc/systemd/system.conf
sudo tee /etc/cloud/cloud.cfg.d/00-azure-swap.cfg <<EOF
#cloud-config
# Generated by Azure cloud image build
disk_setup:
  ephemeral0:
    table_type: mbr
    layout: [66, [33, 82]]
    overwrite: True
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
  - ["ephemeral0.1", "/mnt/resource"]
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-systemd.requires=cloud-init.service,x-systemd.device-timeout=2", "0", "0"]
EOF
```

14. Configure `cloud-init` telemetry to assist with troubleshooting for provisioning issues:

    Bash

    ```
    sudo tee -a /etc/cloud/cloud.cfg.d/10-azure-kvp.cfg <<EOF
    # This config enables cloud-init to report provisioning telemetry to aid with troubleshooting
    Reporting:
      logging:
        type: log
      telemetry:
        type: hyperv
    EOF
    ```

15. If you want to unregister the subscription, run the following command:

    Bash

    ```
    sudo subscription-manager unregister
    ```

16. Deprovision the VM and prepare it for provisioning on Azure:

    ⊗ Caution

If you're migrating a specific VM and don't want to create a generalized image, skip the deprovision step. Running the command `waagent -force -deprovision+user` renders the source machine unusable. This step is intended only to create a generalized image.

Bash

```
sudo rm -f /var/log/waagent.log
sudo cloud-init clean --logs --seed
sudo waagent -force -deprovision+user
sudo rm -f ~/.bash_history
sudo export HISTSIZE=0
```

17. Select **Action** > **Shut Down** in Hyper-V Manager. Your Linux VHD is now ready to be uploaded to Azure.

Last updated on 02/11/2026

Prepare a SLES or openSUSE Leap virtual machine for Azure

09/02/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets **Applies to:** ✔ Uniform scale sets

In some cases, you might want to use customized SUSE Linux Enterprise Server (SLES) or openSUSE Leap Linux virtual machines (VMs) in your Azure environment and be able to build these types of VMs through automation. This article demonstrates how to create and upload a custom Azure virtual hard disk (VHD) that contains a SUSE Linux operating system.

Prerequisites

This article assumes that you already installed a SLES or openSUSE Leap Linux operating system on a virtual hard disk. Multiple tools exist to create .vhd files. For example, you can use a virtualization solution such as Hyper-V. For instructions, see Install Hyper-V and create a virtual machine.

SLES/openSUSE Leap installation notes

- For more tips on preparing Linux images for Azure, see General Linux installation notes.
- Azure doesn't support Windows Hard Disk Image (.vhdx) files. Only VHD (.vhd) files are supported outside virtual machines. You can convert the disk to VHD format by using Hyper-V Manager or the `Convert-VHD` cmdlet.
- Azure supports Gen1 (BIOS boot) and Gen2 (UEFI boot) virtual machines.
- The virtual file allocation table (VFAT) kernel module must be enabled in the kernel.
- Don't configure a swap partition on the OS disk. You can configure the Linux agent to create a swap file on the temporary resource disk. Steps later in this article give more information about configuring swap space.
- All VHDs on Azure must have a virtual size aligned to 1 MB. When you're converting from a raw disk to VHD, ensure that the raw disk size is a multiple of 1 MB before conversion. For more information, see General Linux installation notes.

> **Note**
>
> Cloud-init version 21.2 or later removes the user-defined function (UDF) requirement. But without the `udf` module enabled, the CD-ROM won't mount during provisioning, which prevents the custom data from being applied. A workaround is to apply user data.

However, unlike custom data, user data isn't encrypted. For more information, see User data formats in the cloud-init documentation.

Use SUSE Studio

SUSE Studio can easily create and manage your SLES and openSUSE Leap images for Azure and Hyper-V. SUSE Studio is the recommended approach for customizing your own SLES and openSUSE Leap images.

As an alternative to building your own VHD, SUSE also publishes BYOS (bring your own subscription) images for SLES at VM Depot.

Prepare SLES for Azure

1. Configure the Azure and Hyper-V modules if required.

   If your software hypervisor is not Hyper-V, other modules need to be added into the initial RAM disk (initramfs) to successfully boot in Azure.

   Edit the */etc/dracut.conf* file and add the following line to the file:

   ```config
   add_drivers+=" hv_vmbus hv_netvsc hv_storvsc "
   ```

   Run the `dracut` command to rebuild the initramfs file:

   ```Bash
   sudo dracut --verbose --force
   ```

2. Set up the serial console.

   To successfully work with the serial console, you must set up several variables in the */etc/defaults/grub* file and re-create GRUB on the server:

   ```config
   # Add console=ttyS0 and earlyprintk=ttS0 to the variable.
   # Remove "splash=silent" and "quiet" options.
   GRUB_CMDLINE_LINUX_DEFAULT="audit=1 no-scroll fbcon=scrollback:0
   mitigations=auto security=apparmor crashkernel=228M,high crashkernel=72M,low
   console=ttyS0 earlyprintk=ttyS0"
   ```

```
# Add "console serial" to GRUB_TERMINAL.
GRUB_TERMINAL="console serial"

# Set the GRUB_SERIAL_COMMAND variable.

GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --
stop=1"
```

shell

```
/usr/sbin/grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Register your SUSE Linux Enterprise system to allow it to download updates and install packages.

4. Update the system with the latest patches:

Bash

```
sudo zypper update
```

5. Install the Azure Linux VM Agent (`waagent`) and cloud-init:

Bash

```
sudo SUSEConnect -p sle-module-public-cloud/15.2/x86_64  (SLES 15 SP2)
sudo zypper refresh
sudo zypper install python-azure-agent
sudo zypper install cloud-init
```

6. Enable `waagent` and cloud-init to start on boot:

Bash

```
sudo systemctl enable  waagent
sudo systemctl enable cloud-init-local.service
sudo systemctl enable cloud-init.service
sudo systemctl enable cloud-config.service
sudo systemctl enable cloud-final.service
sudo systemctl daemon-reload
sudo cloud-init clean
```

7. Update the cloud-init configuration:

Bash

```
cat <<EOF | sudo tee /etc/cloud/cloud.cfg.d/91-azure_datasource.cfg
datasource_list: [ Azure ]
datasource:
    Azure:
        apply_network_config: False

EOF
```

Bash

```
sudo cat <<EOF | sudo tee  /etc/cloud/cloud.cfg.d/05_logging.cfg
# This tells cloud-init to redirect its stdout and stderr to
# 'tee -a /var/log/cloud-init-output.log' so the user can see output
# there without needing to look on the console.
output: {all: '| tee -a /var/log/cloud-init-output.log'}
EOF

# Make sure mounts and disk_setup are in the init stage:
echo "Adding mounts and disk_setup to init stage"
sudo sed -i '/ - mounts/d' /etc/cloud/cloud.cfg
sudo sed -i '/ - disk_setup/d' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - mounts' /etc/cloud/cloud.cfg
sudo sed -i '/cloud_init_modules/a\\ - disk_setup' /etc/cloud/cloud.cfg
```

8. If you want to mount, format, and create a swap partition, one option is to pass in a cloud-init configuration every time you create a VM.

   Another option is to use a cloud-init directive in the image to configure swap space every time the VM is created:

Bash

```
cat  <<EOF | sudo tee -a /etc/systemd/system.conf
'DefaultEnvironment="CLOUD_CFG=/etc/cloud/cloud.cfg.d/00-azure-swap.cfg"'
EOF

cat <<EOF | sudo tee /etc/cloud/cloud.cfg.d/00-azure-swap.cfg
#cloud-config
# Generated by Azure cloud image build
disk_setup:
  ephemeral0:
    table_type: mbr
    layout: [66, [33, 82]]
    overwrite: True
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
```

```
  - ["ephemeral0.1", "/mnt"]
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-systemd.requires=cloud-
init.service,x-systemd.device-timeout=2", "0", "0"]
EOF
```

9. Previously, the Azure Linux Agent was used to automatically configure swap space by using the local resource disk that's attached to the virtual machine after the virtual machine is provisioned on Azure. Because cloud-init now handles this step, you *must not* use the Azure Linux Agent to format the resource disk or create the swap file. Use these commands to modify */etc/waagent.conf* appropriately:

   Bash

   ```
   sudo sed -i 's/Provisioning.UseCloudInit=n/Provisioning.UseCloudInit=auto/g'
   /etc/waagent.conf
   sudo sed -i 's/Provisioning.Enabled=y/Provisioning.Enabled=n/g'
   /etc/waagent.conf
   sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g'
   /etc/waagent.conf
   sudo sed -i 's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
   /etc/waagent.conf
   ```

   > **Note**
   >
   > If you're using a cloud-init version earlier than 21.2, make sure the `udf` module is enabled. Removing or disabling it will cause a provisioning or boot failure. Cloud-init version 21.2 or later removes the UDF requirement.

10. Ensure that the */etc/fstab* file references the disk by using its UUID (`by-uuid`).

11. Remove udev rules and network adapter configuration files to avoid generating static rules for the Ethernet interfaces. These rules can cause problems when you're cloning a virtual machine in Microsoft Azure or Hyper-V.

    Bash

    ```
    sudo rm -f /etc/udev/rules.d/70-persistent-net.rules
    sudo rm -f /etc/udev/rules.d/85-persistent-net-cloud-init.rules
    sudo rm -f /etc/sysconfig/network/ifcfg-eth*
    ```

12. We recommend that you edit the */etc/sysconfig/network/dhcp* file and change the `DHCLIENT_SET_HOSTNAME` parameter to the following:

    config

```
DHCLIENT_SET_HOSTNAME="no"
```

13. In the */etc/sudoers* file, comment out or remove the following lines if they exist:

```
Output

Defaults targetpw   # Ask for the password of the target user i.e. root
ALL    ALL=(ALL) ALL   # WARNING! Only use this setting together with
'Defaults targetpw'!
```

14. Ensure that the Secure Shell (SSH) server is installed and configured to start at boot time:

```
Bash

sudo systemctl enable sshd
```

15. Clean the cloud-init stage:

```
Bash

sudo cloud-init clean --seed --logs
```

16. Run the following commands to deprovision the virtual machine and prepare it for provisioning on Azure.

    If you're migrating a specific virtual machine and don't want to create a generalized image, skip the deprovisioning step.

```
Bash

sudo rm -f /var/log/waagent.log
sudo waagent -force -deprovision+user
sudo export HISTSIZE=0
sudo rm -f ~/.bash_history
```

Prepare openSUSE 15.4+

1. On the center pane of Hyper-V Manager, select the virtual machine.
2. Select **Connect** to open the window for the virtual machine.
3. In a terminal, run the command `zypper lr`. If this command returns output similar to the following example, the repositories are configured as expected and no adjustments are

necessary. (Version numbers might vary.)

Expand table

#	Alias	Name	Enabled	GPG Check	Refresh
1	Cloud:Tools_15.4	Cloud:Tools->	Yes	(r) Yes	Yes
2	openSUSE_stable_OSS	openSUSE_st->	Yes	(r) Yes	Yes
3	openSUSE_stable_Updates	openSUSE_st->	Yes	(r) Yes	Yes

If the the message "***No repositories defined***" appears from the `zypper lr` the repositories must be added manually.

Below are examples of commands for adding these repositories (versions and links may vary):

Bash

```
sudo zypper ar -f https://download.opensuse.org/update/openSUSE-stable
openSUSE_stable_Updates
sudo zypper ar -f
https://download.opensuse.org/repositories/Cloud:/Tools/15.4 Cloud:Tools_15.4
sudo zypper ar -f https://download.opensuse.org/distribution/openSUSE-
stable/repo/oss openSUSE_stable_OSS
```

You can then verify that the repositories have been added by running the command `zypper lr` again. If one of the relevant update repositories isn't enabled, enable it by using the following command:

Bash

```
sudo zypper mr -e [NUMBER OF REPOSITORY]
```

4. Update the kernel to the latest available version:

Bash

```
sudo zypper up kernel-default
```

Or update the operating system with all the latest patches:

Bash

```
sudo zypper update
```

5. Install the Azure Linux Agent:

```Bash
sudo zypper install WALinuxAgent
```

6. Modify the kernel boot line in your GRUB configuration to include other kernel parameters for Azure. To do this, open */boot/grub/menu.lst* in a text editor and ensure that the default kernel includes the following parameters:

```config-grub
console=ttyS0 earlyprintk=ttyS0
```

   This option ensures that all console messages are sent to the first serial port, which can assist Azure support with debugging issues. In addition, remove the following parameters from the kernel boot line if they exist:

```config-grub
 libata.atapi_enabled=0 reserve=0x1f0,0x8
```

7. We recommend that you edit the */etc/sysconfig/network/dhcp* file and change the `DHCLIENT_SET_HOSTNAME` parameter to the following setting:

```config
 DHCLIENT_SET_HOSTNAME="no"
```

8. In the */etc/sudoers* file, comment out or remove the following lines if they exist. This is an important step.

```Output
Defaults targetpw   # ask for the password of the target user i.e. root
ALL    ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults
targetpw'!
```

9. Ensure that the SSH server is installed and configured to start at boot time.

10. Don't create swap space on the OS disk.

The Azure Linux Agent can automatically configure swap space by using the local resource disk that's attached to the VM after provisioning on Azure. The local resource disk is a *temporary* disk and will be emptied when the VM is deprovisioned.

After you install the Azure Linux Agent, modify the parameters in */etc/waagent.conf* as follows:

```config-conf
ResourceDisk.Format=n
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt/resource
ResourceDisk.EnableSwap=n
ResourceDisk.SwapSizeMB=2048    ## NOTE: set the size to whatever you need it
to be.
```

11. Ensure that the Azure Linux Agent runs at startup:

```Bash
sudo systemctl enable waagent.service
```

12. Run the following commands to deprovision the virtual machine and prepare it for provisioning on Azure.

    If you're migrating a specific virtual machine and don't want to create a generalized image, skip the deprovisioning step.

```Bash
sudo rm -f ~/.bash_history # Remove current user history
sudo rm -rf /var/lib/waagent/
sudo rm -f /var/log/waagent.log
sudo waagent -force -deprovision+user
sudo rm -f ~/.bash_history # Remove root user history
sudo export HISTSIZE=0
```

13. Select **Action** > **Shut Down** in Hyper-V Manager.

Next steps

You're now ready to use your SUSE Linux VHD to create new virtual machines in Azure. If this is the first time that you're uploading the .vhd file to Azure, see Create a Linux VM from a custom disk.

Prepare an Ubuntu virtual machine for Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Ubuntu provides official Azure-ready cloud images (cloudimg) that include all components required for successful provisioning on Azure, such as *cloud-init*, Azure-optimized kernels, Azure guest agent compatibility, and performance-tuned defaults for virtualized environments. These images are the recommended starting point when creating a custom VHD because they align with Azure VM provisioning requirements and reduce the manual configuration steps required.

Ubuntu publishes these images in multiple formats, including ready-to-use Azure virtual hard disks (VHDs), on the https://cloud-images.ubuntu.com/ website. If you need to build a specialized Ubuntu image for Azure, you can start with these validated VHD builds and customize them as needed. The latest image releases are available at the following locations:

- Ubuntu 20.04/Focal: focal-server-cloudimg-amd64-azure.vhd.tar.gz
- Ubuntu 22.04/Jammy: jammy-server-cloudimg-amd64-azure.vhd.tar.gz
- Ubuntu 24.04/Noble: noble-server-cloudimg-amd64-azure.vhd.tar.gz

Prerequisites

This article assumes that you've already installed an Ubuntu Linux operating system (OS) to a VHD. Multiple tools exist to create .vhd files. An example is a virtualization solution such as Hyper-V. For instructions, see Install the Hyper-V role and configure a virtual machine (VM).

Ubuntu installation notes

- For more tips on preparing Linux for Azure, see General Linux installation notes.
- The VHDX format isn't supported in Azure, only *fixed VHD*. You can convert the disk to VHD format by using Hyper-V Manager or the `Convert-VHD` cmdlet.
- When you install the Linux system, we recommend that you use standard partitions rather than Logical Volume Manager (LVM), which is often the default for many installations. These standard partitions avoid LVM name conflicts with cloned VMs, particularly if an OS disk ever needs to be attached to another VM for troubleshooting. LVM or RAID can also be used on data disks.
- Don't configure a swap partition or swap file on the OS disk. You can configure the `cloud-init` provisioning agent to create a swap file or a swap partition on the temporary

resource disk. For more information about this process, see Create a SWAP partition for an Azure Linux VM.

- All VHDs on Azure must have a virtual size aligned to 1 MB. When you convert from a raw disk to VHD, you must ensure that the raw disk size is a multiple of 1 MB before conversion. For more information, see Linux installation notes.

You can upload a prebuilt Ubuntu image directly to Azure and use the resulting VHD to create new virtual machines. If this is your first time uploading a `.vhd` file, see Create a Linux VM from a custom disk.

Manual steps

> **Note**
>
> Before you attempt to create your own custom Ubuntu image for Azure, consider using the prebuilt and tested images from the Ubuntu Cloud Images webpage instead.

> If you are using Windows to Extract the prebuilt it image , please use Windows WSL and run the following command

Bash

```
sudo tar --sparse -xvzf <image name>-azure.vhd.tar.gz
```

1. In the center pane of Hyper-V Manager, select the VM.
2. Select **Connect** to open the window for the VM.
3. Replace the current repositories in the image to use Ubuntu's Azure repository.

   Before you edit `/etc/apt/sources.list` `/etc/apt/sources.list.d/ubuntu.sources`, we recommend that you make a backup:

   For Ubuntu 22.04 and earlier

   Bash

   ```
   sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
   ```

   For Ubuntu 24.04 and later

Bash

```
sudo cp /etc/apt/sources.list.d/ubuntu.sources
/etc/apt/sources.list.d/ubuntu.sources.bak
```

For Ubuntu 22.04 and earlier

Bash

```
sudo sed -i
's#http://archive\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubuntu
#g' /etc/apt/sources.list
sudo sed -i 's#http://[a-z][a-
z]\.archive\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubuntu#g'
/etc/apt/sources.list
sudo sed -i
's#http://security\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubunt
u#g' /etc/apt/sources.list
sudo sed -i 's#http://[a-z][a-
z]\.security\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubuntu#g'
/etc/apt/sources.list
sudo apt update
```

For Ubuntu 24.04 and later

Bash

```
sudo sed -i
's#http://archive\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubuntu
#g' /etc/apt/sources.list.d/ubuntu.sources
sudo sed -i 's#http://[a-z][a-
z]\.archive\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubuntu#g'
/etc/apt/sources.list.d/ubuntu.sources
sudo sed -i
's#http://security\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubunt
u#g' /etc/apt/sources.list.d/ubuntu.sources
sudo sed -i 's#http://[a-z][a-
z]\.security\.ubuntu\.com/ubuntu#http://azure\.archive\.ubuntu\.com/ubuntu#g'
/etc/apt/sources.list.d/ubuntu.sources
sudo apt update
```

4. The Ubuntu Azure images use Azure-Tailored Kernels Azure-tailored kernel. Update the OS to the latest Azure-tailored kernel and install Azure Linux tools (including Hyper-V dependencies):

Bash

```
sudo apt update
sudo apt install linux-azure linux-image-azure linux-headers-azure linux-tools-
```

```
common linux-cloud-tools-common linux-tools-azure linux-cloud-tools-azure
sudo apt full-upgrade
sudo reboot
```

5. Modify the kernel boot line for GRUB to include extra kernel parameters for Azure. To do this step, open `/etc/default/grub` in a text editor, find the variable called `GRUB_CMDLINE_LINUX_DEFAULT` (or add it if needed), and edit it to include the following parameters:

config

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0,115200n8
earlyprintk=ttyS0,115200 rootdelay=300 quiet splash"
```

6. Save and close this file, and then run `sudo update-grub`. This step ensures that all console messages are sent to the first serial port, which can assist Azure technical support with debugging issues.

7. Ensure that the SSH server is installed and configured to start at boot time. This setting is usually the default.

8. Install `cloud-init` (the provisioning agent) and the Azure Linux agent (the guest extensions handler). `Cloud-init` uses `netplan` to configure the system network configuration (during provisioning and each subsequent boot) and `gdisk` to partition resource disks.

Bash

```
sudo apt update
sudo apt install cloud-init gdisk netplan.io walinuxagent && systemctl stop
walinuxagent
```

> ⓘ **Note**
>
> The `walinuxagent` package might remove the `NetworkManager` and `NetworkManager-gnome` packages, if they're installed.

9. Remove `cloud-init` default configurations and leftover `netplan` artifacts that might conflict with `cloud-init` provisioning on Azure: For Ubuntu 22.04 and earlier

Bash

```
sudo rm -f /etc/cloud/cloud.cfg.d/50-curtin-networking.cfg
/etc/cloud/cloud.cfg.d/curtin-preserve-sources.cfg /etc/cloud/cloud.cfg.d/99-
```

```
installer.cfg /etc/cloud/cloud.cfg.d/subiquity-disable-cloudinit-networking.cfg
sudo rm -f /etc/cloud/ds-identify.cfg
sudo rm -f /etc/netplan/*.yaml
```

For Ubuntu 24.04 and later

Bash

```
sudo rm -f /etc/cloud/cloud.cfg.d/curtin-preserve-sources.cfg
/etc/cloud/cloud.cfg.d/99-installer.cfg /etc/cloud/cloud.cfg.d/subiquity-
disable-cloudinit-networking.cfg
sudo rm -f /etc/cloud/ds-identify.cfg
sudo rm -f /etc/netplan/*.yaml
```

10. Configure `cloud-init` to provision the system by using the Azure data source:

Bash

```
sudo tee /etc/cloud/cloud.cfg.d/90_dpkg.cfg <<EOF
datasource_list: [ Azure ]
EOF

sudo tee /etc/cloud/cloud.cfg.d/90-azure.cfg <<EOF
system_info:
   package_mirrors:
     - arches: [i386, amd64]
       failsafe:
         primary: http://archive.ubuntu.com/ubuntu
         security: http://security.ubuntu.com/ubuntu
       search:
         primary:
           - http://azure.archive.ubuntu.com/ubuntu/
         security: []
     - arches: [armhf, armel, default]
       failsafe:
         primary: http://ports.ubuntu.com/ubuntu-ports
         security: http://ports.ubuntu.com/ubuntu-ports
EOF

sudo tee /etc/cloud/cloud.cfg.d/10-azure-kvp.cfg <<EOF
reporting:
  logging:
    type: log
  telemetry:
    type: hyperv
EOF
```

11. Configure the Azure Linux agent to rely on `cloud-init` to perform provisioning. For more information on these options, look at the WALinuxAgent project.

Bash

```
sudo sed -i 's/Provisioning.Enabled=y/Provisioning.Enabled=n/g'
/etc/waagent.conf
sudo sed -i 's/Provisioning.UseCloudInit=n/Provisioning.UseCloudInit=y/g'
/etc/waagent.conf
sudo sed -i 's/ResourceDisk.Format=y/ResourceDisk.Format=n/g' /etc/waagent.conf
sudo sed -i 's/ResourceDisk.EnableSwap=y/ResourceDisk.EnableSwap=n/g'
/etc/waagent.conf
```

Bash

```
sudo tee -a /etc/waagent.conf <<EOF
# For Azure Linux agent version >= 2.2.45, this is the option to configure,
# enable, or disable the provisioning behavior of the Linux agent.
# Accepted values are auto (default), waagent, cloud-init, or disabled.
# A value of auto means that the agent will rely on cloud-init to handle
# provisioning if it is installed and enabled, which in this case it will.
Provisioning.Agent=auto
EOF
```

12. Clean `cloud-init` and Azure Linux agent runtime artifacts and logs:

Bash

```
sudo cloud-init clean --logs --seed
sudo rm -rf /var/lib/cloud/
sudo systemctl stop walinuxagent.service
sudo rm -rf /var/lib/waagent/
sudo rm -f /var/log/waagent.log
```

13. Deprovision the VM and prepare it for provisioning on Azure.

> ⓘ **Note**
>
> The `sudo waagent -force -deprovision+user` command generalizes the image by attempting to clean the system and make it suitable for reprovisioning. The `+user` option deletes the last provisioned user account and associated data.

Bash

```
sudo waagent -force -deprovision+user
sudo rm -f ~/.bash_history
```

> ⚠ **Warning**

> Deprovisioning by using the preceding command doesn't guarantee that the image is cleared of all sensitive information and is suitable for redistribution.

14. Select **Action** > **Shut Down** in Hyper-V Manager.

15. Azure only accepts fixed-size VHDs. If the VM's OS disk isn't a fixed-size VHD, use the `Convert-VHD` PowerShell cmdlet and specify the `-VHDType Fixed` option. For more information, look at the docs for `Convert-VHD` at Convert-VHD.

16. To bring a Generation 2 VM on Azure, follow these steps:

    a. Change the directory to the `boot EFI` directory:

    ```
    Bash
    cd /boot/efi/EFI
    ```

    b. Copy the `ubuntu` directory to a new directory named `boot`:

    ```
    Bash
    sudo cp -r ubuntu/ boot
    ```

    c. Change the directory to the newly created boot directory:

    ```
    Bash
    cd boot
    ```

    d. Rename the `shimx64.efi` file:

    ```
    Bash
    sudo mv shimx64.efi bootx64.efi
    ```

    e. Rename the `grub.cfg` file to `bootx64.cfg`:

    ```
    Bash
    sudo mv grub.cfg bootx64.cfg
    ```

Related content

You're now ready to use your Ubuntu Linux VHD to create new VMs in Azure. If this is the first time that you're uploading the .vhd file to Azure, see Create a Linux VM from a custom disk.

Last updated on 02/04/2026

cloud-init support for virtual machines in Azure

Article • 03/20/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article explains the support that exists for cloud-init to configure a virtual machine (VM) or Virtual Machine Scale Sets at provisioning time in Azure. These cloud-init configurations are run on first boot once the resources have been provisioned by Azure.

VM Provisioning is the process where Azure passes down your VM Create parameter values, such as hostname, username, and password, and make them available to the VM as it boots up. A 'provisioning agent' will consume those values, configure the VM, and report back when completed.

Azure supports two provisioning agents cloud-init, and the Azure Linux Agent (WALA).

cloud-init overview

cloud-init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. Because cloud-init is called during the initial boot process, there are no additional steps or required agents to apply your configuration. For more information on how to properly format your `#cloud-config` files or other inputs, see the cloud-init documentation site. `#cloud-config` files are text files encoded in base64.

cloud-init also works across distributions. For example, you don't use **apt-get install** or **yum install** to install a package. Instead you can define a list of packages to install. cloud-init automatically uses the native package management tool for the distro you select.

We're actively working with our endorsed Linux distro partners in order to have cloud-init enabled images available in the Azure Marketplace. These images will make your cloud-init deployments and configurations work seamlessly with VMs and virtual machine scale sets. Initially we collaborate with the endorsed Linux distro partners and upstream to ensure cloud-init functions with the OS on Azure, then the packages are updated and made publicly available in the distro package repositories.

There are two stages to making cloud-init available to the supported Linux distributions on Azure, package support, and then image support:

- 'cloud-init package support on Azure' documents, which cloud-init packages onwards are supported or in preview, so you can use these packages with the OS in a custom image.
- 'image cloud-init ready' documents if the image is already configured to use cloud-init.

Canonical

Expand table

Publisher / Version	Offer	SKU	Version	image cloud-init ready	cloud-init package support on Azure
Canonical 24.04	UbuntuServer	24.04-LTS	latest	yes	yes
Canonical 22.04	UbuntuServer	22.04-LTS	latest	yes	yes
Canonical 20.04	UbuntuServer	20.04-LTS	latest	yes	yes

RHEL

Expand table

Publisher / Version	Offer	SKU	Version	image cloud-init ready	cloud-init package support on Azure
RedHat 7	RHEL	7.7, 7.8, 7_9	latest	yes	yes
RedHat 8	RHEL	8.1, 8.2, 8_3, 8_4	latest	yes	yes
RedHat 9	RHEL	9_0, 9_1	latest	yes	yes

- All other RedHat SKUs starting from RHEL 7 (version 7.7) and RHEL 8 (version 8.1) including both Gen1 and Gen2 images are provisioned using cloud-init. Cloud-init isn't supported on RHEL 6.

Oracle

Expand table

Publisher / Version	Offer	SKU	Version	image cloud-init ready	cloud-init package support on Azure
Oracle 7	Oracle Linux	77, 78, ol79	latest	yes	yes
Oracle 8	Oracle Linux	81, ol82, ol83-lvm, ol84-lvm	latest	yes	yes

- All other Oracle SKUs starting from Oracle 7 (version 7.7) and Oracle 8 (version 8.1) including both Gen1 and Gen2 images are provisioned using cloud-init.

SUSE SLES

Expand table

Publisher / Version	Offer	SKU	Version	image cloud-init ready	cloud-init package support on Azure
SUSE 15	SLES (SUSE Linux Enterprise Server)	sp1, sp2, sp3	latest	yes	yes
SUSE 12	SLES (SUSE Linux Enterprise Server)	sp5	latest	yes	yes

Debian

Expand table

Publisher / Version	Offer	SKU	Version	image cloud-init ready	cloud-init package support on Azure
debian-10	Debian	10-cloudinit-gen2	Debian:debian-10:10-cloudinit-gen2:0.0.1015	yes	yes
debian-10	Debian	10-cloudinit-	Debian:debian-10:10-cloudinit-gen2:0.0.991	yes	yes

Publisher / Version	Offer	SKU	Version	image cloud-init ready	cloud-init package support on Azure
		gen2			
debian-10	Debian	10-cloudinit-gen2	Debian:debian-10:10-cloudinit-gen2:0.0.999	yes	yes

Currently, Azure Stack supports the provisioning of cloud-init enabled images.

What is the difference between cloud-init and the Linux Agent (WALA)?

WALA is an Azure platform-specific agent used to provision and configure VMs, and handle Azure extensions.

We're enhancing the task of configuring VMs to use cloud-init instead of the Linux Agent in order to allow existing cloud-init customers to use their current cloud-init scripts, or new customers to take advantage of the rich cloud-init configuration functionality. If you have existing investments in cloud-init scripts for configuring Linux systems, there are **no additional settings required** to enable cloud-init process them.

cloud-init can't process Azure extensions, so WALA is still required in the image to process extensions but needs to have its provisioning code disabled. For endorsed Linux distros images that are being converted to provision by cloud-init, they have WALA installed, and setup correctly.

When creating a VM, if you don't include the Azure CLI `--custom-data` switch at provisioning time, cloud-init or WALA takes the minimal VM provisioning parameters required to provision the VM and complete the deployment with the defaults. If you reference the cloud-init configuration with the `--custom-data` switch, whatever is contained in your custom data will be available to cloud-init when the VM boots.

cloud-init configurations applied to VMs don't have time constraints and won't cause a deployment to fail by timing out. This isn't true for WALA, if you change the WALA defaults to process custom-data, it can't exceed the total VM provisioning time allowance of 40 minutes, if so, the VM Create will fail.

cloud-init VM provisioning without a UDF driver

Beginning with cloud-init 21.2, you can use cloud-init to provision a VM in Azure without a UDF driver. If a UDF driver isn't available in the image, cloud-init uses the metadata that's available in the Azure Instance Metadata Service to provision the VM. This option works only for SSH key and user data. To pass in a password or custom data to a VM during provisioning, you must use a UDF driver.

Deploying a cloud-init enabled Virtual Machine

Deploying a cloud-init enabled virtual machine is as simple as referencing a cloud-init enabled distribution during deployment. Linux distribution maintainers have to choose to enable and integrate cloud-init into their base Azure published images. Once you confirm the image you want to deploy is cloud-init enabled, you can use the Azure CLI to deploy the image.

The first step in deploying this image is to create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

The next step is to create a file in your current shell, named *cloud-init.txt* and paste the following configuration. For this example, create the file in the Cloud Shell not on your local machine. You can use any editor of your choice. Enter sensible-editor cloud-init.txt to create the file and see a list of available editors. Use the editor of your choice. Some typical choices are nano, vim, or ed. Make sure that the whole cloud-init file is copied correctly, especially the first line:

Expand table

SLES	Ubuntu	RHEL
`# cloud-config` `package_upgrade: true`	`# cloud-config` `package_upgrade: true`	`# cloud-config` `package_upgrade: true`

SLES	Ubuntu	RHEL
`packages:` `- apache2`	`packages:` `- httpd`	`packages:` `- httpd`

> **Note**
>
> cloud-init has multiple input types, cloud-init will use first line of the customData/userData to indicate how it should process the input, for example `#cloud-config` indicates that the content should be processed as a cloud-init config.

Exit the file and save the file according to the editor. Verify file name on exit.

The final step is to create a VM with the az vm create command.

The following example creates a VM named `ubuntu2204` and creates SSH keys if they don't already exist in a default key location. To use a specific set of keys, use the `--ssh-key-value` option. Use the `--custom-data` parameter to pass in your cloud-init config file. Provide the full path to the *cloud-init.txt* config if you saved the file outside of your present working directory.

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name ubuntu2204 \
  --image Canonical:UbuntuServer:22_04-lts:latest \
  --custom-data cloud-init.txt \
  --generate-ssh-keys
```

When the VM is created, the Azure CLI shows information specific to your deployment. Take note of the `publicIpAddress`. This address is used to access the VM. It takes some time for the VM to be created, the packages to install, and the app to start. There are background tasks that continue to run after the Azure CLI returns you to the prompt. You can SSH into the VM and use the steps outlined in the Troubleshooting section to view the cloud-init logs.

You can also deploy a cloud-init enabled VM by passing the parameters in ARM template.

Troubleshooting cloud-init

Once the VM has been provisioned, cloud-init runs through all the modules and script defined in `--custom-data` in order to configure the VM. If you need to troubleshoot any errors or omissions from the configuration, you need to search for the module name (`disk_setup` or `runcmd` for example) in the cloud-init log - located in **/var/log/cloud-init.log**.

> **Note**
>
> Not every module failure results in a fatal cloud-init overall configuration failure. For example, using the `runcmd` module, if the script fails, cloud-init will still report provisioning succeeded because the runcmd module executed.

For more information of cloud-init logging, see the cloud-init documentation

Telemetry

cloud-init collects usage data and sends it to Microsoft to help improve our products and services. Telemetry is only collected during the provisioning process (first boot of the VM). The data collected helps us investigate provisioning failures and monitor performance and reliability. Data collected doesn't include any identifiers (personal identifiers). Read our privacy statement to learn more. Some examples of telemetry being collected are (this isn't an exhaustive list): OS-related information (cloud-init version, distro version, kernel version), performance metrics of essential VM provisioning actions (time to obtain DHCP lease, time to retrieve metadata necessary to configure the VM, etc.), cloud-init log, and dmesg log.

Telemetry collection is currently enabled for most of our marketplace images that use cloud-init. It's enabled by specifying KVP telemetry reporter for cloud-init. In most Azure Marketplace images, this configuration can be found in the file /etc/cloud/cloud.cfg.d/10-azure-kvp.cfg. Removing this file during image preparation disables telemetry collection for any VM created from this image.

Sample content of 10-azure-kvp.cfg

```
reporting:
  logging:
    type: log
  telemetry:
    type: hyperv
```

Next steps

Troubleshoot issues with cloud-init.

For cloud-init examples of configuration changes, see the following documents:

- Add an additional Linux user to a VM
- Run a package manager to update existing packages on first boot
- Change VM local hostname
- Install an application package, update configuration files and inject keys

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Diving deeper into cloud-init

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

To learn more about cloud-init or troubleshoot it at a deeper level, you need to understand how it works. This document highlights the important parts, and explains the Azure specifics.

When cloud-init is included in a generalized image and a VM is created from that image, it processes configurations and runs through five stages during the initial boot. These stages show you at what point cloud-init applies its configurations.

Understand Cloud-Init configuration

Configuring a VM to run on a platform uses cloud-init to apply multiple configurations. The main configurations you interact with is `User data` (customData), which supports multiple formats. For more information, see User-Data Formats & cloud-init 21.2 documentation. You also have the ability to add and run scripts (/var/lib/cloud/scripts) for other configuration.

Preconfigured Azure Marketplace images

Some configurations are already baked into Azure Marketplace images that come with cloud-init.

- **Cloud data source** - cloud-init contains code that can interact with cloud platforms, these codes are called 'datasources'. When a VM is created from a cloud-init image in Azure, cloud-init loads the Azure datasource, which interacts with the Azure metadata endpoints to get the VM specific configuration.
- **Runtime config** (/run/cloud-init).
- **Image config** (/etc/cloud), like `/etc/cloud/cloud.cfg`, `/etc/cloud/cloud.cfg.d/*.cfg`. An example of where this configuration is used in Azure, it's common for the Linux OS images with cloud-init to have an Azure datasource directive that tells cloud-init what datasource it should use, this configuration saves cloud-init time:

  ```
  Bash
  ```

```
sudo cat /etc/cloud/cloud.cfg.d/90_dpkg.cfg
```

Output

```
# to update this file, run dpkg-reconfigure cloud-init
datasource_list: [ Azure ]
```

Cloud-init boot stages (processing configuration)

When you're provisioning VMs with cloud-init, there are five configuration boot stages. The output from these stages is visible in the logs.

1. Generator Stage: The cloud-init systemd generator starts, and determines that cloud-init should be included in the boot goals, and if so, it enables cloud-init.

2. Cloud-init Local Stage: Here, cloud-init looks for the local "Azure" datasource, which enables cloud-init to interface with Azure, and apply a networking configuration, including fallback.

3. Cloud-init init Stage (Network): Networking should be online, and the NIC and route table information should be generated. At this stage, the modules listed in `cloud_init_modules` in `/etc/cloud/cloud.cfg` are run. The VM in Azure is mounted, the ephemeral disk is formatted, the hostname is set, along with other tasks.

   The following are some of the `cloud_init_modules`:

   config

   ```
   - migrator
   - seed_random
   - bootcmd
   - write-files
   - growpart
   - resizefs
   - disk_setup
   - mounts
   - set_hostname
   - update_hostname
   - ssh
   ```

   After this stage, cloud-init sends a signal to the Azure platform that the VM has been provisioned successfully. Some modules may have failed, however not all

module failures automatically result in a provisioning failure.

4. Cloud-init Config Stage : At this stage, the modules in `cloud_config_modules` defined and listed in `/etc/cloud/cloud`.cfg runs.

5. Cloud-init Final Stage : At this final stage, the modules in `cloud_final_modules`, listed in `/etc/cloud/cloud.cfg`, runs. Here modules that need to be run late in the boot process run, such as package installations and run scripts etc.

    - During this stage, you can run scripts by placing them in the directories under `/var/lib/cloud/scripts`:
        - `per-boot` - scripts within this directory, run on every reboot
        - `per-instance` - scripts within this directory run when a new instance is first booted
        - `per-once` - scripts within this directory run only once

Next steps

Troubleshooting cloud-init.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshooting VM provisioning with cloud-init

Applies to: ✔ Linux VMs ✔ Flexible scale sets

If you create generalized custom images and use cloud-init for provisioning, the VM might not build correctly. In that case, troubleshoot the image to find the issue.

Some examples, of issues with provisioning:

- The Compute Resource Provider API returns an error, and cloud-init reports the resulting failure.
- VM gets stuck at 'creating' for 40 minutes, and the VM creation is marked as failed.
- Custom data or User data doesn't get processed.
- The ephemeral disk fails to mount (for VM skus that come with SCSI resource disks).
- Users don't get created, or there are user access issues.
- Networking isn't set up correctly.
- Swap file or partition failures.

This article steps you through how to troubleshoot cloud-init. For more in-depth details, see cloud-init deep dive.

> **⊗ Caution**
>
> This article references CentOS, a Linux distribution that is End Of Support (EOS) status. Consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Troubleshooting failures reported by cloud-init and logged as error

Cloud-init emits structured errors when reporting failure to Azure during provisioning. These error messages include a reason and supporting data (such as timestamp, VM identifier, documentation URL, etc.) to help investigate the failure.

Expand table

Reason	Description	Action
failure to find DHCP interface	No network interface was found.	Delete and re-create VM. If issue persists, ensure networking drivers or Azure-specific kernel is

Reason	Description	Action
		installed and check boot diagnostics to verify eth0 is enumerated.
failure to obtain DHCP lease	DHCP service fails to respond due to transient platform issue.	Redeploy the VM (delete + recreate) due to a transient Azure DHCP platform issue.
failure to find primary DHCP interface	Primary DHCP interface wasn't found.	Check boot diagnostics to ensure primary network interface is named `eth0` and it isn't renamed.
connection timeout querying IMDS	Connections to IMDS may timeout due to transient platform issue, NSG, or OS firewall configuration.	Delete and re-create VM. If issue persists, validate that NSG or OS firewall isn't preventing access to IMDS.
read timeout querying IMDS	Connections to IMDS may timeout due to transient platform issue or OS firewall configuration.	Delete and re-create VM. If issue persists, validate OS firewall isn't preventing access to IMDS.
unexpected metadata parsing ovf-env.xml	Malformed VM metadata in `ovf-env.xml`.	Submit the issue to the cloud-init tracker.
error waiting for host shutdown	Failure during host shutdown handling.	Submit the issue to the cloud-init tracker.
Azure-proxy-agent not found	The `azure-proxy-agent` binary is missing.	Ensure Azure proxy agent is installed in the image. For more troubleshooting, check out MSP troubleshooting guide.
Azure-proxy-agent status failure	Proxy agent reported a status error.	Review proxy agent logs and update if needed. For more troubleshooting, check out MSP troubleshooting guide.
unhandled exception	An unexpected error occurred inside cloud-init.	Submit the issue to the cloud-init tracker.

For help with enabling and checking boot diagnostics, see Boot Diagnostics.

If any of these issues persist on subsequent attempts at provisioning, it's due to a misconfiguration in the image. If there's reason to believe there's a cloud-init issue, report it to cloud-init GitHub issue tracker.

Troubleshooting other failures unreported by cloud-init

Depending on the failure, consider these steps.

Step 1: Test the deployment without `customData`

Cloud-init can accept `customData` that is passed to it, when the VM is created. First, you should ensure this configuration isn't causing any issues with deployments. Some common issues can include:

- The YAML is malformed
- Scripts inside customData fail or hang
- Override something cloud-init relies on (for example, disk setup or networking)
- The data contains unsupported characters or encoding problems Try to provisioning the VM without passing in any configuration. If the VM fails to provision, follow the recommended troubleshooting steps. If the configuration isn't applied, refer to step 4.

Step 2: Review image requirements

The primary cause of VM provisioning failure is the OS image doesn't satisfy the prerequisites for running on Azure. Make sure your images are properly prepared before attempting to provision them in Azure.

The following articles illustrate the steps to prepare various linux distributions that are supported in Azure:

- CentOS-based Distributions
- Debian Linux
- Flatcar Container Linux
- Oracle Linux
- Red Hat Enterprise Linux
- SLES & openSUSE
- Ubuntu
- Others: Non-Endorsed Distributions

For the supported Azure cloud-init images, the Linux distributions already have all the required packages and configurations in place to correctly provision the image in Azure. If you find your VM is failing to create from your own curated image, try a supported Azure Marketplace image that already is configured for cloud-init, with your optional `customData`. If the `customData` works correctly with an Azure Marketplace image, then there's probably an issue with your curated image.

Step 3: Collect & review VM logs

When the VM fails to provision, Azure shows 'creating' status, for 20 minutes, and then reboot the VM, and wait another 20 minutes before finally marking the VM deployment as failed, before finally marking it with an `OSProvisioningTimedOut` error.

While the VM is running, you need the logs from the VM to understand why provisioning failed. To understand why VM provisioning failed, don't stop the VM. Keep the VM running. You need to keep the failed VM in a running state in order to collect logs. To collect the logs, use one of the following methods:

- Enable Boot Diagnostics before creating the VM and then View them during the boot.
- Serial Console
- Run AZ VM Repair to attach and mount the OS disk (*lvm*, *no lvm*), which will allow you to collect and examine these logs:

```bash
/rescue/var/log/waagent*
/rescue/var/log/syslog*
/rescue/var/log/rsyslog*
/rescue/var/log/messages*
/rescue/var/log/kern*
/rescue/var/log/dmesg*
/rescue/var/log/boot*
/rescue/ /var/log/cloud-init.log
/rescue//var/log/cloud-init-output.log
```

> **Note**
>
> Alternatively, you can create a rescue VM manually by using the Azure portal. For more information, see Troubleshoot a Linux VM by attaching the OS disk to a recovery VM using the Azure portal. To start initial troubleshooting, begin with the serial logs and cloud-init logs to understand where the failure occurred. Then use the other logs for a deeper dive to help provide additional insights.

- /var/log/cloud-init.log
- /var/log/cloud-init-output.log
- Serial/boot logs

In all logs, start searching for "Failed," "WARNING," "WARN," "err," "error," and "ERROR." Setting configuration to ignore case-sensitive searches is recommended.

Alternatively, use command `cloud-init collect-logs` to collect all necessary logs. Azure's latest cloud-init versions (≥ 18.2) include the collect-logs command, which:

Gathers essential logs: /var/log/cloud-init*.log, instance metadata, system info.

Packages everything into a timestamped .tar.gz archive.

Saves the archive locally (for example, `/tmp/cloud-init-logs-timestamp.tar.gz`).

> **Tip**
>
> If you're troubleshooting a custom image, you should consider adding a user during the image. If the provisioning fails to set the admin user, you can still log in to the OS.

Analyzing the logs

Here are more details about what to look for in each cloud-init log.

/var/log/cloud-init.log

By default, all cloud-init events with a priority of debug or higher, are written to `/var/log/cloud-init.log`. This log provides verbose logs of every event that occurred during cloud-init initialization.

For example:

Console

```
2019-10-10 04:51:25,321 - util.py[DEBUG]: Failed mount of '/dev/sr0' as 'auto':
Unexpected error while running command.
Command: ['mount', '-o', 'ro,sync', '-t', 'auto', u'/dev/sr0', '/run/cloud-
init/tmp/tmpLIrklc']
Exit code: 32
Reason: -
Stdout:
Stderr: mount: unknown filesystem type 'udf'
2020-01-31 00:21:53,352 - DataSourceAzure.py[WARNING]: /dev/sr0 was not mountable
```

When you find an error or warning, read backwards in the cloud-init log to understand what cloud-init was attempting before it hit the error or warning. In many cases, cloud-init runs OS commands or performs provisioning steps before the error occurs. These actions can help explain why the error appears in the logs. The following example shows that cloud-init attempted to mount a device right before it encountered the issue.

Output

```
2019-10-10 04:51:24,010 - util.py[DEBUG]: Running command ['mount', '-o', 'ro,sync',
'-t', 'auto', u'/dev/sr0', '/run/cloud-init/tmp/tmpXXXXX'] with allowed return codes
[0] (shell=False, capture=True)
```

If you have access to the Serial Console, you can try to rerun the command that cloud-init was trying to run.

The logging for `/var/log/cloud-init.log` can also be reconfigured within /etc/cloud/cloud.cfg.d/05_logging.cfg. For more information of cloud-init logging, see the cloud-init documentation.

/var/log/cloud-init-output.log

You can get information from the `stdout` and `stderr` during the stages of cloud-init. This data normally involves routing table information, networking information, ssh host key verification information, `stdout,` and `stderr` for each stage of cloud-init, along with the timestamps. If desired, `stderr` and `stdout` logging can be reconfigured from `/etc/cloud/cloud.cfg.d/05_logging.cfg`.

Serial/boot logs

Cloud-init has multiple dependencies. These dependencies are documented in the required prerequisites for images on Azure, such as networking, storage, the ability to mount an ISO, and to mount and format the temporary disk. Any of these dependencies may throw errors and cause cloud-init to fail. For example, If the VM can't get a DHCP lease, cloud-init fails.

If you still can't isolate why cloud-init failed to provision then you need to understand what cloud-init stages, and when modules run. For more information, see Diving deeper into cloud-init.

Step 4: Investigate why the configuration isn't being applied

Not every failure in cloud-init results in a fatal provisioning failure. For example, if you use the `runcmd` module in a cloud-init config, a nonzero exit code from the command causes the VM provisioning to fail. This behavior occurs because the module runs after the core provisioning steps in the first three stages of cloud-init. To troubleshoot why the configuration didn't apply, review the logs in Step 3, and cloud-init modules manually. For example:

- `runcmd` - do the scripts run without errors? To ensure they run as expected, run the configuration manually from the terminal.
- Installing packages - does the VM have access to package repositories?

- Check the `customData` configuration that was provided to the VM. This file is located in `/var/lib/cloud/instances/<unique-instance-identifier>/user-data.txt`.

Next steps

If cloud-init skips the configuration, examine each cloud-init stage and the timing of module execution to identify the cause. For more information, see Diving deeper into cloud-init configuration.

Last updated on 01/19/2026

Use cloud-init to set hostname for a Linux VM in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to use cloud-init to configure a specific hostname on a virtual machine (VM) or virtual machine scale sets (VMSS) at provisioning time in Azure. These cloud-init scripts run on first boot once the resources have been provisioned by Azure. For more information about how cloud-init works natively in Azure and the supported Linux distros, see cloud-init overview

Set the hostname with cloud-init

By default, the hostname is the same as the VM name when you create a new virtual machine in Azure. To run a cloud-init script to change this default hostname when you create a VM in Azure with az vm create, specify the cloud-init file with the `--custom-data` switch.

To see upgrade process in action, create a file in your current shell named *cloud_init_hostname.txt* and paste the following configuration. For this example, create the file in the Cloud Shell not on your local machine. You can use any editor you wish. Make sure that the whole cloud-init file is copied correctly, especially the first line.

YAML

```yaml
#cloud-config
fqdn: myhostname
```

Before deploying this image, you need to create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLI

```azurecli
az group create --name myResourceGroup --location eastus
```

Now, create a VM with az vm create and specify the cloud-init file with `--custom-data cloud_init_hostname.txt` as follows:

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name vmName \
  --image imageCIURN \
  --custom-data cloud_init_hostname.txt \
  --generate-ssh-keys
```

> **Note**
>
> Replace **myResourceGroup**, **vmName**, and **imageCIURN** values accordingly. Make sure an image with Cloud-init is chosen.

Once created, the Azure CLI shows information about the VM. Use the `publicIpAddress` to SSH to your VM. Enter your own address as follows:

Bash

```
ssh <user>@<publicIpAddress>
```

To see the VM name, use the `hostname` command as follows:

Bash

```
sudo hostname
```

The VM should report the hostname as that value set in the cloud-init file, as shown in the following example output:

Output

```
myhostname
```

Next steps

For additional cloud-init examples of configuration changes, see the following:

- Add an additional Linux user to a VM
- Run a package manager to update existing packages on first boot
- Change VM local hostname
- Install an application package, update configuration files and inject keys

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use cloud-init to update and install packages in a Linux VM in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to use cloud-init to update packages on a Linux virtual machine (VM) or virtual machine scale sets at provisioning time in Azure. These cloud-init scripts run on first boot once the resources have been provisioned by Azure. For more information about how cloud-init works natively in Azure and the supported Linux distros, see cloud-init overview

Update a VM with cloud-init

For security purposes, you may want to configure a VM to apply the latest updates on first boot. As cloud-init works across different Linux distros, there's no need to specify `apt`, `zypper` or `yum` for the package manager. Instead, you define `package_upgrade` and let the cloud-init process determine the appropriate mechanism for the distro in use.

For this example, we use the Azure Cloud Shell. To see the upgrade process in action, create a file named *cloud_init_upgrade.txt* and paste the following configuration. You can use any editor you wish. Make sure that the whole cloud-init file is copied correctly, especially the first line.

Copy the text below and paste it into the `cloud_init_upgrade.txt` file. Make sure that the whole cloud-init file is copied correctly, especially the first line.

YAML

```yaml
#cloud-config
package_upgrade: true
packages:
- httpd
```

Before deploying, you need to create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Now, create a VM with az vm create and specify the cloud-init file with the `--custom-data` parameter as follows:

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name vmName \
  --image imageCIURN \
  --custom-data cloud_init_upgrade.txt \
  --admin-username azureuser \
  --generate-ssh-keys
```

> **Note**
>
> Replace **myResourceGroup**, **vmName**, and **imageCIURN** values accordingly. Make sure an image with Cloud-init is chosen.

SSH to the public IP address of your VM shown in the output from the preceding command. Enter your own **user** and **publicIpAddress** as follows:

Bash

```
ssh <user>@<publicIpAddress>
```

Run the package management tool and check for updates:

RHEL/Oracle Linux

- Execute the following command to confirm there are no pending updates

Bash

```
sudo yum check-update
```

As cloud-init checked for and installed updates on boot, there should be no additional updates to apply.

- You can see the update process, number of altered packages, and the installation of `httpd` by running the following command and review the

output.

Bash

```
sudo yum history
```

Output

```
ID     | Command line                                  | Date and time
| Action(s)      | Altered
-------------------------------------------------------------------------------
--------------------------
     3 | -y install httpd                              | 2022-02-18 18:30
| Install        |    7
     2 | -y upgrade                                    | 2022-02-18 18:23
| I, O, U        |  321 EE
     1 |                                               | 2021-02-04 19:20
| Install        |  496 EE
```

Next steps

For additional cloud-init examples of configuration changes, see the following:

- Add an additional Linux user to a VM
- Run a package manager to update existing packages on first boot
- Change VM local hostname
- Install an application package, update configuration files and inject keys

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use cloud-init to add a user to a Linux VM in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to use cloud-init to add a user on a virtual machine (VM) or virtual machine scale sets (VMSS) at provisioning time in Azure. This cloud-init script runs on first boot once the resources have been provisioned by Azure. For more information about how cloud-init works natively in Azure and the supported Linux distros, see cloud-init overview.

Add a user to a VM with cloud-init

One of the first tasks on any new Linux VM is to add an additional user for yourself to avoid the use of *root*. SSH keys are best practice for security and usability. Keys are added to the *~/.ssh/authorized_keys* file with this cloud-init script.

To add a user to a Linux VM, create a file in your current shell named *cloud_init_add_user.txt* and paste the following configuration. For this example, create the file in the Cloud Shell not on your local machine. You can use any editor you wish. Make sure that the whole cloud-init file is copied correctly, especially the first line. You need to provide your own public key (such as the contents of *~/.ssh/id_rsa.pub*) for the value of `ssh-authorized-keys:` - it has been shortened here to simplify the example.

YAML

```yaml
#cloud-config
users:
  - default
  - name: myadminuser
    groups: sudo
    shell: /bin/bash
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    ssh-authorized-keys:
      - ssh-rsa AAAAB3<snip>
```

> **Note**
>
> The #cloud-config file includes the `- default` parameter included. This will append the user, to the existing admin user created during provisioning. If you create a user

without the `- default` parameter - the auto generated admin user created by the Azure platform would be overwritten.

Before deploying this image, you need to create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Now, create a VM with az vm create and specify the cloud-init file with `--custom-data cloud_init_add_user.txt` as follows:

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name vmName \
  --image imageCIURN \
  --custom-data cloud_init_add_user.txt \
  --generate-ssh-keys
```

> **Note**
>
> Replace **myResourceGroup**, **vmName**, and **imageCIURN** values accordingly. Make sure an image with Cloud-init is chosen.

SSH to the public IP address of your VM shown in the output from the preceding command. Enter your own **user** and **publicIpAddress** as follows:

Bash

```
ssh <user>@<publicIpAddress>
```

To confirm your user was added to the VM and the specified groups, view the contents of the */etc/group* file as follows:

Bash

```
sudo cat /etc/group
```

The following example output shows the user from the *cloud_init_add_user.txt* file has been added to the VM and the appropriate group:

Output

```
root:x:0:
<snip />
sudo:x:27:myadminuser
<snip />
myadminuser:x:1000:
```

Next steps

For additional cloud-init examples of configuration changes, see the following:

- Add an additional Linux user to a VM
- Run a package manager to update existing packages on first boot
- Change VM local hostname
- Install an application package, update configuration files and inject keys

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use cloud-init to configure a swap partition on a Linux VM

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to use cloud-init to configure the swap partition on various Linux distributions. The swap partition was traditionally configured by the Linux Agent (WALA) based on which distributions required one. This document outlines the process for building the swap partition on demand during provisioning time using cloud-init. For more information about how cloud-init works natively in Azure and the supported Linux distros, see cloud-init overview

Create swap partition for Ubuntu based images

By default on Azure, Ubuntu gallery images don't create swap partitions. To enable swap partition configuration during VM provisioning time using cloud-init - please see the AzureSwapPartitions document on the Ubuntu wiki.

Create swap partition for RHEL based images

Create a file in your current shell named *cloud_init_swappart.txt* and paste the following configuration. For this example, create the file in the Cloud Shell not on your local machine. You can use any editor you wish. Make sure that the whole cloud-init file is copied correctly, especially the first line.

YAML

```yaml
#cloud-config
disk_setup:
  ephemeral0:
    table_type: gpt
    layout: [66, [33,82]]
    overwrite: true
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
  - ["ephemeral0.1", "/mnt"]
```

```
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-systemd.requires=cloud-
init.service", "0", "0"]
```

The mount is created with the `nofail` option to ensure that the boot process continues even if the mount isn't completed successfully.

Before deploying this image, you need to create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Now, create a VM with az vm create and specify the cloud-init file with `--custom-data cloud_init_swappart.txt` as follows:

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name vmName \
  --image imageCIURN \
  --custom-data cloud_init_swappart.txt \
  --generate-ssh-keys
```

> **Note**
>
> Replace **myResourceGroup**, **vmName**, and **imageCIURN** values accordingly. Make sure an image with Cloud-init is chosen.

Modify an already running machine

If you already provisioned your server and wish to modify the mount point of the ephemeral storage and want to configure a part of the disk as swap space, use the following steps.

Create cloud-init configuration file named `00-azure-swap.cfg` in the `/etc/cloud/cloud.cfg.d` directory with the following content:

YAML

```
#cloud-config
disk_setup:
  ephemeral0:
    table_type: gpt
    layout: [66, [33,82]]
    overwrite: true
fs_setup:
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:
  - ["ephemeral0.1", "/mnt"]
  - ["ephemeral0.2", "none", "swap", "sw,nofail,x-systemd.requires=cloud-
init.service", "0", "0"]
```

Next, append a line to the `/etc/systemd/system.conf` file with following content:

config

```
DefaultEnvironment="CLOUD_CFG=/etc/cloud/cloud.cfg.d/00-azure-swap.cfg"
```

> ⓘ **Note**
>
> The name of the file is totally arbitrary, it can be replaced with any particular name of your preference, it just needs the .cfg suffix and make sure to reflect the changes in the CLOUD_CFG parameter line as well.

After the changes are done, the machine needs to be deallocated or redeployed for the changes to take effect.

Verify swap partition was created

SSH to the public IP address of your VM shown in the output from the preceding command. Enter your own **user** and **publicIpAddress** as follows:

Bash

```
ssh <user>@<publicIpAddress>
```

Once you have SSH'ed into the vm, check if the swap partition was created

Bash

```
sudo swapon -s
```

The output from this command should look like this:

Output

```
Filename                Type        Size    Used    Priority
/dev/sdb2  partition   2494440 0   -1
```

> **Note**
>
> If you have an existing Azure image that has a swap partition configured and you want to change the swap partition configuration for new images, you should remove the existing swap partition. Please see Customize Images to provision by cloud-init document for more details.

Next steps

For more cloud-init examples of configuration changes, see the following:

- Add an additional Linux user to a VM
- Run a package manager to update existing packages on first boot
- Change VM local hostname
- Install an application package, update configuration files and inject keys

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use cloud-init to run a bash script in a Linux VM in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to use cloud-init to run an existing bash script on a Linux virtual machine (VM) or virtual machine scale sets (VMSS) at provisioning time in Azure. These cloud-init scripts run on first boot once the resources have been provisioned by Azure. For more information about how cloud-init works natively in Azure and the supported Linux distros, see cloud-init overview

Run a bash script with cloud-init

With cloud-init you do not need to convert your existing scripts into a cloud-config, cloud-init accepts multiple input types, one of which is a bash script.

If you have been using the Linux Custom Script Azure Extension to run your scripts, you can migrate them to use cloud-init. However, Azure Extensions have integrated reporting to alert to script failures, a cloud-init image deployment will NOT fail if the script fails.

To see this functionality in action, create a simple bash script for testing. Like the cloud-init `#cloud-config` file, this script must be local to where you will be running the AzureCLI commands to provision your virtual machine. For this example, create the file in the Cloud Shell not on your local machine. You can use any editor you wish. Make sure that the whole cloud-init file is copied correctly, especially the first line.

Bash

```
#!/bin/sh
echo "this has been written via cloud-init" + $(date) >> /tmp/myScript.txt
```

Before deploying this image, you need to create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location.

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Now, create a VM with az vm create and specify the bash script file with `--custom-data simple_bash.sh` as follows:

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name vmName \
  --image imageCIURN \
  --custom-data simple_bash.sh \
  --generate-ssh-keys
```

> ⓘ Note
>
> Replace **myResourceGroup**, **vmName**, and **imageCIURN** values accordingly. Make sure an image with Cloud-init is chosen.

Verify bash script has run

SSH to the public IP address of your VM shown in the output from the preceding command. Enter your own **user** and **publicIpAddress** as follows:

Bash

```
ssh <user>@<publicIpAddress>
```

Verify that `/tmp/myScript.txt` file exists and has the appropriate text inside of it.

Bash

```
sudo cat /tmp/myScript
```

If it does not, you can check the `/var/log/cloud-init.log` for more details. Search for the following entry:

Bash

```
sudo cat /var/log/cloud-init.log
```

Output

```
Running config-scripts-user using lock Running command
['/var/lib/cloud/instance/scripts/part-001']
```

Next steps

For additional cloud-init examples of configuration changes, see the following:

- Add an additional Linux user to a VM
- Run a package manager to update existing packages on first boot
- Change VM local hostname
- Install an application package, update configuration files and inject keys

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Creating generalized images without a provisioning agent

09/02/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Microsoft Azure provides provisioning agents for Linux VMs in the form of the walinuxagent or cloud-init (recommended). But there could be a scenario when you don't want to use either of these applications for your provisioning agent, such as:

- Your Linux distro/version doesn't support cloud-init/Linux Agent.
- You require specific VM properties to be set, such as hostname.

> **Note**
>
> If you do not require any properties to be set or any form of provisioning to happen you should consider creating a specialized image.

This article shows how you can set up your VM image to satisfy the Azure platform requirements and set the hostname, without installing a provisioning agent.

Networking and reporting ready

In order to have your Linux VM communicate with Azure components, a DHCP client is required. The client is used to retrieve a host IP, DNS resolution, and route management from the virtual network. Most distros ship with these utilities out-of-the-box. Tools that are tested on Azure by Linux distro vendors include `dhclient`, `network-manager`, `systemd-networkd` and others.

> **Note**
>
> Currently creating generalized images without a provisioning agent only supports DHCP-enabled VMs.

After networking has been set up and configured, select "report ready". This tells Azure that the VM has been successfully provisioned.

> **Important**

Failing to report ready to Azure will result in your VM being rebooted!

Demo/sample

An existing Marketplace image (in this case, a Debian Buster VM) with the Linux Agent (walinuxagent) removed and a custom python script added is the easiest way to tell Azure that the VM is "ready".

Create the resource group and base VM:

Azure CLI

```
$ az group create --location eastus --name demo1
```

Create the base VM:

Azure CLI

```
$ az vm create \
    --resource-group demo1 \
    --name demo1 \
    --location eastus \
    --ssh-key-value <ssh_pub_key_path> \
    --public-ip-address-dns-name demo1 \
    --image "debian:debian-10:10:latest"
```

Remove the image provisioning Agent

Once the VM is provisioning, you can connect to it via SSH and remove the Linux Agent:

Bash

```
$ sudo apt purge -y waagent
$ sudo rm -rf /var/lib/waagent /etc/waagent.conf /var/log/waagent.log
```

Add required code to the VM

Also inside the VM, because we've removed the Azure Linux Agent we need to provide a mechanism to report ready.

Python script

Python

```python
import http.client
import sys
from xml.etree import ElementTree

wireserver_ip = '168.63.129.16'
wireserver_conn = http.client.HTTPConnection(wireserver_ip)

print('Retrieving goal state from the Wireserver')
wireserver_conn.request(
    'GET',
    '/machine?comp=goalstate',
    headers={'x-ms-version': '2012-11-30'}
)

resp = wireserver_conn.getresponse()

if resp.status != 200:
    print('Unable to connect with wireserver')
    sys.exit(1)

wireserver_goalstate = resp.read().decode('utf-8')

xml_el = ElementTree.fromstring(wireserver_goalstate)

container_id = xml_el.findtext('Container/ContainerId')
instance_id =
xml_el.findtext('Container/RoleInstanceList/RoleInstance/InstanceId')
incarnation = xml_el.findtext('Incarnation')
print(f'ContainerId: {container_id}')
print(f'InstanceId: {instance_id}')
print(f'Incarnation: {incarnation}')

# Construct the XML response we need to send to Wireserver to report ready.
health = ElementTree.Element('Health')
goalstate_incarnation = ElementTree.SubElement(health, 'GoalStateIncarnation')
goalstate_incarnation.text = incarnation
container = ElementTree.SubElement(health, 'Container')
container_id_el = ElementTree.SubElement(container, 'ContainerId')
container_id_el.text = container_id
role_instance_list = ElementTree.SubElement(container, 'RoleInstanceList')
role = ElementTree.SubElement(role_instance_list, 'Role')
instance_id_el = ElementTree.SubElement(role, 'InstanceId')
instance_id_el.text = instance_id
health_second = ElementTree.SubElement(role, 'Health')
state = ElementTree.SubElement(health_second, 'State')
state.text = 'Ready'

out_xml = ElementTree.tostring(
    health,
    encoding='unicode',
    method='xml'
)
```

```
print('Sending the following data to Wireserver:')
print(out_xml)

wireserver_conn.request(
    'POST',
    '/machine?comp=health',
    headers={
        'x-ms-version': '2012-11-30',
        'Content-Type': 'text/xml;charset=utf-8',
        'x-ms-agent-name': 'custom-provisioning'
    },
    body=out_xml
)

resp = wireserver_conn.getresponse()
print(f'Response: {resp.status} {resp.reason}')

wireserver_conn.close()
```

Bash script

```
#!/bin/bash

attempts=1
until [ "$attempts" -gt 5 ]
do
    echo "obtaining goal state - attempt $attempts"
    goalstate=$(curl --fail -v -X 'GET' -H "x-ms-agent-name: azure-vm-register" \
                                        -H "Content-Type: text/xml;charset=utf-8"
\
                                        -H "x-ms-version: 2012-11-30" \
                                           "http://168.63.129.16/machine/?
comp=goalstate")
    if [ $? -eq 0 ]
    then
       echo "successfully retrieved goal state"
       retrieved_goal_state=true
       break
    fi
    sleep 5
    attempts=$((attempts+1))
done

if [ "$retrieved_goal_state" != "true" ]
then
    echo "failed to obtain goal state - cannot register this VM"
    exit 1
fi

container_id=$(grep ContainerId <<< "$goalstate" | sed 's/\s*<\/*ContainerId>//g'
```

```
| sed 's/\r$//')
instance_id=$(grep InstanceId <<< "$goalstate" | sed 's/\s*<\/*InstanceId>//g' |
sed 's/\r$//')

ready_doc=$(cat << EOF
<?xml version="1.0" encoding="utf-8"?>
<Health xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <GoalStateIncarnation>1</GoalStateIncarnation>
  <Container>
    <ContainerId>$container_id</ContainerId>
    <RoleInstanceList>
      <Role>
        <InstanceId>$instance_id</InstanceId>
        <Health>
          <State>Ready</State>
        </Health>
      </Role>
    </RoleInstanceList>
  </Container>
</Health>
EOF
)

attempts=1
until [ "$attempts" -gt 5 ]
do
    echo "registering with Azure - attempt $attempts"
    curl --fail -v -X 'POST' -H "x-ms-agent-name: azure-vm-register" \
                             -H "Content-Type: text/xml;charset=utf-8" \
                             -H "x-ms-version: 2012-11-30" \
                             -d "$ready_doc" \
                             "http://168.63.129.16/machine?comp=health"
    if [ $? -eq 0 ]
    then
       echo "successfully register with Azure"
       break
    fi
    sleep 5 # sleep to prevent throttling from wire server
done
```

Generic steps (if not using Python or Bash)

If your VM doesn't have Python installed or available, you can programmatically reproduce this above script logic with the following steps:

1. Retrieve the `ContainerId`, `InstanceId`, and `Incarnation` by parsing the response from the WireServer: `curl -X GET -H 'x-ms-version: 2012-11-30' http://168.63.129.16/machine?comp=goalstate`.

2. Construct the following XML data, injecting the parsed `ContainerId`, `InstanceId`, and `Incarnation` from the above step:

```xml
<Health>
  <GoalStateIncarnation>INCARNATION</GoalStateIncarnation>
  <Container>
    <ContainerId>CONTAINER_ID</ContainerId>
    <RoleInstanceList>
      <Role>
        <InstanceId>INSTANCE_ID</InstanceId>
        <Health>
          <State>Ready</State>
        </Health>
      </Role>
    </RoleInstanceList>
  </Container>
</Health>
```

3. Post this data to WireServer: `curl -X POST -H 'x-ms-version: 2012-11-30' -H "x-ms-agent-name: WALinuxAgent" -H "Content-Type: text/xml;charset=utf-8" -d "$REPORT_READY_XML" http://168.63.129.16/machine?comp=health`

Automating running the code at first boot

This demo uses systemd, which is the most common init system in modern Linux distros. So the easiest and most native way to ensure this report ready mechanism runs at the right time is to create a systemd service unit. You can add the following unit file to `/etc/systemd/system` (this example names the unit file `azure-provisioning.service`):

```bash
[Unit]
Description=Azure Provisioning

[Service]
Type=oneshot
ExecStart=/usr/bin/python3 /usr/local/azure-provisioning.py
ExecStart=/bin/bash -c "hostnamectl set-hostname $(curl \
    -H 'metadata: true' \
    'http://169.254.169.254/metadata/instance/compute/name?api-version=2019-06-01&format=text')"
ExecStart=/usr/bin/systemctl disable azure-provisioning.service

[Install]
WantedBy=multi-user.target
```

This systemd service does three things for basic provisioning:

1. Reports ready to Azure (to indicate that it came up successfully).
2. Renames the VM based off of the user-supplied VM name by pulling this data from Azure Instance Metadata Service (IMDS). **Note** IMDS also provides other instance metadata, such as SSH Public Keys, so you can set more than the hostname.
3. Disables itself so that it only runs on first boot and not on subsequent reboots.

With the unit on the filesystem, run the following to enable it:

Bash

```
$ sudo systemctl enable azure-provisioning.service
```

Now the VM is ready to be generalized and have an image created from it.

Completing the preparation of the image

Back on your development machine, run the following to prepare for image creation from the base VM:

Azure CLI

```
$ az vm deallocate --resource-group demo1 --name demo1
$ az vm generalize --resource-group demo1 --name demo1
```

And create the image from this VM:

Azure CLI

```
$ az image create \
    --resource-group demo1 \
    --source demo1 \
    --location eastus \
    --name demo1img
```

Now we're ready to create a new VM from the image. This can also be used to create multiple VMs:

Azure CLI

```
$ IMAGE_ID=$(az image show -g demo1 -n demo1img --query id -o tsv)
$ az vm create \
    --resource-group demo12 \
    --name demo12 \
```

```
    --location eastus \
    --ssh-key-value <ssh_pub_key_path> \
    --public-ip-address-dns-name demo12 \
    --image "$IMAGE_ID"
    --enable-agent false
```

> **Note**
>
> It is important to set `--enable-agent` to `false` because walinuxagent doesn't exist on this VM that is going to be created from the image.

The VM should be provisioned successfully. After Logging into the newly provisioning VM, you should be able to see the output of the report ready systemd service:

Bash

```
$ sudo journalctl -u azure-provisioning.service
-- Logs begin at Thu 2020-06-11 20:28:45 UTC, end at Thu 2020-06-11 20:31:24 UTC.
--
Jun 11 20:28:49 thstringnopa systemd[1]: Starting Azure Provisioning...
Jun 11 20:28:54 thstringnopa python3[320]: Retrieving goal state from the
Wireserver
Jun 11 20:28:54 thstringnopa python3[320]: ContainerId: 7b324f53-983a-43bc-b919-
1775d6077608
Jun 11 20:28:54 thstringnopa python3[320]: InstanceId: fbb84507-46cd-4f4e-bd78-
a2edaa9d059b._thstringnopa2
Jun 11 20:28:54 thstringnopa python3[320]: Sending the following data to
Wireserver:
Jun 11 20:28:54 thstringnopa python3[320]: <Health>
<GoalStateIncarnation>1</GoalStateIncarnation><Container><ContainerId>7b324f53-
983a-43bc-b919-1775d6077608</ContainerId><RoleInstanceList><Role>
<InstanceId>fbb84507-46cd-4f4e-bd78-a2edaa9d059b._thstringnopa2</InstanceId>
<Health><State>Ready</State></Health></Role></RoleInstanceList></Container>
</Health>
Jun 11 20:28:54 thstringnopa python3[320]: Response: 200 OK
Jun 11 20:28:56 thstringnopa bash[472]:   % Total    % Received % Xferd  Average
Speed   Time    Time     Time  Current
Jun 11 20:28:56 thstringnopa bash[472]:                                  Dload
Upload   Total   Spent    Left  Speed
Jun 11 20:28:56 thstringnopa bash[472]: [158B blob data]
Jun 11 20:28:56 thstringnopa2 systemctl[475]: Removed /etc/systemd/system/multi-
user.target.wants/azure-provisioning.service.
Jun 11 20:28:56 thstringnopa2 systemd[1]: azure-provisioning.service: Succeeded.
Jun 11 20:28:56 thstringnopa2 systemd[1]: Started Azure Provisioning.
```

Support

If you implement your own provisioning code/agent, then you own the support of this code, Microsoft support will only investigate issues relating to the provisioning interfaces not being available. We're continually making improvements and changes in this area, so you must monitor for changes in cloud-init and Azure Linux Agent for provisioning API changes.

Next steps

For more information, see Linux provisioning.

Disable or remove the Linux Agent from VMs and images

09/17/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Before removing the Linux Agent, you must understand of what VM will not be able to do after the Linux Agent is removed.

Azure virtual machine (VM) extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs, extensions are installed and managed by the Azure control plane. It is the job of the Azure Linux Agent to process the platform extension commands and ensure the correct state of the extension inside the VM.

The Azure platform hosts many extensions that range from VM configuration, monitoring, security, and utility applications. There is a large choice of first and third-party extensions, examples of key scenarios that extensions are used for:

- Supporting first party Azure services, such as Azure Backup, Monitoring, Disk Encryption, Security, Site Replication and others.
- SSH / Password resets
- VM configuration - Running custom scripts, installing Chef, Puppet agents etc..
- Third-party products, such as AV products, VM vulnerability tools, VM and App monitoring tooling.
- Extensions can be bundled with a new VM deployment. For example, they can be part of a larger deployment, configuring applications on VM provision, or run against any supported extension operated systems post deployment.

Disabling extension processing

There are several ways to disable extension processing, depending on your needs, but before you continue, you **MUST** remove all extensions deployed to the VM, for example using the Azure CLI, you can list and delete:

Azure CLI

```
az vm extension delete -g MyResourceGroup --vm-name MyVm -n extension_name
```

> ⓘ Note

If you do not do the above, the platform will try to send the extension configuration and timeout after 40min.

Disable at the control plane

If you are not sure whether you will need extensions in the future, you can leave the Linux Agent installed on the VM, then disable extension processing capability from the platform. This is option is available in `Microsoft.Compute` api version `2018-06-01` or higher, and does not have a dependency on the Linux Agent version installed.

Azure CLI

```
az vm update -g <resourceGroup> -n <vmName> --set
osProfile.allowExtensionOperations=false
```

You can easily reenable this extension processing from the platform, with the above command, but set it to 'true'.

Remove the Linux Agent from a running VM

Ensure you have **removed** all existing extensions from the VM before, as per above.

Step 1: Remove the Azure Linux Agent

If you just remove the Linux Agent, and not the associated configuration artifacts, you can reinstall at a later date. Run one of the following, as root, to remove the Azure Linux Agent:

Ubuntu

Bash

```
#For Ubuntu 18.04+:
sudo apt -y remove walinuxagent
```

Step 2: (Optional) Remove the Azure Linux Agent artifacts

ⓘ **Important**

You can remove all associated artifacts of the Linux Agent, but this will mean you cannot reinstall it at a later date. Therefore, it is strongly recommended you consider disabling the Linux Agent first, removing the Linux Agent using the above only.

If you know you will not ever reinstall the Linux Agent again, then you can run the following:

Ubuntu

Bash

```
#For Ubuntu 18.04+:
sudo apt -y purge walinuxagent
sudo cp -rp /var/lib/waagent /var/lib/waagent.bkp
sudo rm -rf /var/lib/waagent
sudo rm -f /var/log/waagent.log
```

Preparing an image without the Linux Agent

If you have an image that already contains cloud-init, and you want to remove the Linux agent, but still provision using cloud-init, run the steps in Step 2 (and optionally Step 3) as root to remove the Azure Linux Agent and then the following will remove the cloud-init configuration and cached data, and prepare the VM to create a custom image.

Bash

```
sudo cloud-init clean --logs --seed
```

Deprovision and create an image

The Linux Agent has the ability to clean up some of the existing image metadata, with the step "waagent -deprovision+user", however, after it has been removed, you will need to perform actions such as the below, and remove any other sensitive data from it.

- Remove all existing ssh host keys

  Bash

  ```
  sudo rm /etc/ssh/ssh_host_*key*
  ```

- Delete the admin account

Bash

```
sudo touch /var/run/utmp
sudo userdel -f -r <admin_user_account>
```

- Delete the root password

Bash

```
sudo passwd -d root
```

Once you have completed the above, you can create the custom image using the Azure CLI.

Create a regular managed image

Azure CLI

```
az vm deallocate -g <resource_group> -n <vm_name>
az vm generalize -g <resource_group> -n <vm_name>
az image create -g <resource_group> -n <image_name> --source <vm_name>
```

Create an image version in a Azure Compute Gallery

Azure CLI

```
az sig image-version create \
    -g $sigResourceGroup
    --gallery-name $sigName
    --gallery-image-definition $imageDefName
    --gallery-image-version 1.0.0
    --managed-image /subscriptions/00000000-0000-0000-0000-
00000000xxxx/resourceGroups/imageGroups/providers/images/MyManagedImage
```

Creating a VM from an image that does not contain a Linux Agent

When you create the VM from the image with no Linux Agent, you need to ensure the VM deployment configuration indicates extensions are not supported on this VM.

> ⓘ **Note**

If you do not do the above, the platform will try to send the extension configuration and timeout after 40min.

To deploy the VM with extensions disabled, you can use the Azure CLI with --enable-agent.

Azure CLI

```
az vm create \
    --resource-group $resourceGroup \
    --name $prodVmName \
    --image RedHat:RHEL:8.1-ci:latest \
    --admin-username azadmin \
    --ssh-key-value "$sshPubkeyPath" \
    --enable-agent false
```

Alternatively, you can do this using Azure Resource Manager (ARM) templates, by setting `"provisionVMAgent": false,`.

JSON

```
"osProfile": {
    "computerName": "[parameters('virtualMachineName')]",
    "adminUsername": "[parameters('adminUsername')]",
    "linuxConfiguration": {
        "disablePasswordAuthentication": "true",
        "provisionVMAgent": false,
        "ssh": {
            "publicKeys": [
                {
                    "path": "[concat('/home/', parameters('adminUsername'),
'/.ssh/authorized_keys')]",
                    "keyData": "[parameters('adminPublicKey')]"
```

Next steps

For more information, see Provisioning Linux.

Prepare a Windows VHD or VHDX to upload to Azure

Article • 08/22/2024

Applies to: ✔ Windows VMs

Before you upload a Windows virtual machine (VM) from on-premises to Azure, you must prepare the virtual hard disk (VHD or VHDX). Azure supports both generation 1 and generation 2 VMs that are in VHD file format and that have a fixed-size disk. The maximum size allowed for the OS VHD on a generation 1 VM is 2 TB. You can validate your VHD or VHDX file by refering to this documentation.

You can convert a VHDX file to VHD, convert a dynamically expanding disk to a fixed-size disk, but you can't change a VM's generation. For more information, see Should I create a generation 1 or 2 VM in Hyper-V? and Support for generation 2 VMs on Azure.

For information about the support policy for Azure VMs, see Microsoft server software support for Azure VMs.

> **Note**
>
> The instructions in this article apply to:
>
> - The 64-bit version of Windows Server 2008 R2 and later Windows Server operating systems. For information about running a 32-bit operating system in Azure, see Support for 32-bit operating systems in Azure VMs.
> - If any Disaster Recovery tool will be used to migrate the workload, like Azure Site Recovery or Azure Migrate, this process is still required on the Guest OS to prepare the image before the migration.

System File Checker

Run Windows System File Checker utility before generalization of OS image

The System File Checker (SFC) is used to verify and replace Windows system files.

> **Important**
>
> Use an elevated PowerShell session to run the examples in this article.

Run the SFC command:

PowerShell

```
sfc.exe /scannow
```

Output

```
Beginning system scan.  This process will take some time.

Beginning verification phase of system scan.
Verification 100% complete.

Windows Resource Protection did not find any integrity violations.
```

After the SFC scan completes, install Windows Updates and restart the computer.

Set Windows configurations for Azure

> **Note**
>
> Azure platform mounts an ISO file to the DVD-ROM when a Windows VM is created from a generalized image. For this reason, the DVD-ROM must be enabled in the OS in the generalized image. If it is disabled, the Windows VM will be stuck at out-of-box experience (OOBE).

1. Remove any static persistent routes in the routing table:

   - To view the routing table, run `route.exe print`.
   - Check the **Persistence Routes** section. If there's a persistent route, use the `route.exe delete` command to remove it.

2. Remove the WinHTTP proxy:

   PowerShell

   ```
   netsh.exe winhttp reset proxy
   ```

   If the VM needs to work with a specific proxy, add a proxy exception for the Azure IP address (168.63.129.16) so the VM can connect to Azure:

   ```
   $proxyAddress='<your proxy server>'
   $proxyBypassList='<your list of bypasses>;168.63.129.16'
   netsh.exe winhttp set proxy $proxyAddress $proxyBypassList
   ```

3. Open DiskPart:

   PowerShell

   ```
   diskpart.exe
   ```

   Set the disk SAN policy to Onlineall:

   DiskPart

   ```
   DISKPART> san policy=onlineall
   DISKPART> exit
   ```

4. Set Coordinated Universal Time (UTC) time for Windows. Also, set the startup type of the Windows time service **w32time** to **Automatic**:

   PowerShell

   ```
   Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\TimeZoneInformation -Name
   RealTimeIsUniversal -Value 1 -Type DWord -Force
   Set-Service -Name w32time -StartupType Automatic
   ```

5. Set the power profile to high performance:

   PowerShell

   ```
   powercfg.exe /setactive SCHEME_MIN
   powercfg /setacvalueindex SCHEME_CURRENT SUB_VIDEO VIDEOIDLE 0
   ```

6. Make sure the environmental variables **TEMP** and **TMP** are set to their default values:

PowerShell

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment' -Name TEMP -
Value "%SystemRoot%\TEMP" -Type ExpandString -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Environment' -Name TMP -
Value "%SystemRoot%\TEMP" -Type ExpandString -Force
```

7. For VMs with legacy operating systems (Windows Server 2012 R2 or Windows 8.1 and below), make sure the latest Hyper-V Integration Component Services are installed. For more information, see Hyper-V integration components update for Windows VM.

> **Note**
>
> In a scenario where VMs are to be set up with a disaster recovery solution between the on-premises VMware server and Azure, the Hyper-V Integration Component Services can't be used. If that's the case, please contact the VMware support to migrate the VM to Azure and make it co-reside in VMware server.

Check the Windows services

Make sure that each of the following Windows services is set to the Windows default value. These services are the minimum that must be configured to ensure VM connectivity. To set the startup settings, run the following example:

PowerShell

```
Get-Service -Name BFE, Dhcp, Dnscache, IKEEXT, iphlpsvc, nsi, mpssvc, RemoteRegistry |
  Where-Object StartType -ne Automatic |
    Set-Service -StartupType Automatic

Get-Service -Name Netlogon, Netman, TermService |
  Where-Object StartType -ne Manual |
    Set-Service -StartupType Manual
```

Update remote desktop registry settings

Make sure the following settings are configured correctly for remote access:

> **Note**
>
> If you receive an error message when running `Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name <string> -Value <object>`, you can safely ignore it. It means the domain isn't setting that configuration through a Group Policy Object.

1. Remote Desktop Protocol (RDP) is enabled:

   PowerShell

   ```
   Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server' -Name fDenyTSConnections -
   Value 0 -Type DWord -Force
   Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
   fDenyTSConnections -Value 0 -Type DWord -Force
   ```

2. The RDP port is set up correctly using the default port of 3389:

   PowerShell

   ```
   Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name
   PortNumber -Value 3389 -Type DWord -Force
   ```

When you deploy a VM, the default rules are created for port 3389. To change the port number, do that after the VM is deployed in Azure.

3. The listener is listening on every network interface:

```
PowerShell

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name
LanAdapter -Value 0 -Type DWord -Force
```

4. Configure network-level authentication (NLA) mode for the RDP connections:

```
PowerShell

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp' -Name
UserAuthentication -Value 1 -Type DWord -Force
```

5. Set the keep-alive value:

```
PowerShell

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
KeepAliveEnable -Value 1  -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
KeepAliveInterval -Value 1  -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name
KeepAliveTimeout -Value 1 -Type DWord -Force
```

6. Set the reconnect options:

```
PowerShell

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services' -Name
fDisableAutoReconnect -Value 0 -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name
fInheritReconnectSame -Value 1 -Type DWord -Force
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name
fReconnectSame -Value 0 -Type DWord -Force
```

7. Limit the number of concurrent connections:

```
PowerShell

Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\Winstations\RDP-Tcp' -Name
MaxInstanceCount -Value 4294967295 -Type DWord -Force
```

8. Remove any self-signed certificates tied to the RDP listener:

```
PowerShell

if ((Get-Item -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp').Property
-contains 'SSLCertificateSHA1Hash')
{
    Remove-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp'
-Name SSLCertificateSHA1Hash -Force
}
```

This code ensures that you can connect when you deploy the VM. You can also review these settings after the VM is deployed in Azure.

9. If the VM is part of a domain, check the following policies to make sure the previous settings aren't reverted.

Expand table

Goal	Policy	Value
RDP is enabled	Computer Configuration\Policies\Windows Settings\Administrative Templates\Components\Remote Desktop Services\Remote Desktop Session Host\Connections	Allow users to connect remotely by using Remote Desktop
NLA group policy	Settings\Administrative Templates\Components\Remote Desktop Services\Remote Desktop Session Host\Security	Require user authentication for remote access by using NLA
Keep-alive settings	Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections	Configure keep-alive connection interval
Reconnect settings	Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections	Reconnect automatically
Limited number of connection settings	Computer Configuration\Policies\Windows Settings\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections	Limit number of connections

Configure Windows Firewall rules

1. Turn on Windows Firewall on the three profiles (domain, standard, and public):

```
PowerShell

Set-NetFirewallProfile -Profile Domain, Public, Private -Enabled True
```

2. Run the following example to allow WinRM through the three firewall profiles (domain, private, and public), and enable the PowerShell remote service:

```
PowerShell

Enable-PSRemoting -Force
```

> **Note**
>
> When this command runs, it enables the appropriate firewall rules automatically.

1. Enable the following firewall rules to allow the RDP traffic:

```
PowerShell

Get-NetFirewallRule -DisplayGroup 'Remote Desktop' | Set-NetFirewallRule -Enabled True
```

2. Enable the rule for file and printer sharing so the VM can respond to ping requests inside the virtual network:

```
PowerShell

Set-NetFirewallRule -Name FPS-ICMP4-ERQ-In -Enabled True
```

3. Create a rule for the Azure platform network:

```
PowerShell

New-NetFirewallRule -DisplayName AzurePlatform -Direction Inbound -RemoteAddress 168.63.129.16 -Profile
Any -Action Allow -EdgeTraversalPolicy Allow
New-NetFirewallRule -DisplayName AzurePlatform -Direction Outbound -RemoteAddress 168.63.129.16 -Profile
Any -Action Allow
```

> **Important**
>
> 168.63.129.16 is a special public IP address that is owned by Microsoft for Azure. For more information, see What is IP address 168.63.129.16.

1. If the VM is part of a domain, check the following Microsoft Entra policies to make sure the previous settings aren't reverted.

Expand table

Goal	Policy	Value
Enable the Windows Firewall profiles	Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall	Protect all network connections
Enable RDP	Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall	Allow inbound Remote Desktop exceptions
	Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Standard Profile\Windows Firewall	Allow inbound Remote Desktop exceptions
Enable ICMP-V4	Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Domain Profile\Windows Firewall	Allow ICMP exceptions
	Computer Configuration\Policies\Windows Settings\Administrative Templates\Network\Network Connection\Windows Firewall\Standard Profile\Windows Firewall	Allow ICMP exceptions

Verify the VM

Make sure the VM is healthy, secure, and RDP accessible:

1. To make sure the disk is healthy and consistent, check the disk at the next VM restart:

```
PowerShell

chkdsk.exe /f
```

   Make sure the report shows a clean and healthy disk.

2. Set the Boot Configuration Data (BCD) settings.

```
PowerShell

cmd

bcdedit.exe /set "{bootmgr}" integrityservices enable
bcdedit.exe /set "{default}" device partition=C:
bcdedit.exe /set "{default}" integrityservices enable
bcdedit.exe /set "{default}" recoveryenabled Off
bcdedit.exe /set "{default}" osdevice partition=C:
bcdedit.exe /set "{default}" bootstatuspolicy IgnoreAllFailures

#Enable Serial Console Feature
bcdedit.exe /set "{bootmgr}" displaybootmenu yes
bcdedit.exe /set "{bootmgr}" timeout 5
bcdedit.exe /set "{bootmgr}" bootems yes
bcdedit.exe /ems "{current}" ON
bcdedit.exe /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

```
exit
```

3. The dump log can be helpful in troubleshooting Windows crash issues. Enable the dump log collection:

```PowerShell
# Set up the guest OS to collect a kernel dump on an OS crash event
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name CrashDumpEnabled -Type
DWord -Force -Value 2
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name DumpFile -Type
ExpandString -Force -Value "%SystemRoot%\MEMORY.DMP"
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\CrashControl' -Name NMICrashDump -Type
DWord -Force -Value 1

# Set up the guest OS to collect user mode dumps on a service crash event
$key = 'HKLM:\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps'
if ((Test-Path -Path $key) -eq $false) {(New-Item -Path 'HKLM:\SOFTWARE\Microsoft\Windows\Windows Error
Reporting' -Name LocalDumps)}
New-ItemProperty -Path $key -Name DumpFolder -Type ExpandString -Force -Value 'C:\CrashDumps'
New-ItemProperty -Path $key -Name CrashCount -Type DWord -Force -Value 10
New-ItemProperty -Path $key -Name DumpType -Type DWord -Force -Value 2
Set-Service -Name WerSvc -StartupType Manual
```

4. Verify that the Windows Management Instrumentation (WMI) repository is consistent:

```PowerShell
winmgmt.exe /verifyrepository
```

   If the repository is corrupted, see WMI: Repository corruption or not.

5. Make sure no other applications than TermService are using port 3389. This port is used for the RDP service in Azure. To see which ports are used on the VM, run `netstat.exe -anob`:

```PowerShell
netstat.exe -anob
```

   The following is an example.

```PowerShell
netstat.exe -anob | findstr 3389
TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       4056
TCP    [::]:3389              [::]:0                 LISTENING       4056
UDP    0.0.0.0:3389           *:*                                    4056
UDP    [::]:3389              *:*                                    4056

tasklist /svc | findstr 4056
svchost.exe                  4056 TermService
```

6. To upload a Windows VHD that's a domain controller:

   - Follow these extra steps to prepare the disk.
   - Make sure you know the Directory Services Restore Mode (DSRM) password in case you ever have to start the VM in DSRM. For more information, see Set a DSRM password.

7. Make sure you know the built-in administrator account and password. You might want to reset the current local administrator password and make sure you can use this account to sign in to Windows through the RDP connection. This access permission is controlled by the "Allow log on through Remote Desktop Services" Group Policy Object. View this object in the Local Group Policy Editor:

   - `Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment`

8. Check the following Microsoft Entra policies to make sure they're not blocking RDP access:

   - `Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Deny access to this computer from the network`
   - `Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Deny log on through Remote Desktop Services`

9. Check the following Microsoft Entra policy to make sure they're not removing any of the required access accounts:

   - `Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Access this computer from the network`

   The policy should list the following groups:

   - Administrators
   - Backup Operators
   - Everyone
   - Users

10. Restart the VM to make sure that Windows is still healthy and can be reached through the RDP connection. At this point, consider creating a VM on your local Hyper-V server to make sure the VM starts completely. Then test to make sure you can reach the VM through RDP.
11. Remove any extra Transport Driver Interface (TDI) filters. For example, remove software that analyzes TCP packets or extra firewalls.
12. Uninstall any other third-party software or driver that's related to physical components or any other virtualization technology.

Install Windows updates

> **Note**
>
> To avoid an accidental reboot during the VM provisioning, we recommend completing all Windows update installations and to make sure there's no pending restart. One way to do this is to install all Windows updates and to reboot the VM before performing the migration to Azure.
>
> If you also need to do a generalization of the OS (sysprep), you must update Windows and restart the VM before running the Sysprep command.

Ideally, you should keep the machine updated to the *patch level*, if this isn't possible, make sure the following updates are installed. To get the latest updates, see the Windows update history pages: Windows 10, and Windows Server 2019, Windows 8.1, and Windows Server 2012 R2 and Windows 7 SP1, and Windows Server 2008 R2 SP1.

Expand table

Component	Binary	Windows 7 SP1, Windows Server 2008 R2 SP1	Windows 8, Windows Server 2012	Windows 8.1, Windows Server 2012 R2	Windows 10 v1607, Windows Server 2016 v1607	Windows 10 v1703	Windows 10 v1709, Windows Server 2016 v1709	Windows 10 v1803, Windows Server 2016 v1803
Storage	disk.sys	6.1.7601.23403 - KB3125574	6.2.9200.17638 /	6.3.9600.18203 - KB3137061	-	-	-	-

Component	Binary	Windows 7 SP1, Windows Server 2008 R2 SP1	Windows 8, Windows Server 2012	Windows 8.1, Windows Server 2012 R2	Windows 10 v1607, Windows Server 2016 v1607	Windows 10 v1703	Windows 10 v1709, Windows Server 2016 v1709	Windows 10 v1803, Windows Server 2016 v1803
			6.2.9200.21757 - KB3137061					
	storport.sys	6.1.7601.23403 - KB3125574	6.2.9200.17188 / 6.2.9200.21306 - KB3018489	6.3.9600.18573 - KB4022726	10.0.14393.1358 - KB4022715	10.0.15063.332	-	-
	ntfs.sys	6.1.7601.23403 - KB3125574	6.2.9200.17623 / 6.2.9200.21743 - KB3121255	6.3.9600.18654 - KB4022726	10.0.14393.1198 - KB4022715	10.0.15063.447	-	-
	lologmsg.dll	6.1.7601.23403 - KB3125574	6.2.9200.16384 - KB2995387	-	-	-	-	-
	Classpnp.sys	6.1.7601.23403 - KB3125574	6.2.9200.17061 / 6.2.9200.21180 - KB2995387	6.3.9600.18334 - KB3172614	10.0.14393.953 - KB4022715	-	-	-
	Volsnap.sys	6.1.7601.23403 - KB3125574	6.2.9200.17047 / 6.2.9200.21165 - KB2975331	6.3.9600.18265 - KB3145384	-	10.0.15063.0	-	-
	partmgr.sys	6.1.7601.23403 - KB3125574	6.2.9200.16681 - KB2877114	6.3.9600.17401 - KB3000850	10.0.14393.953 - KB4022715	10.0.15063.0	-	-
	volmgr.sys					10.0.15063.0	-	-
	Volmgrx.sys	6.1.7601.23403 - KB3125574	-	-	-	10.0.15063.0	-	-
	Msiscsi.sys	6.1.7601.23403 - KB3125574	6.2.9200.21006 - KB2955163	6.3.9600.18624 - KB4022726	10.0.14393.1066 - KB4022715	10.0.15063.447	-	-
	Msdsm.sys	6.1.7601.23403 - KB3125574	6.2.9200.21474 - KB3046101	6.3.9600.18592 - KB4022726	-	-	-	-
	Mpio.sys	6.1.7601.23403 - KB3125574	6.2.9200.21190 - KB3046101	6.3.9600.18616 - KB4022726	10.0.14393.1198 - KB4022715	-	-	-
	vmstorfl.sys	6.3.9600.18907 - KB4072650	6.3.9600.18080 - KB3063109	6.3.9600.18907 - KB4072650	10.0.14393.2007 - KB4345418	10.0.15063.850 - KB4345419	10.0.16299.371 - KB4345420	-
	Fveapi.dll	6.1.7601.23311 - KB3125574	6.2.9200.20930 - KB2930244	6.3.9600.18294 - KB3172614	10.0.14393.576 - KB4022715	-	-	-
	Fveapibase.dll	6.1.7601.23403 - KB3125574	6.2.9200.20930 - KB2930244	6.3.9600.17415 - KB3172614	10.0.14393.206 - KB4022715	-	-	-
Network	netvsc.sys	-	-	-	10.0.14393.1198 - KB4022715	10.0.15063.250 - KB4020001	-	-
	mrxsmb10.sys	6.1.7601.23816 - KB4022722	6.2.9200.22108 - KB4022724	6.3.9600.18603 - KB4022726	10.0.14393.479 - KB4022715	10.0.15063.483	-	-
	mrxsmb20.sys	6.1.7601.23816 - KB4022722	6.2.9200.21548 - KB4022724	6.3.9600.18586 - KB4022726	10.0.14393.953 - KB4022715	10.0.15063.483	-	-
	mrxsmb.sys	6.1.7601.23816 - KB4022722	6.2.9200.22074 - KB4022724	6.3.9600.18586 - KB4022726	10.0.14393.953 - KB4022715	10.0.15063.0	-	-

Component	Binary	Windows 7 SP1, Windows Server 2008 R2 SP1	Windows 8, Windows Server 2012	Windows 8.1, Windows Server 2012 R2	Windows 10 v1607, Windows Server 2016 v1607	Windows 10 v1703	Windows 10 v1709, Windows Server 2016 v1709	Windows 10 v1803, Windows Server 2016 v1803
	tcpip.sys	6.1.7601.23761 - KB4022722	6.2.9200.22070 - KB4022724	6.3.9600.18478 - KB4022726	10.0.14393.1358 - KB4022715	10.0.15063.447	-	-
	http.sys	6.1.7601.23403 - KB3125574	6.2.9200.17285 - KB3042553	6.3.9600.18574 - KB4022726	10.0.14393.251 - KB4022715	10.0.15063.483	-	-
	vmswitch.sys	6.1.7601.23727 - KB4022719	6.2.9200.22117 - KB4022724	6.3.9600.18654 - KB4022726	10.0.14393.1358 - KB4022715	10.0.15063.138	-	-
Core	ntoskrnl.exe	6.1.7601.23807 - KB4022719	6.2.9200.22170 - KB4022718	6.3.9600.18696 - KB4022726	10.0.14393.1358 - KB4022715	10.0.15063.483	-	-
Remote Desktop Services	rdpcorets.dll	6.2.9200.21506 - KB4022719	6.2.9200.22104 - KB4022724	6.3.9600.18619 - KB4022726	10.0.14393.1198 - KB4022715	10.0.15063.0	-	-
	termsrv.dll	6.1.7601.23403 - KB3125574	6.2.9200.17048 - KB2973501	6.3.9600.17415 - KB3000850	10.0.14393.0 - KB4022715	10.0.15063.0	-	-
	termdd.sys	6.1.7601.23403 - KB3125574	-	-	-	-	-	-
	win32k.sys	6.1.7601.23807 - KB4022719	6.2.9200.22168 - KB4022718	6.3.9600.18698 - KB4022726	10.0.14393.594 - KB4022715	-	-	-
	rdpdd.dll	6.1.7601.23403 - KB3125574	-	-	-	-	-	-
	rdpwd.sys	6.1.7601.23403 - KB3125574	-	-	-	-	-	-
Security	MS17-010	KB4012212	KB4012213	KB4012213	KB4012606	KB4012606	-	-
			KB4012216		KB4013198	KB4013198	-	-
		KB4012215	KB4012214	KB4012216	KB4013429	KB4013429	-	-
			KB4012217		KB4013429	KB4013429	-	-
	CVE-2018-0886	KB4103718	KB4103730	KB4103725	KB4103723	KB4103731	KB4103727	KB4103721
		KB4103712	KB4103726	KB4103715				

> **Note**
>
> To avoid an accidental reboot during VM provisioning, we recommend ensuring that all Windows Update installations are finished and that no updates are pending. One way to do this is to install all possible Windows updates and reboot once before you run the `sysprep.exe` command.

Determine when to use Sysprep

System Preparation Tool (`sysprep.exe`) is a process you can run to reset a Windows installation. Sysprep provides an "out of the box" experience by removing all personal data and resetting several components.

You typically run `sysprep.exe` to create a template from which you can deploy several other VMs that have a specific configuration. The template is called a *generalized image*.

To create only one VM from one disk, you don't have to use Sysprep. Instead, you can create the VM from a *specialized image*. For information about how to create a VM from a specialized disk, see:

- Create a VM from a specialized disk
- Create a VM from a specialized VHD disk

To create a generalized image, you need to run Sysprep. For more information, see How to use Sysprep: An introduction.

Not every role or application that's installed on a Windows-based computer supports generalized images. Before you use this procedure, make sure Sysprep supports the role of the computer. For more information, see Sysprep support for server roles.

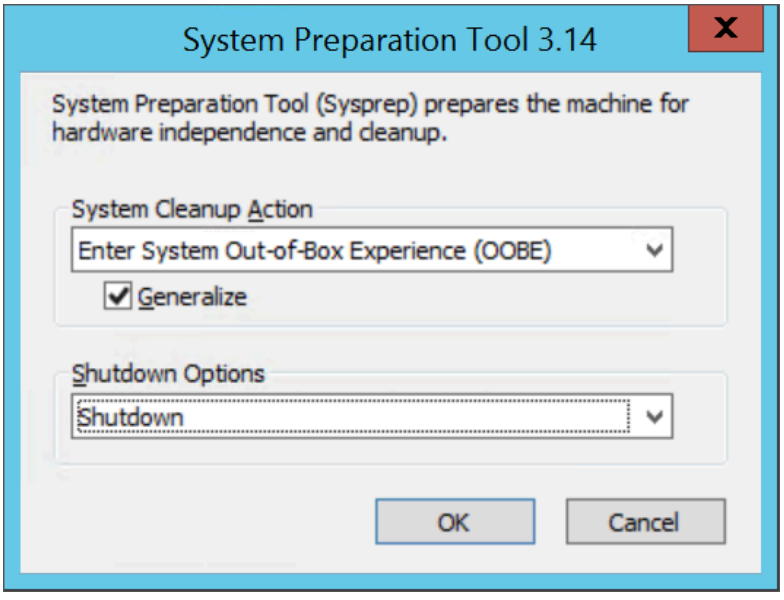
In particular, Sysprep requires the drives to be fully decrypted before execution. If you have enabled encryption on your VM, disable it before running Sysprep.

Generalize a VHD

> **Note**
>
> If you're creating a generalized image from an existing Azure VM, we recommend to remove the VM extensions before running the sysprep.

> **Note**
>
> After you run `sysprep.exe` in the following steps, turn off the VM. Don't turn it back on until you create an image from it in Azure.

1. Sign in to the Windows VM.
2. Run a PowerShell session as an administrator.
3. Delete the panther directory (C:\Windows\Panther).
4. Change the directory to `%windir%\system32\sysprep`. Then run `sysprep.exe`.
5. In the **System Preparation Tool** dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure the **Generalize** checkbox is selected.



6. In **Shutdown Options**, select **Shutdown**.
7. Select **OK**.
8. When Sysprep finishes, shut down the VM. Don't use **Restart** to shut down the VM.

Now the VHD is ready to be uploaded. For more information about how to create a VM from a generalized disk, see Upload a generalized VHD and use it to create a new VM in Azure.

> ⓘ Note
>
> A custom *unattend.xml* file is not supported. Although we do support the **additionalUnattendContent** property, that provides only limited support for adding microsoft-windows-shell-setup options into the *unattend.xml* file that the Azure provisioning agent uses. You can use, for example, additionalUnattendContent to add FirstLogonCommands and LogonCommands. For more information, see additionalUnattendContent FirstLogonCommands example.

Convert the virtual disk to a fixed size VHD

> ⓘ Note
>
> If you're going to use Azure PowerShell to upload your disk to Azure and you have Hyper-V enabled, this step is optional. Add-AzVHD will perform it for you.

Use one of the methods in this section to convert and resize your virtual disk to the required format for Azure:

1. Back up the VM before you run the virtual disk conversion or resize process.
2. Make sure that the Windows VHD works correctly on the local server. Resolve any errors within the VM itself before you try to convert or upload it to Azure.
3. Convert the virtual disk to type fixed.
4. Resize the virtual disk to meet Azure requirements:

   a. Disks in Azure must have a virtual size aligned to 1 MiB. If your VHD is a fraction of 1 MiB, you'll need to resize the disk to a multiple of 1 MiB. Disks that are fractions of a MiB cause errors when creating images from the uploaded VHD. To verify the size you can use the PowerShell Get-VHD cmdlet to show "Size", which must be a multiple of 1 MiB in Azure, and "FileSize", which will be equal to "Size" plus 512 bytes for the VHD footer.

   ```
   PowerShell
   ```

   ```
   $vhd = Get-VHD -Path C:\test\MyNewVM.vhd
   $vhd.Size % 1MB
   0
   $vhd.FileSize - $vhd.Size
   512
   ```

   b. The maximum size allowed for the OS VHD with a generation 1 VM is 2,048 GiB (2 TiB),

   c. The maximum size for a data disk is 32,767 GiB (32 TiB).

> ⓘ Note
>
> - If you are preparing a Windows OS disk after you convert to a fixed disk and resize if needed, create a VM that uses the disk. Start and sign in to the VM and continue with the sections in this article to finish preparing it for uploading.
> - If you are preparing a data disk you may stop with this section and proceed to uploading your disk.

Use Hyper-V Manager to convert the disk

1. Open Hyper-V Manager and select your local computer on the left. In the menu above the computer list, select **Action** > **Edit Disk**.
2. On the **Locate Virtual Hard Disk** page, select your virtual disk.
3. On the **Choose Action** page, select **Convert** > **Next**.

4. To convert from VHDX, select **VHD** > **Next**.
5. To convert from a dynamically expanding disk, select **Fixed size** > **Next**.
6. Locate and select a path to save the new VHD file.
7. Select **Finish**.

Use PowerShell to convert the disk

You can convert a virtual disk using the Convert-VHD cmdlet in PowerShell. If you need information about installing this cmdlet see Install the Hyper-V role.

> ⓘ **Note**
>
> If you're going to use Azure PowerShell to upload your disk to Azure and you have Hyper-V enabled, this step is optional. Add-AzVHD will perform it for you.

The following example converts the disk from VHDX to VHD. It also converts the disk from a dynamically expanding disk to a fixed-size disk.

```
PowerShell

Convert-VHD -Path C:\test\MyVM.vhdx -DestinationPath C:\test\MyNewVM.vhd -VHDType Fixed
```

In this example, replace the value for **Path** with the path to the virtual hard disk that you want to convert. Replace the value for **DestinationPath** with the new path and name of the converted disk.

Use Hyper-V Manager to resize the disk

> ⓘ **Note**
>
> If you're going to use Azure PowerShell to upload your disk to Azure and you have Hyper-V enabled, this step is optional. Add-AzVHD will perform it for you.

1. Open Hyper-V Manager and select your local computer on the left. In the menu above the computer list, select **Action** > **Edit Disk**.
2. On the **Locate Virtual Hard Disk** page, select your virtual disk.
3. On the **Choose Action** page, select **Expand** > **Next**.
4. On the **Locate Virtual Hard Disk** page, enter the new size in GiB > **Next**.
5. Select **Finish**.

Use PowerShell to resize the disk

> ⓘ **Note**
>
> If you're going to use Azure PowerShell to upload your disk to Azure and you have Hyper-V enabled, this step is optional. Add-AzVHD will perform it for you.

You can resize a virtual disk using the Resize-VHD cmdlet in PowerShell. If you need information about installing this cmdlet see Install the Hyper-V role.

The following example resizes the disk from 100.5 MiB to 101 MiB to meet the Azure alignment requirement.

```
PowerShell

Resize-VHD -Path C:\test\MyNewVM.vhd -SizeBytes 105906176
```

In this example, replace the value for **Path** with the path to the virtual hard disk that you want to resize. Replace the value for **SizeBytes** with the new size in bytes for the disk.

Convert from VMware VMDK disk format

If you have a Windows VM image in the VMDK file format, then you can use Azure Migrate to convert the VMDK and upload it to Azure.

Complete the recommended configurations

The following settings don't affect VHD uploading. However, we strongly recommend that you configured them.

- Install the Azure Virtual Machine Agent. Then you can enable VM extensions. The VM extensions implement most of the critical functionality that you might want to use with your VMs. You'll need the extensions, for example, to reset passwords or configure RDP. For more information, see the Azure Virtual Machine Agent overview.
- After you create the VM in Azure, we recommend that you put the page file on the *temporal drive volume* to improve performance. You can set up the file placement as follows:

  PowerShell

  ```
  Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management' -Name
  PagingFiles -Value 'D:\pagefile.sys' -Type MultiString -Force
  ```

  If a data disk is attached to the VM, the temporal drive volume's letter is typically *D*. This designation could be different, depending on your settings and the number of available drives.
  - We recommend disabling script blockers that might be provided by antivirus software. They might interfere and block the Windows Provisioning Agent scripts executed when you deploy a new VM from your image.

> **Tip**
>
> **Optional** Use DISM to optimize your image and reduce your VM's first boot time.
>
> To optimize your image, mount your VHD by double-clicking on it in Windows explorer, and then run DISM with the `/optimize-image` parameter.
>
> Windows Command Prompt
>
> ```
> DISM /image:D:\ /optimize-image /boot
> ```
>
> Where D: is the mounted VHD's path.
>
> Running `DISM /optimize-image` should be the last modification you make to your VHD. If you make any changes to your VHD prior to deployment, you'll have to run `DISM /optimize-image` again.

Next steps

- Upload a Windows VM image to Azure for Resource Manager deployments
- Troubleshoot Azure Windows VM activation problems

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Visual Studio images on Azure

Using Visual Studio in a preconfigured Azure virtual machine (VM) is a quick, easy way to go from nothing to an up-and-running development environment. System images with different Visual Studio configurations are available in the Azure Marketplace.

New to Azure? Create a free Azure account.

What configurations and versions are available?

Images for the latest major versions, Visual Studio 2022, Visual Studio 2019, and Visual Studio 2017 are available in the Azure Marketplace. These images are updated at least once a month to include the latest Visual Studio and Windows updates. Although the image names remain unchanged, each image's description includes the installed product version and the date it reflects

Expand table

Release version	Editions	Product version
Visual Studio 2022 (Microsoft Dev Box compatible)	Enterprise, Professional, Community	Version 17.12
Visual Studio 2019 (Microsoft Dev Box compatible)	Enterprise, Professional	Version 16.11
Visual Studio 2017: Latest (Version 15.9)	Enterprise, Community	Version 15.9
Visual Studio on Windows Server	Enterprise, Professional, Community	VS 2022 and VS 2019 on Windows Server 2025

The Azure VM image offers help you create and test applications more quickly and easily on both Azure VMs and the Microsoft Dev Box service. For updates on newly added or deprecated images, see the DevBlog post Giving our Azure marketplace images a makeover. To learn more, visit the Visual Studio Servicing Policy.

Windows client for developers (Microsoft Dev Box compatible), a Windows 11 Enterprise image is meant to be used as a starting point for creating custom Dev Box images, and contains additional tools and settings to optimize the developer experience. This image doesn't include Visual Studio. For Visual Studio developers, we recommend using our Visual Studio 2022 or Visual Studio 2019 Dev Box compatible images as a starting point.

What features are installed?

Each image contains the recommended feature set for that Visual Studio edition. Generally, the installation includes:

- All available workloads, including each workload's recommended optional components. Learn more about workloads, components, and SDKs included Visual Studio.
- .NET 4.6.2, .NET 4.7, and .NET 4.8 SDKs, Targeting Packs, and Developer Tools
- Visual F#
- GitHub Extension for Visual Studio
- LINQ to SQL Tools

We use the following command line to install Visual Studio when building the images:

shell

```
vs_enterprise.exe --allWorkloads --includeRecommended --passive ^
   --add Microsoft.Net.Component.4.8.SDK ^
   --add Microsoft.Net.Component.4.7.2.SDK ^
   --add Microsoft.Net.Component.4.7.2.TargetingPack ^
   --add Microsoft.Net.Component.4.6.2.SDK ^
   --add Microsoft.Net.Component.4.6.2.TargetingPack ^
   --add Microsoft.Net.ComponentGroup.4.8.DeveloperTools ^
   --add Microsoft.Net.ComponentGroup.4.7.2.DeveloperTools ^
   --add Microsoft.VisualStudio.Component.FSharp ^
   --add Component.GitHub.VisualStudio ^
   --add Microsoft.VisualStudio.Component.LinqToSql
```

If the images don't include a Visual Studio feature that you require, please provide feedback through the feedback tool in the upper-right corner of the page.

What size VM should I choose?

Azure offers a full range of virtual machine sizes. Because Visual Studio is a powerful, multi-threaded application, you want a VM size that includes at least two processors and 7 GB of memory. We recommend the following VM sizes for the Visual Studio images:

- Standard_D2_v3
- Standard_D2s_v3
- Standard_D4_v3
- Standard_D4s_v3
- Standard_D2_v2
- Standard_D2S_v2

- Standard_D3_v2

For more information on the latest machine sizes, see Sizes for Windows virtual machines in Azure.

With Azure, you can rebalance your initial choice by resizing the VM. You can either provision a new VM with a more appropriate size, or resize your existing VM to different underlying hardware. For more information, see Resize a Windows VM.

After the VM is running, what's next?

Visual Studio follows the “bring your own license” model in Azure. As with an installation on proprietary hardware, one of the first steps is licensing your Visual Studio installation. To unlock Visual Studio, either:

- Sign in with a Microsoft account that’s associated with a Visual Studio subscription
- Unlock Visual Studio with the product key that came with your initial purchase

For more information, see Sign in to Visual Studio and How to unlock Visual Studio.

How do I save the development VM for future or team use?

The spectrum of development environments is huge, and there’s real cost associated with building out the more complex environments. Regardless of your environment’s configuration, you can save, or capture, your configured VM as a "base image" for future use or for other members of your team. Then, when booting a new VM, you provision it from the base image rather than the Azure Marketplace image.

A quick summary: Use the System Preparation tool (Sysprep) and shut down the running VM, and then capture *(Figure 1)* the VM as an image through the UI in the Azure portal. Azure saves the `.vhd` file that contains the image in the storage account of your choosing. The new image then shows up as an Image resource in your subscription’s list of resources.



(Figure 1) Capture an image through the Azure portal’s UI.

For more information, see Create a managed image of a generalized VM in Azure.

> **Important**
>
> Don’t forget to use Sysprep to prepare the VM. If you miss that step, Azure can't provision a VM from the image.

> **Note**
>
> You still incur some cost for storage of the images, but that incremental cost can be insignificant compared to the overhead costs to rebuild the VM from scratch for each team member who needs one. For instance, it costs a few dollars to create and store a 127-GB image for a month that's reusable by your entire team. However, these costs are insignificant compared to hours each employee invests to build out and validate a properly configured dev box for their individual use.

Additionally, your development tasks or technologies might need more scale, like varieties of development configurations and multiple machine configurations. You can use Azure DevTest Labs to create *recipes* that automate construction of your "golden image." You can also use DevTest Labs to manage policies for your team’s running VMs. Using Azure DevTest Labs for developers is the best source for more information on DevTest Labs.

Related content

Now that you know about the preconfigured Visual Studio images, the next step is to create a new VM:

- Create a VM through the Azure portal
- Windows Virtual Machines overview

Last updated on 10/30/2025

Azure VM Image Builder overview

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

By using standardized virtual machine (VM) images, your organization can more easily migrate to the cloud and help ensure consistency in your deployments. Images ordinarily include predefined security, configuration settings, and any necessary software. Setting up your own imaging pipeline requires time, infrastructure, and many other details. With Azure VM Image Builder, you need only create a configuration that describes your image and submit it to the service, where the image is built and then distributed.

With VM Image Builder, you can migrate your existing image customization pipeline to Azure as you continue to use existing scripts, commands, and processes. You can integrate your core applications into a VM image, so that your VMs can take on workloads after the images are created. You can even add configurations to build images for Azure Virtual Desktop, as virtual hard disks (VHDs) for use in Azure Stack, or for ease of exporting.

VM Image Builder lets you start with Windows or Linux images either from Azure Marketplace or as existing custom images, and then add your own customizations. You can also specify where you want your resulting images to be hosted in Azure Compute Gallery (formerly Shared Image Gallery), as managed images or as VHDs.

Features

Although it's possible to create custom VM images by hand or by other tools, the process can be cumbersome and unreliable. VM Image Builder, which is built on HashiCorp Packer, gives you the benefits of a managed service.

Simplicity

To reduce the complexity of creating VM images, VM Image Builder:

- Removes the need to use complex tooling, processes, and manual steps to create a VM image. VM Image Builder abstracts out all these details and hides Azure-specific requirements, such as the need to generalize the image (Sysprep). And it gives more advanced users the ability to override such requirements.

- Can be integrated with existing image build pipelines for a click-and-go experience. To do so, you can either call VM Image Builder from your pipeline or use an Azure VM Image Builder service DevOps task (preview).
- Can fetch customization data from various sources, which removes the need to collect them all from one place.
- Can be integrated with Compute Gallery, which creates an image management system with which to distribute, replicate, version, and scale images globally. Additionally, you can distribute the same resulting image as a VHD or as one or more managed images, without having to rebuild them from scratch.

Infrastructure as code

With VM Image Builder, there's no need to manage your long-term infrastructure (for example, storage accounts that hold customization data) or transient infrastructure (for example, temporary VMs for building images).

VM Image Builder stores your VM image build artifacts as Azure resources. This feature removes both the need to maintain offline definitions and the risk of environment drifts that are caused by accidental deletions or updates.

Security

To help keep your images secure, VM Image Builder:

- Enables you to create baseline images (that is, your minimum security and corporate configurations) and allows other departments to customize them further. You can help keep these images secure and compliant by using VM Image Builder to quickly rebuild a golden image that uses the latest patched version of a source image. VM Image Builder also makes it easier for you to build images that meet the Azure Windows security baseline. For more information, see VM Image Builder - Windows baseline template.
- Enables you to fetch your customization artifacts without having to make them publicly accessible. VM Image Builder can use your Azure Managed Identity to fetch these resources, and you can restrict the privileges of this identity as tightly as required by using Azure role-based access control (Azure RBAC). You can both keep your artifacts secret and prevent tampering by unauthorized actors.
- Securely stores copies of customization artifacts, transient compute and storage resources, and their resulting images within your subscription, because access is controlled by Azure RBAC. This level of security, which also applies to the build VM

that's used to create the customized image, helps prevent your customization scripts and files from being copied to an unknown VM in an unknown subscription. And you can achieve a high degree of separation from other customers' workloads by using Isolated VM offerings for the build VM.

- Enables you to connect VM Image Builder to your existing virtual networks, so that you can communicate with existing configuration servers, such as DSC (desired state configuration pull server), Chef, and Puppet, file shares, or any other routable servers and services.

- Can be configured to assign your user-assigned identities to the VM Image Builder build VM (that is, the VM that the VM Image Builder service creates in your subscription and uses to build and customize the image). You can then use these identities at customization time to access Azure resources, including secrets, in your subscription. There's no need to assign VM Image Builder direct access to those resources.

Regions

The VM Image Builder service is available in the following regions:

> **Note**
>
> You can still distribute images outside these regions.

- East US
- East US 2
- West Central US
- West US
- West US 2
- West US 3
- South Central US
- North Europe
- West Europe
- South East Asia
- Australia Southeast
- Australia East
- UK South
- UK West
- Brazil South
- Canada Central

- Central India
- Central US
- France Central
- Germany West Central
- Japan East
- North Central US
- Norway East
- Switzerland North
- Jio India West
- UAE North
- East Asia
- Korea Central
- South Africa North
- Qatar Central
- USGov Arizona (public preview)
- USGov Virginia (public preview)
- China North 3 (public preview)
- Sweden Central
- Poland Central
- Italy North
- Israel Central

To access the Azure VM Image Builder public preview in the Fairfax regions (USGov Arizona and USGov Virginia), you must register the *Microsoft.VirtualMachineImages/FairfaxPublicPreview* feature. To do so, run the following command in either PowerShell or Azure CLI:

Azure PowerShell

Azure PowerShell

```
Register-AzProviderPreviewFeature -ProviderNamespace
Microsoft.VirtualMachineImages -Name FairfaxPublicPreview
```

To access the Azure VM Image Builder public preview in the China North 3 region, you must register the *Microsoft.VirtualMachineImages/MooncakePublicPreview* feature. To do so, run the following command in either PowerShell or Azure CLI:

Azure PowerShell

Azure PowerShell

```
Register-AzProviderPreviewFeature -ProviderNamespace
Microsoft.VirtualMachineImages -Name MooncakePublicPreview
```

OS support

VM Image Builder is designed to work with all Azure Marketplace base operating system images.

> **Note**
>
> You can now use the Azure Image Builder service inside the portal as of March 2023. Get started with building and validating custom images inside the portal.

Confidential VM and Trusted Launch Support

VM Image Builder has extended support for TrustedLaunchSupported and ConfidentialVMSupported images, with certain constraints. Below is the list of constraints:

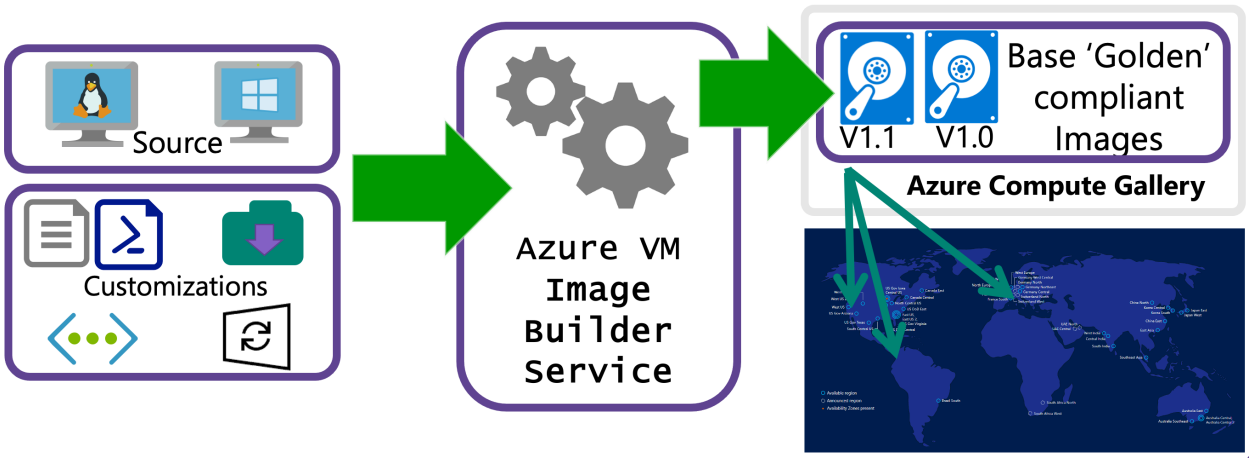
Expand table

SecurityType	Support status
TrustedLaunchSupported	Support as a source image for image builds
ConfidentialVMSupported	Support as a source image for image builds
TrustedLaunch	Not supported as a source image
ConfidentialVM	Not supported as a source image

> **Note**
>
> When using TrustedLaunchSupported images, it's important that the source and distribute must both be TrustedLaunchSupported for it to be supported. If the source is normal and the distribute is TrustedLaunchSupported, or if the source is TrustedLaunchSupported and the distribute is normal Gen2, it's not supported.

How it works

VM Image Builder is a fully managed Azure service that's accessible to Azure resource providers. Resource providers configure it by specifying a source image, a customization to perform, and where the new image is to be distributed. A high-level workflow is illustrated in the following diagram:



You can pass template configurations by using Azure PowerShell, the Azure CLI, or Azure Resource Manager templates, or by using a VM Image Builder DevOps task. When you submit the configuration to the service, Azure creates an *image template resource*. When the image template resource is created, a *staging resource group* is created in your subscription, in the following format: `IT_\<DestinationResourceGroup>_\<TemplateName>_\(GUID)`. The staging resource group contains files and scripts, which are referenced in the File, Shell, and PowerShell customization in the ScriptURI property.

To run the build, you invoke `Run` on the VM Image Builder template resource. The service then deploys additional resources for the build, such as a VM, network, disk, and network adapter.

If you build an image without using an existing virtual network, VM Image Builder also deploys a public IP and network security group, and it connects to the build VM by using Secure Shell (SSH) or Windows Remote Management (WinRM) protocol.

If you select an existing virtual network, the service is deployed via Azure Private Link, and a public IP address isn't required. For more information, see VM Image Builder networking overview.

When the build finishes, all resources are deleted, except for the staging resource group and the storage account. You can remove them by deleting the image template resource, or you can leave them in place to run the build again.

For multiple examples, step-by-step guides, configuration templates, and solutions, go to the VM Image Builder GitHub repository.

Move support

The image template resource is immutable, and it contains links to resources and the staging resource group. Therefore, this resource type doesn't support being moved.

If you want to move the image template resource, either make sure that you have a copy of the configuration template or, if you don't have a copy, extract the existing configuration from the resource. Then, create a new image template resource in the new resource group with a new name, and delete the previous image template resource.

Permissions

When you register for the VM Image Builder service, you're granting the service permission to create, manage, and delete a staging resource group, which is prefixed with `IT_*`. And you have rights to add to it any resources that are required for the image build. This happens because a VM Image Builder service principal name is made available in your subscription after you've registered successfully.

To allow VM Image Builder to distribute images to either the managed images or Compute Gallery, you need to create an Azure user-assigned identity that has permissions to read and write images. If you're accessing Azure Storage, you'll need permissions to read private and public containers.

In API version 2021-10-01 and later, VM Image Builder supports adding Azure user-assigned identities to the build VM to enable scenarios where you need to authenticate with services such as Azure Key Vault in your subscription.

For more information about permissions, see

- Configure VM Image Builder permissions by using PowerShell
- Configure VM Image Builder permissions by using the Azure CLI
- Create a VM Image Builder template

Costs

You'll incur some compute, networking, and storage costs when you create, build, and store images by using VM Image Builder. These costs are similar to those that you incur when you create custom images manually. Your resources are charged at your Azure rates.

> ⓘ **Note**

The Azure Image Builder service currently does not support Azure Hybrid Benefit for Windows Server at this time.

During the image-creation process, files are downloaded and stored in the `IT_<DestinationResourceGroup>_<TemplateName>` resource group, which incurs a small storage cost. If you don't want to keep these files, delete the image template after you've built the image.

VM Image Builder creates a VM by using the default Standard_D1_v2 VM size for Gen1 images and Standard_D2ds_v4 for Gen2 images, along with the storage and networking that's needed for the VM. These resources last for the duration of the build process and are deleted after VM Image Builder has finished creating the image.

VM Image Builder distributes the image to your chosen regions, which might incur network egress charges.

Hyper-V generation

VM Image Builder currently supports creating Hyper-V Gen1 and Gen2 images in a Compute Gallery and as managed images or VHDs. Keep in mind that the distributed image is always in the same generation as the provided image.

For Gen2 images, ensure that you're using the correct SKU. For example, the SKU for an Ubuntu Server 18.04 Gen2 image would be 18_04-lts-gen2. The SKU for an Ubuntu Server 18.04 Gen1 image would be 18.04-lts.

Here's how to find SKUs that are based on the image publisher:

Azure CLI

```
# Find all Gen2 SKUs published by Microsoft Windows Desktop
az vm image list --publisher MicrosoftWindowsDesktop --sku g2 --output table
--all

# Find all Gen2 SKUs published by Canonical
az vm image list --publisher Canonical --sku gen2 --output table --all
```

For more information about Azure VM images that support Gen2, see Gen2 VM images in Azure Marketplace.

Next steps

To try out VM Image Builder, see the articles about building Linux or Windows images.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a Linux image and distribute it to an Azure Compute Gallery by using the Azure CLI

Applies to: ✔ Linux VMs ✔ Flexible scale sets

In this article, you learn how to use Azure VM Image Builder and the Azure CLI to create an image version in an Azure Compute Gallery (formerly Shared Image Gallery) and then distribute the image globally. You can also create an image version by using Azure PowerShell.

This article uses a sample JSON template to configure the image. The JSON file is at helloImageTemplateforSIG.json.

To distribute the image to an Azure Compute Gallery, the template uses sharedImage as the value for the `distribute` section of the template.

Register the providers

To use VM Image Builder, you need to register the providers. Check your registration by running the following commands:

Azure CLI

```
az provider show -n Microsoft.VirtualMachineImages -o json | grep registrationState
az provider show -n Microsoft.KeyVault -o json | grep registrationState
az provider show -n Microsoft.Compute -o json | grep registrationState
az provider show -n Microsoft.Storage -o json | grep registrationState
az provider show -n Microsoft.Network -o json | grep registrationState
az provider show -n Microsoft.ContainerInstance -o json | grep registrationState
```

If the output doesn't say *registered*, run the following commands:

Azure CLI

```
az provider register -n Microsoft.VirtualMachineImages
az provider register -n Microsoft.Compute
az provider register -n Microsoft.KeyVault
az provider register -n Microsoft.Storage
az provider register -n Microsoft.Network
az provider register -n Microsoft.ContainerInstance
```

Set variables and permissions

Because you'll be using some pieces of information repeatedly, create some variables to store that information.

VM Image Builder supports creating custom images only in the same resource group as the source-managed image. In the following example, update the resource group name to be the same resource group as your source-managed image.

Azure CLI

```
# Resource group name - ibLinuxGalleryRG in this example
sigResourceGroup=ibLinuxGalleryRG
# Datacenter location - West US 2 in this example
location=westus2
# Additional region to replicate the image to - East US in this example
additionalregion=eastus
# Name of the Azure Compute Gallery - myGallery in this example
sigName=myIbGallery
# Name of the image definition to be created - myImageDef in this example
imageDefName=myIbImageDef
# Reference name in the image distribution metadata
runOutputName=aibLinuxSIG
```

Create a variable for your subscription ID:

Azure CLI

```
subscriptionID=$(az account show --query id --output tsv)
```

Create the resource group:

Azure CLI

```
az group create -n $sigResourceGroup -l $location
```

Create a user-assigned identity and set permissions on the resource group

VM Image Builder uses the provided user-identity to inject the image into an Azure Compute Gallery. In this example, you create an Azure role definition with specific actions for distributing the image. The role definition is then assigned to the user identity.

The success of the following command relies on RBAC propagation, which can take up to ~30 minutes in some cases. Wait about 5 minutes before running the command and retry after a delay if an error occurs.

Azure CLI

```
# Create user-assigned identity for VM Image Builder to access the storage account
where the script is stored
identityName=aibBuiUserId$(date +'%s')
az identity create -g $sigResourceGroup -n $identityName

# Get the identity ID
imgBuilderCliId=$(az identity show -g $sigResourceGroup -n $identityName --query
clientId -o tsv)

# Get the user identity URI that's needed for the template
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$sigResourceGroup/provid
ers/Microsoft.ManagedIdentity/userAssignedIdentities/$identityName

# Download an Azure role-definition template, and update the template with the
parameters that were specified earlier
curl
https://raw.githubusercontent.com/Azure/azvmimagebuilder/master/solutions/12_Creati
ng_AIB_Security_Roles/aibRoleImageCreation.json -o aibRoleImageCreation.json

imageRoleDefName="Azure Image Builder Image Def"$(date +'%s')

# Update the definition
sed -i -e "s/<subscriptionID>/$subscriptionID/g" aibRoleImageCreation.json
sed -i -e "s/<rgName>/$sigResourceGroup/g" aibRoleImageCreation.json
sed -i -e "s/Azure Image Builder Service Image Creation Role/$imageRoleDefName/g"
aibRoleImageCreation.json

# Create role definitions
az role definition create --role-definition ./aibRoleImageCreation.json

# Grant a role definition to the user-assigned identity. The success of the
following command relies on RBAC propagation, which can take up to ~30 minutes.
Wait about 5 minutes before running the command and wait a few more minutes before
retrying if you get an error.

az role assignment create \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope /subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup
```

Create an image definition and gallery

To use VM Image Builder with Azure Compute Gallery, you need to have an existing gallery and image definition. VM Image Builder doesn't create the gallery and image definition for you.

If you don't already have a gallery and image definition to use, start by creating them.

First, create a gallery:

Azure CLI

```
az sig create \
    -g $sigResourceGroup \
    --gallery-name $sigName
```

Then, create an image definition:

Azure CLI

```
az sig image-definition create \
   -g $sigResourceGroup \
   --gallery-name $sigName \
   --gallery-image-definition $imageDefName \
   --publisher myIbPublisher \
   --offer myOffer \
   --sku 20_04-lts-gen2 \
   --os-type Linux \
   --hyper-v-generation V2 \
   --features SecurityType=TrustedLaunchSupported
```

Download and configure the JSON file

Download the JSON template and configure it with your variables:

Azure CLI

```
curl
https://raw.githubusercontent.com/Azure/azvmimagebuilder/master/quickquickstarts/1_
Creating_a_Custom_Linux_Shared_Image_Gallery_Image/helloImageTemplateforSIG.json -o
helloImageTemplateforSIG.json
sed -i -e "s/<subscriptionID>/$subscriptionID/g" helloImageTemplateforSIG.json
sed -i -e "s/<rgName>/$sigResourceGroup/g" helloImageTemplateforSIG.json
sed -i -e "s/<imageDefName>/$imageDefName/g" helloImageTemplateforSIG.json
sed -i -e "s/<sharedImageGalName>/$sigName/g" helloImageTemplateforSIG.json
sed -i -e "s/<region1>/$location/g" helloImageTemplateforSIG.json
sed -i -e "s/<region2>/$additionalregion/g" helloImageTemplateforSIG.json
sed -i -e "s/<runOutputName>/$runOutputName/g" helloImageTemplateforSIG.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g" helloImageTemplateforSIG.json
sed -i -e "s/sudo mkdir/sleep 300;sudo mkdir/g" helloImageTemplateforSIG.json
```

Create the image version

In this section you create the image version in the gallery.

Submit the image configuration to the Azure VM Image Builder service:

Azure CLI

```
az resource create \
    --resource-group $sigResourceGroup \
    --properties @helloImageTemplateforSIG.json \
    --is-full-object \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateforSIG01
```

Start the image build:

Azure CLI

```
az resource invoke-action \
     --resource-group $sigResourceGroup \
     --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
     -n helloImageTemplateforSIG01 \
     --action Run
```

It can take a few moments to create the image and replicate it to both regions. Wait until this part is finished before you move on to create a VM.

Create the VM

Create the VM from the image version that was created by VM Image Builder.

Azure CLI

```
az vm create \
  --resource-group $sigResourceGroup \
  --name myAibGalleryVM \
  --admin-username aibuser \
  --location $location \
  --image
"/subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup/providers/Microsof
t.Compute/galleries/$sigName/images/$imageDefName/versions/latest" \
  --security-type TrustedLaunch \
  --generate-ssh-keys
```

Connect to the VM via Secure Shell (SSH):

Azure CLI

```
ssh aibuser@<publicIpAddress>
```

As soon as your SSH connection is established, you should see that the image was customized with a *Message of the Day*:

Console

```
**************************************************
**            This VM was built from the:       **
**      !! AZURE VM IMAGE BUILDER Custom Image !!  **
**         You have just been Customized :-)      **
**************************************************
```

Clean up your resources

> **Note**
>
> If you now want to try to recustomize the image version to create a new version of the same image, *skip the step outlined here* and go to Use VM Image Builder to create another image version.

If you no longer need the resources that were created as you followed the process in this article, you can delete them by doing the following.

This process deletes both the image that you created and all the other resource files. Make sure that you've finished this deployment before you delete the resources.

When you're deleting gallery resources, you need delete all the image versions before you can delete the image definition that was used to create them. To delete a gallery, you first need to have deleted all the image definitions in the gallery.

1. Delete the VM Image Builder template.

   Azure CLI

   ```azurecli
   az resource delete \
       --resource-group $sigResourceGroup \
       --resource-type Microsoft.VirtualMachineImages/imageTemplates \
       -n helloImageTemplateforSIG01
   ```

2. Delete permissions assignments, roles, and identity.

   Azure CLI

   ```azurecli
   az role assignment delete \
       --assignee $imgBuilderCliId \
       --role "$imageRoleDefName" \
       --scope /subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup

   az role definition delete --name "$imageRoleDefName"
   ```

```
az identity delete --ids $imgBuilderId
```

3. Get the image version that was created by VM Image Builder (it always starts with `0.`), and then delete it.

Azure CLI

```
sigDefImgVersion=$(az sig image-version list \
-g $sigResourceGroup \
--gallery-name $sigName \
--gallery-image-definition $imageDefName \
--subscription $subscriptionID --query [].'name' -o json | grep 0. | tr -d
'"')
az sig image-version delete \
-g $sigResourceGroup \
--gallery-image-version $sigDefImgVersion \
--gallery-name $sigName \
--gallery-image-definition $imageDefName \
--subscription $subscriptionID
```

4. Delete the image definition.

Azure CLI

```
az sig image-definition delete \
-g $sigResourceGroup \
--gallery-name $sigName \
--gallery-image-definition $imageDefName \
--subscription $subscriptionID
```

5. Delete the gallery.

Azure CLI

```
az sig delete -r $sigName -g $sigResourceGroup
```

6. Delete the resource group.

Azure CLI

```
az group delete -n $sigResourceGroup -y
```

Next steps

Learn more about Azure Compute Gallery.

Last updated on 03/20/2026

Create a Windows VM by using Azure VM Image Builder

Article • 10/28/2024

Applies to: ✔ Windows VMs

In this article, you learn how to create a customized Windows image by using Azure VM Image Builder. The example in this article uses customizers for customizing the image:

- PowerShell (ScriptUri): Download and run a PowerShell script.
- Windows Restart: Restarts the VM.
- PowerShell (inline): Runs a specific command. In this example, it creates a directory on the VM by using `mkdir c:\\buildActions`.
- File: Copies a file from GitHub to the VM. This example copies index.md to `c:\buildArtifacts\index.html` on the VM.
- `buildTimeoutInMinutes`: Specifies a build time, in minutes. The default is 240 minutes, which you can increase to allow for longer-running builds. The minimum allowed value is 6 minutes. Values shorter than 6 minutes will cause errors.
- `vmProfile`: Specifies a `vmSize` and network properties.
- `osDiskSizeGB`: Can be used to increase the size of an image.
- `identity`. Provides an identity for VM Image Builder to use during the build.

Use the following sample JSON template to configure the image: helloImageTemplateWin.json.

> **Note**
>
> Windows users can run the following Azure CLI examples on Azure Cloud Shell by using Bash.

Register the providers

To use VM Image Builder, you need to register the feature. Check your registration by running the following commands:

Azure CLI

```
az provider show -n Microsoft.VirtualMachineImages | grep registrationState
az provider show -n Microsoft.KeyVault | grep registrationState
az provider show -n Microsoft.Compute | grep registrationState
```

```
az provider show -n Microsoft.Storage | grep registrationState
az provider show -n Microsoft.Network | grep registrationState
az provider show -n Microsoft.ContainerInstance -o json | grep
registrationState
```

If the output doesn't say *registered*, run the following commands:

Azure CLI

```
az provider register -n Microsoft.VirtualMachineImages
az provider register -n Microsoft.Compute
az provider register -n Microsoft.KeyVault
az provider register -n Microsoft.Storage
az provider register -n Microsoft.Network
az provider register -n Microsoft.ContainerInstance
```

Set variables

Because you'll be using some pieces of information repeatedly, create some variables to store that information:

Azure CLI

```
# Resource group name - we're using myImageBuilderRG in this example
imageResourceGroup='myWinImgBuilderRG'
# Region location
location='WestUS2'
# Run output name
runOutputName='aibWindows'
# The name of the image to be created
imageName='aibWinImage'
```

Create a variable for your subscription ID:

Azure CLI

```
subscriptionID=$(az account show --query id --output tsv)
```

Create the resource group

To store the image configuration template artifact and the image, use the following resource group:

Azure CLI

```
az group create -n $imageResourceGroup -l $location
```

Create a user-assigned identity and set permissions on the resource group

VM Image Builder uses the provided user-identity to inject the image into the resource group. In this example, you create an Azure role definition with specific permissions for distributing the image. The role definition is then assigned to the user identity.

Create a user-assigned managed identity and grant permissions

Create a user-assigned identity so that VM Image Builder can access the storage account where the script is stored.

Azure CLI

```
identityName=aibBuiUserId$(date +'%s')
az identity create -g $imageResourceGroup -n $identityName

# Get the identity ID
imgBuilderCliId=$(az identity show -g $imageResourceGroup -n $identityName -
-query clientId -o tsv)

# Get the user identity URI that's needed for the template
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$imageResourceGro
up/providers/Microsoft.ManagedIdentity/userAssignedIdentities/$identityName

# Download the preconfigured role definition example
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12
_Creating_AIB_Security_Roles/aibRoleImageCreation.json -o
aibRoleImageCreation.json

imageRoleDefName="Azure Image Builder Image Def"$(date +'%s')

# Update the definition
sed -i -e "s%<subscriptionID>%$subscriptionID%g" aibRoleImageCreation.json
sed -i -e "s%<rgName>%$imageResourceGroup%g" aibRoleImageCreation.json
sed -i -e "s%Azure Image Builder Service Image Creation
Role%$imageRoleDefName%g" aibRoleImageCreation.json

# Create role definitions
az role definition create --role-definition ./aibRoleImageCreation.json

# Grant a role definition to the user-assigned identity
```

```
az role assignment create \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup
```

Download the image configuration template

We've created a parameterized image configuration template for you to try. Download the example JSON file, and then configure it with the variables that you set earlier.

Azure CLI

```
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickquickstarts/0_Creating_a_Custom_Windows_Managed_Image/helloImageTemplateWin.json -o
helloImageTemplateWin.json

sed -i -e "s%<subscriptionID>%$subscriptionID%g" helloImageTemplateWin.json
sed -i -e "s%<rgName>%$imageResourceGroup%g" helloImageTemplateWin.json
sed -i -e "s%<region>%$location%g" helloImageTemplateWin.json
sed -i -e "s%<imageName>%$imageName%g" helloImageTemplateWin.json
sed -i -e "s%<runOutputName>%$runOutputName%g" helloImageTemplateWin.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g" helloImageTemplateWin.json
```

You can modify this example in the terminal by using a text editor such as `vi`.

Azure CLI

```
vi helloImageTemplateWin.json
```

> **Note**
>
> For the source image, always specify a version. You can't specify `latest` as the version.
>
> If you add or change the resource group that the image is distributed to, make sure that the permissions are set on the resource group.

Create the image

Submit the image configuration to the VM Image Builder service by running the following commands:

Azure CLI

```
az resource create \
    --resource-group $imageResourceGroup \
    --properties @helloImageTemplateWin.json \
    --is-full-object \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateWin01
```

When you're done, a success message is returned to the console, and a VM Image Builder configuration template is created in the `$imageResourceGroup`.

In the background, VM Image Builder also creates a staging resource group in your subscription. This resource group is used to build the image in the following format: `IT_<DestinationResourceGroup>_<TemplateName>`.

> **Note**
>
> If deletion is required, don't delete the staging resource group directly. First, delete the image template artifact, which subsequently will delete the staging resource group. Otherwise, the image template may enter a failed state.

If the service reports a failure when you submit the image configuration template, do the following:

- See Troubleshoot the Azure VM Image Builder service.
- Before you try to resubmit the template, delete it by running the following commands:

Azure CLI

```
az resource delete \
    --resource-group $imageResourceGroup \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateWin01
```

Start the image build

Start the image-building process by using az resource invoke-action.

Azure CLI

```
az resource invoke-action \
     --resource-group $imageResourceGroup \
```

```
    --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateWin01 \
    --action Run
```

Wait until the build is complete.

If you encounter any errors, see Troubleshoot the Azure VM Image Builder service.

Create the VM

Create the VM by using the image that you built. In the following code, replace *<password>* with your own password for the *aibuser* on the VM.

Azure CLI

```
az vm create \
  --resource-group $imageResourceGroup \
  --name aibImgWinVm00 \
  --admin-username aibuser \
  --admin-password <password> \
  --image $imageName \
  --location $location
```

Verify the customization

Create a Remote Desktop connection to the VM by using the username and password that you set when you created the VM. In the VM, open a Command Prompt window, and then type:

Console

```
dir c:\
```

The following two directories are created during the image customization:

- buildActions
- buildArtifacts

Clean up your resources

When you're done, delete the resources you've created.

1. Delete the VM Image Builder template.

Azure CLI

```
az resource delete \
    --resource-group $imageResourceGroup \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateWin01
```

2. Delete the role assignment, role definition, and user identity.

Azure CLI

```
az role assignment delete \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup

az role definition delete --name "$imageRoleDefName"

az identity delete --ids $imgBuilderId
```

3. Delete the image resource group.

Azure CLI

```
az group delete -n $imageResourceGroup
```

Next steps

To learn more about the components of the JSON file that this article uses, see the VM Image Builder template reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a Windows VM with VM Image Builder by using PowerShell

Article • 08/22/2024

Applies to: ✔ Windows VMs

This article demonstrates how to create a customized Windows VM image by using the Azure VM Image Builder PowerShell module.

Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

If you choose to use PowerShell locally, this article requires that you install the Azure PowerShell module and connect to your Azure account by using the Connect-AzAccount cmdlet. For more information, see Install Azure PowerShell.

Some of the steps require cmdlets from the Az.ImageBuilder module. Install separately by using the following command.

Azure PowerShell

```
Install-Module -Name Az.ImageBuilder
```

Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article, without having to install anything on your local environment.

To start Azure Cloud Shell:

Expand table

Option	Example/Link
Select **Try It** in the upper-right corner of a code or command block. Selecting **Try It** doesn't automatically copy the code or command to Cloud Shell.	

Option	Example/Link
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To use Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block (or command block) to copy the code or command.
3. Paste the code or command into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux, or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code or command.

If you have multiple Azure subscriptions, choose the appropriate subscription in which the resources should be billed. Select a specific subscription by using the Set-AzContext cmdlet.

Azure PowerShell

```
Set-AzContext -SubscriptionId 00000000-0000-0000-0000-000000000000
```

Register providers

If you haven't already done so, register the following resource providers to use with your Azure subscription:

- Microsoft.Compute
- Microsoft.KeyVault
- Microsoft.Storage
- Microsoft.Network
- Microsoft.VirtualMachineImages
- Microsoft.ManagedIdentity
- Microsoft.ContainerInstance

Azure PowerShell

```
Get-AzResourceProvider -ProviderNamespace Microsoft.Compute,
Microsoft.KeyVault, Microsoft.Storage, Microsoft.VirtualMachineImages,
```

```
Microsoft.Network, Microsoft.ManagedIdentity |
  Where-Object RegistrationState -ne Registered |
    Register-AzResourceProvider
```

Define variables

Because you'll be using some pieces of information repeatedly, create some variables to store that information:

Azure PowerShell

```
# Destination image resource group name
$imageResourceGroup = 'myWinImgBuilderRG'

# Azure region
$location = 'WestUS2'

# Name of the image to be created
$imageTemplateName = 'myWinImage'

# Distribution properties of the managed image upon completion
$runOutputName = 'myDistResults'
```

Create a variable for your Azure subscription ID. To confirm that the `subscriptionID` variable contains your subscription ID, you can run the second line in the following example:

Azure PowerShell

```
# Your Azure Subscription ID
$subscriptionID = (Get-AzContext).Subscription.Id
Write-Output $subscriptionID
```

Create a resource group

Create an Azure resource group by using the New-AzResourceGroup cmdlet. A resource group is a logical container in which Azure resources are deployed and managed as a group.

The following example creates a resource group that's based on the name in the `$imageResourceGroup` variable in the region that you've specified in the `$location` variable. This resource group is used to store the image configuration template artifact and the image.

Azure PowerShell

```
New-AzResourceGroup -Name $imageResourceGroup -Location $location
```

Create a user identity and set role permissions

Grant Azure image builder permissions to create images in the specified resource group by using the following example. Without this permission, the image build process won't finish successfully.

1. Create variables for the role definition and identity names. These values must be unique.

   Azure PowerShell

   ```
   [int]$timeInt = $(Get-Date -UFormat '%s')
   $imageRoleDefName = "Azure Image Builder Image Def $timeInt"
   $identityName = "myIdentity$timeInt"
   ```

2. Create a user identity.

   Azure PowerShell

   ```
   New-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -Name
   $identityName -Location $location
   ```

3. Store the identity resource and principal IDs in variables.

   Azure PowerShell

   ```
   $identityNameResourceId = (Get-AzUserAssignedIdentity -
   ResourceGroupName $imageResourceGroup -Name $identityName).Id
   $identityNamePrincipalId = (Get-AzUserAssignedIdentity -
   ResourceGroupName $imageResourceGroup -Name $identityName).PrincipalId
   ```

Assign permissions for the identity to distribute the images

1. Download the JSON configuration file, and then modify it based on the settings that are defined in this article.

   Azure PowerShell

```
$myRoleImageCreationUrl =
'https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12_Creating_AIB_Security_Roles/aibRoleImageCreation.json'
$myRoleImageCreationPath = "myRoleImageCreation.json"

Invoke-WebRequest -Uri $myRoleImageCreationUrl -OutFile
$myRoleImageCreationPath -UseBasicParsing

$Content = Get-Content -Path $myRoleImageCreationPath -Raw
$Content = $Content -replace '<subscriptionID>', $subscriptionID
$Content = $Content -replace '<rgName>', $imageResourceGroup
$Content = $Content -replace 'Azure Image Builder Service Image
Creation Role', $imageRoleDefName
$Content | Out-File -FilePath $myRoleImageCreationPath -Force
```

2. Create the role definition.

Azure PowerShell

```
New-AzRoleDefinition -InputFile $myRoleImageCreationPath
```

3. Grant the role definition to the VM Image Builder service principal.

Azure PowerShell

```
$RoleAssignParams = @{
  ObjectId = $identityNamePrincipalId
  RoleDefinitionName = $imageRoleDefName
  Scope =
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
}
New-AzRoleAssignment @RoleAssignParams
```

> **Note**
>
> If you receive the error "New-AzRoleDefinition: Role definition limit exceeded. No more role definitions can be created," see Troubleshoot Azure RBAC (role-based access control).

Create an Azure Compute Gallery

1. Create the gallery.

Azure PowerShell

```
$myGalleryName = 'myImageGallery'
$imageDefName = 'winSvrImages'

New-AzGallery -GalleryName $myGalleryName -ResourceGroupName
$imageResourceGroup -Location $location
```

2. Create a gallery definition.

Azure PowerShell

```
$GalleryParams = @{
  GalleryName = $myGalleryName
  ResourceGroupName = $imageResourceGroup
  Location = $location
  Name = $imageDefName
  OsState = 'generalized'
  OsType = 'Windows'
  Publisher = 'myCo'
  Offer = 'Windows'
  Sku = 'Win2019'
}
New-AzGalleryImageDefinition @GalleryParams
```

Create an image

1. Create a VM Image Builder source object. For valid parameter values, see Find Windows VM images in Azure Marketplace with Azure PowerShell.

Azure PowerShell

```
$SrcObjParams = @{
  PlatformImageSource = $true
  Publisher = 'MicrosoftWindowsServer'
  Offer = 'WindowsServer'
  Sku = '2019-Datacenter'
  Version = 'latest'
}
$srcPlatform = New-AzImageBuilderTemplateSourceObject @SrcObjParams
```

2. Create a VM Image Builder distributor object.

Azure PowerShell

```
$disObjParams = @{
  SharedImageDistributor = $true
  ArtifactTag = @{tag='dis-share'}
  GalleryImageId =
```

```
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup/prov
iders/Microsoft.Compute/galleries/$myGalleryName/images/$imageDefName"
  ReplicationRegion = $location
  RunOutputName = $runOutputName
  ExcludeFromLatest = $false
}
$disSharedImg = New-AzImageBuilderTemplateDistributorObject
@disObjParams
```

3. Create a VM Image Builder customization object.

Azure PowerShell

```
$ImgCustomParams01 = @{
  PowerShellCustomizer = $true
  Name = 'settingUpMgmtAgtPath'
  RunElevated = $false
  Inline = @("mkdir c:\\buildActions", "mkdir c:\\buildArtifacts",
"echo Azure-Image-Builder-Was-Here  >
c:\\buildActions\\buildActionsOutput.txt")
}
$Customizer01 = New-AzImageBuilderTemplateCustomizerObject
@ImgCustomParams01
```

4. Create a second VM Image Builder customization object.

Azure PowerShell

```
$ImgCustomParams02 = @{
  FileCustomizer = $true
  Name = 'downloadBuildArtifacts'
  Destination = 'c:\\buildArtifacts\\index.html'
  SourceUri =
'https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickq
uickstarts/exampleArtifacts/buildArtifacts/index.html'
}
$Customizer02 = New-AzImageBuilderTemplateCustomizerObject
@ImgCustomParams02
```

5. Create a VM Image Builder template.

Azure PowerShell

```
$ImgTemplateParams = @{
  ImageTemplateName = $imageTemplateName
  ResourceGroupName = $imageResourceGroup
  Source = $srcPlatform
  Distribute = $disSharedImg
  Customize = $Customizer01, $Customizer02
  Location = $location
```

```
  UserAssignedIdentityId = $identityNameResourceId
}
New-AzImageBuilderTemplate @ImgTemplateParams
```

When the template has been created, a message is returned, and a VM Image Builder configuration template is created in `$imageResourceGroup`.

To determine whether the template creation process was successful, use the following example:

Azure PowerShell

```
Get-AzImageBuilderTemplate -ImageTemplateName $imageTemplateName -
ResourceGroupName $imageResourceGroup |
  Select-Object -Property Name, LastRunStatusRunState, LastRunStatusMessage,
ProvisioningState
```

In the background, VM Image Builder also creates a staging resource group in your subscription. This resource group is used for the image build. It's in the format `IT_<DestinationResourceGroup>_<TemplateName>`.

> ⚠ **Warning**
>
> Don't delete the staging resource group directly. To cause the staging resource group to be deleted, delete the image template artifact.

If the service reports a failure when the image configuration template is submitted, do the following:

- See Troubleshoot Azure VM Image Builder failures.
- Before you retry submitting the template, delete it by following this example:

  Azure PowerShell

  ```
  Remove-AzImageBuilderTemplate -ImageTemplateName $imageTemplateName -
  ResourceGroupName $imageResourceGroup
  ```

Start the image build

Submit the image configuration to the VM Image Builder service by running the following command:

Azure PowerShell

```
Start-AzImageBuilderTemplate -ResourceGroupName $imageResourceGroup -Name
$imageTemplateName
```

Wait for the image building process to finish, which could take up to an hour.

If you encounter errors, review Troubleshoot Azure VM Image Builder failures.

Create a VM

1. Store the VM login credentials in a variable. The password must be complex.

   Azure PowerShell

   ```
   $Cred = Get-Credential
   ```

2. Create the VM by using the image you created.

   Azure PowerShell

   ```
   $ArtifactId = (Get-AzImageBuilderTemplateRunOutput -ImageTemplateName
   $imageTemplateName -ResourceGroupName $imageResourceGroup).ArtifactId

   New-AzVM -ResourceGroupName $imageResourceGroup -Image $ArtifactId -
   Name myWinVM01 -Credential $Cred
   ```

Verify the customizations

1. Create a Remote Desktop connection to the VM by using the username and password that you set when you created the VM.

2. Inside the VM, open PowerShell and run `Get-Content`, as shown in the following example:

   Azure PowerShell

   ```
   Get-Content -Path C:\buildActions\buildActionsOutput.txt
   ```

   The output is based on the contents of the file that you created during the image customization process.

   Output

```
Azure-Image-Builder-Was-Here
```

3. From the same PowerShell session, verify that the second customization finished successfully by checking for the presence of `c:\buildArtifacts\index.html`, as shown in the following example:

Azure PowerShell

```
Get-ChildItem c:\buildArtifacts\
```

The result should be a directory listing showing that the file was downloaded during the image customization process.

Output

```
    Directory: C:\buildArtifacts

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          29/01/2021    10:04            276 index.html
```

Clean up your resources

If you no longer need the resources that were created during this process, you can delete them by doing the following:

1. Delete the VM Image Builder template.

Azure PowerShell

```
Remove-AzImageBuilderTemplate -ResourceGroupName $imageResourceGroup -
Name $imageTemplateName
```

2. Delete the image resource group.

> **⊗ Caution**
>
> The following example deletes the specified resource group and all the resources that it contains. If any resources outside the scope of this article exist in the resource group, they'll also be deleted.

```
```

Azure PowerShell

```
Remove-AzResourceGroup -Name $imageResourceGroup
```

Next steps

To learn more about the components of the JSON file that this article uses, see the VM Image Builder template reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an Azure Virtual Desktop image by using VM Image Builder and PowerShell

Article • 08/22/2024

Applies to: ✔ Windows VMs

In this article, you learn how to create an Azure Virtual Desktop image with these customizations:

- FSLogix setup
- Azure Virtual Desktop optimization
- Microsoft Teams installation
- Windows Restart customizer
- Windows Update customizer

The article discusses how to automate the customizations by using Azure VM Image Builder. You can then distribute the image to an Azure Compute Gallery (formerly Shared Image Gallery), where you can replicate it to other regions, control the scale, and share the image within and beyond your organization.

To simplify deploying a VM Image Builder configuration, our example uses an Azure Resource Manager template with the VM Image Builder template nested within it. This approach gives you a few more benefits, such as variables and parameter inputs. You can also pass parameters from the command line.

This article is intended as a copy-and-paste exercise.

> **Note**
>
> You'll find the scripts for installing the apps on GitHub. They're for illustration and testing purposes only. Do not use them for production workloads.

Tips for building Windows images

- VM size: For Windows, use `Standard_D2_v2` or greater. The default size is `Standard_D1_v2`, which isn't suitable for Windows.

- This article uses PowerShell customizer scripts. Use the following settings, or the build will stop responding:

  ```JSON
    "runElevated": true,
    "runAsSystem": true,
  ```

  For example:

  ```JSON
    {
        "type": "PowerShell",
        "name": "installFSLogix",
        "runElevated": true,
        "runAsSystem": true,
        "scriptUri":
  "https://raw.githubusercontent.com/azure/azvmimagebuilder/main/solution
  s/14_Building_Images_WVD/0_installConfFsLogix.ps1"
  ```

- Comment your code: The VM Image Builder build log, *customization.log*, is verbose. If you comment your scripts by using 'write-host', they'll be sent to the logs, which should make troubleshooting easier.

  ```PowerShell
  write-host 'AIB Customization: Starting OS Optimizations script'
  ```

- Exit codes: VM Image Builder expects all scripts to return a `0` exit code. If you use a non-zero exit code, VM Image Builder fails the customization and stops the build. If you have complex scripts, add instrumentation and emit exit codes, which will be shown in the *customization.log* file.

  ```PowerShell
  Write-Host "Exit code: " $LASTEXITCODE
  ```

- Test: Test and retest your code on a standalone VM. Ensure that there are no user prompts, that you're using the correct privileges, and so on.

- Networking: `Set-NetAdapterAdvancedProperty` is set in the optimization script but fails the VM Image Builder build. Because it disconnects the network, it's commented out. We're investigating this issue.

Prerequisites

You must have the latest Azure PowerShell cmdlets installed. For more information, see Overview of Azure PowerShell.

PowerShell

```
# Check to ensure that you're registered for the providers and
RegistrationState is set to 'Registered'
Get-AzResourceProvider -ProviderNamespace Microsoft.VirtualMachineImages
Get-AzResourceProvider -ProviderNamespace Microsoft.Storage
Get-AzResourceProvider -ProviderNamespace Microsoft.Compute
Get-AzResourceProvider -ProviderNamespace Microsoft.KeyVault
Get-AzResourceProvider -ProviderNamespace Microsoft.ContainerInstance

# If they don't show as 'Registered', run the following commented-out code

## Register-AzResourceProvider -ProviderNamespace
Microsoft.VirtualMachineImages
## Register-AzResourceProvider -ProviderNamespace Microsoft.Storage
## Register-AzResourceProvider -ProviderNamespace Microsoft.Compute
## Register-AzResourceProvider -ProviderNamespace Microsoft.KeyVault
## Register-AzResourceProvider -ProviderNamespace
Microsoft.ContainerInstance
```

Set up the environment and variables

Azure PowerShell

```
# Step 1: Import module
Import-Module Az.Accounts

# Step 2: get existing context
$currentAzContext = Get-AzContext

# Destination image resource group
$imageResourceGroup="avdImageDemoRg"

# Location (see possible locations in the main docs)
$location="westus2"

# Your subscription. This command gets your current subscription
$subscriptionID=$currentAzContext.Subscription.Id

# Image template name
$imageTemplateName="avd10ImageTemplate01"

# Distribution properties object name (runOutput). Gives you the properties
of the managed image on completion
$runOutputName="sigOutput"
```

```
# Create resource group
New-AzResourceGroup -Name $imageResourceGroup -Location $location
```

Permissions, user identity, and role

1. Create a user identity.

   Azure PowerShell

   ```
   # setup role def names, these need to be unique
   $timeInt=$(get-date -UFormat "%s")
   $imageRoleDefName="Azure Image Builder Image Def"+$timeInt
   $identityName="aibIdentity"+$timeInt

   ## Add Azure PowerShell modules to support AzUserAssignedIdentity and
   Azure VM Image Builder
   'Az.ImageBuilder', 'Az.ManagedServiceIdentity' | ForEach-Object
   {Install-Module -Name $_ -AllowPrerelease}

   # Create the identity
   New-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -Name
   $identityName -Location $location

   $identityNameResourceId=$(Get-AzUserAssignedIdentity -ResourceGroupName
   $imageResourceGroup -Name $identityName).Id
   $identityNamePrincipalId=$(Get-AzUserAssignedIdentity -
   ResourceGroupName $imageResourceGroup -Name $identityName).PrincipalId
   ```

2. Assign permissions to the identity to distribute images. The following commands download and update the template with the previously specified parameters.

   Azure PowerShell

   ```
   $aibRoleImageCreationUrl="https://raw.githubusercontent.com/azure/azvmi
   magebuilder/main/solutions/12_Creating_AIB_Security_Roles/aibRoleImageC
   reation.json"
   $aibRoleImageCreationPath = "aibRoleImageCreation.json"

   # Download the config
   Invoke-WebRequest -Uri $aibRoleImageCreationUrl -OutFile
   $aibRoleImageCreationPath -UseBasicParsing

   ((Get-Content -path $aibRoleImageCreationPath -Raw) -replace
   '<subscriptionID>',$subscriptionID) | Set-Content -Path
   $aibRoleImageCreationPath
   ((Get-Content -path $aibRoleImageCreationPath -Raw) -replace
   '<rgName>', $imageResourceGroup) | Set-Content -Path
   $aibRoleImageCreationPath
   ```

```
((Get-Content -path $aibRoleImageCreationPath -Raw) -replace 'Azure
Image Builder Service Image Creation Role', $imageRoleDefName) | Set-
Content -Path $aibRoleImageCreationPath

# Create a role definition
New-AzRoleDefinition -InputFile  ./aibRoleImageCreation.json

# Grant the role definition to the VM Image Builder service principal
New-AzRoleAssignment -ObjectId $identityNamePrincipalId -
RoleDefinitionName $imageRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
```

> ⓘ **Note**
>
> If you receive the error "New-AzRoleDefinition: Role definition limit exceeded. No more role definitions can be created," see Troubleshoot Azure RBAC (role-based access control).

Create an Azure Compute Gallery

If you don't already have an Azure Compute Gallery, you need to create one.

Azure PowerShell

```
$sigGalleryName= "myaibsig01"
$imageDefName ="win10avd"

# Create the gallery
New-AzGallery -GalleryName $sigGalleryName -ResourceGroupName
$imageResourceGroup  -Location $location

# Create the gallery definition
New-AzGalleryImageDefinition -GalleryName $sigGalleryName -ResourceGroupName
$imageResourceGroup -Location $location -Name $imageDefName -OsState
generalized -OsType Windows -Publisher 'myCo' -Offer 'Windows' -Sku '10avd'
```

Configure the VM Image Builder template

For this example, we've prepared a template that downloads and updates the VM Image Builder template with the parameters that were specified earlier. The template installs FSLogix, operating system optimizations, and Microsoft Teams, and it runs Windows Update at the end.

If you open the template, you can see in the source property the image that's being used. In this example, it uses a Windows 10 multi-session image.

Windows 10 images

You should be aware of two key types of images: multi-session and single-session.

Multi-session images are intended for pooled usage. Here's an example of the image details in Azure:

JSON

```
"publisher": "MicrosoftWindowsDesktop",
"offer": "Windows-10",
"sku": "20h2-avd",
"version": "latest"
```

Single-session images are intended for individual usage. Here's an example of the image details in Azure:

JSON

```
"publisher": "MicrosoftWindowsDesktop",
"offer": "Windows-10",
"sku": "19h2-ent",
"version": "latest"
```

You can also change which Windows 10 images are available:

Azure PowerShell

```
Get-AzVMImageSku -Location westus2 -PublisherName MicrosoftWindowsDesktop -
Offer windows-10
```

Download and configure the template

Now, download the template and configure it for your own use.

Azure PowerShell

```
$templateUrl="https://raw.githubusercontent.com/azure/azvmimagebuilder/main/
solutions/14_Building_Images_WVD/armTemplateWVD.json"
$templateFilePath = "armTemplateWVD.json"

Invoke-WebRequest -Uri $templateUrl -OutFile $templateFilePath -
```

```
UseBasicParsing

((Get-Content -path $templateFilePath -Raw) -replace
'<subscriptionID>',$subscriptionID) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace
'<rgName>',$imageResourceGroup) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<region>',$location) |
Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace
'<runOutputName>',$runOutputName) | Set-Content -Path $templateFilePath

((Get-Content -path $templateFilePath -Raw) -replace
'<imageDefName>',$imageDefName) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace
'<sharedImageGalName>',$sigGalleryName) | Set-Content -Path
$templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<region1>',$location)
| Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace
'<imgBuilderId>',$identityNameResourceId) | Set-Content -Path
$templateFilePath
```

Feel free to view the template. All the code is viewable.

Submit the template

Your template must be submitted to the service. Doing so downloads any dependent artifacts, such as scripts, and validates, checks permissions, and stores them in the staging resource group, which is prefixed with *IT_*.

Azure PowerShell

```
New-AzResourceGroupDeployment -ResourceGroupName $imageResourceGroup -
TemplateFile $templateFilePath -TemplateParameterObject @{"api-Version" =
"2020-02-14"; "imageTemplateName" = $imageTemplateName; "svclocation" =
$location}

# Optional - if you have any errors running the preceding command, run:
$getStatus=$(Get-AzImageBuilderTemplate -ResourceGroupName
$imageResourceGroup -Name $imageTemplateName)
$getStatus.ProvisioningErrorCode
$getStatus.ProvisioningErrorMessage
```

Build the image

Azure PowerShell

```
Start-AzImageBuilderTemplate -ResourceGroupName $imageResourceGroup -Name
$imageTemplateName -NoWait
```

> **Note**
>
> The command doesn't wait for the VM Image Builder service to complete the image build, so you can query the status as shown here.

Azure PowerShell

```
$getStatus=$(Get-AzImageBuilderTemplate -ResourceGroupName
$imageResourceGroup -Name $imageTemplateName)

# Shows all the properties
$getStatus | Format-List -Property *

# Shows the status of the build
$getStatus.LastRunStatusRunState
$getStatus.LastRunStatusMessage
$getStatus.LastRunStatusRunSubState
```

Create a VM

Now that the image is built, you can build a VM from it. Use the examples from New-AzVM (Az PowerShell module.Compute).

Clean up your resources

If you no longer need the resources that were created during this process, you can delete them by doing the following:

> **Important**
>
> Delete the resource group template first. If you delete only the resource group, the staging resource group (*IT_*) that's used by VM Image Builder won't be cleaned up.

1. Remove the VM Image Builder template.

   Azure PowerShell

```
Remove-AzImageBuilderTemplate -ResourceGroupName $imageResourceGroup -
Name vd10ImageTemplate
```

2. Delete the role assignment.

Azure PowerShell

```
Remove-AzRoleAssignment -ObjectId $identityNamePrincipalId -
RoleDefinitionName $imageRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"

## Remove the definitions
Remove-AzRoleDefinition -Name "$identityNamePrincipalId" -Force -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"

## Delete the identity
Remove-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -
Name $identityName -Force
```

3. Delete the resource group.

Azure PowerShell

```
Remove-AzResourceGroup $imageResourceGroup -Force
```

Next steps

To try more VM Image Builder examples, go to GitHub.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Isolated Image Builds for Azure VM Image Builder

Isolated Image Builds is a feature of Azure VM Image Builder (AIB). It transitions the core process of VM image customization/validation from shared platform infrastructure to dedicated Azure Container Instances (ACI) resources in your subscription, providing compute and network isolation.

Advantages of Isolated Image Builds

Isolated Image Builds enables defense-in-depth by limiting network access of your build VM to your subscription. Isolated Image Builds also provides more transparency by allowing you to inspect processing performed by AIB to customize and validate your VM image. Further, Isolated Image Builds simplifies viewing live build logs. Specifically:

1. **Compute Isolation:** Isolated Image Builds performs a major portion of image-building processing in ACI resources in your subscription instead of on AIB's shared platform resources. ACI provides hypervisor isolation for each container group to ensure containers run in isolation without sharing a kernel.

2. **Network Isolation:** Isolated Image Builds removes all direct network WinRM/SSH communication between your build VM and backend components of the AIB service.

    - If you're provisioning an AIB template without your own subnet for build VM, then a Public IP Address resource is no longer provisioned in your staging resource group at image build time.
    - If you're provisioning an AIB template with an existing subnet for build VM, then a Private Link-based communication channel is no longer set up between your build VM and AIB's backend platform resources. Instead, the communication channel is set up between the ACI and the build VM resources, both of which reside in the staging resource group in your subscription.
    - Starting with API version 2024-02-01, you can specify a second subnet for deploying the ACI in addition to the subnet for the build VM. If specified, AIB deploys the ACI on this subnet, and there is no need for AIB to set up the Private Link-based communication channel between the ACI and the build VM.

3. **Transparency:** AIB is built on HashiCorp Packer. Isolated Image Builds executes Packer in the ACI in your subscription, which allows you to inspect the ACI resource and its containers. Similarly, having the entire network communication pipeline in your subscription allows you to inspect all the network resources, their settings, and their allowances.

4. **Better viewing of live logs:** AIB writes customization logs to a storage account in the staging resource group in your subscription. Isolated Image Builds provides another way to follow the same logs directly in the Azure portal by navigating to AIB's container in the ACI resource.

> **Note**
>
> To access the live logs during the image build or the customization and validation log files after the build is complete, please refer to the troubleshooting guide.

For more information about networking topologies and connectivity behavior, see Azure VM Image Builder networking options.

Backward compatibility

Isolated Image Builds is a platform-level change and doesn't affect AIB's interfaces. So, your existing Image Template and Trigger resources continue to function and there's no change in the way you deploy new resources of these types. You need to create new templates or update existing templates if you want to use the network topology that allows you to bring your own ACI subnet.

Your image builds are automatically migrated to Isolated Image Builds and you need to take no action to opt in. Also, customization logs continue to be available in the storage account.

Depending on the network topology specified in the Image Template, you might observe a few new resources temporarily appear in the staging resource group (for example, ACI, Virtual Network, Network Security Group, and Private Endpoint), while some other resources no longer appear (for example, Public IP Address). As before, these temporary resources exist only during the build and AIB deletes them thereafter.

> **Important**
>
> Make sure your subscription is registered for `Microsoft.ContainerInstance provider`:
>
> - Azure CLI: `az provider register -n Microsoft.ContainerInstance`
> - PowerShell: `Register-AzResourceProvider -ProviderNamespace Microsoft.ContainerInstance`
>
> After successfully registering your subscription, make sure there are no Azure Policies in your subscription that deny deployment of ACI resources. Policies allowing only a restricted set of resource types not including ACI would cause Isolated Image Builds to fail.

Ensure that your subscription also has a sufficient quota of resources required for deployment of ACI resources.

> **Important**
>
> Depending on the network topology specified in the Image Template, AIB may need to deploy temporary networking-related resources in the staging resource group in your subscription. Ensure that no Azure Policies deny the deployment of such resources (Virtual Network with Subnets, Network Security Group, Private Endpoint) in the resource group.
>
> If you have Azure Policies applying DDoS protection plans to any newly created Virtual Network, either relax the Policy for the resource group or ensure that the Template Managed Identity has permissions to join the plan. Alternatively, you can use the network topology that does not require deployment of a new Virtual Network by AIB.

> **Important**
>
> Make sure you follow all best practices while using AIB.

> **Note**
>
> AIB is in the process of rolling this change out to all locations and customers. Some of these details (especially around deployment of new networking-related resources) might change as the process is fine-tuned based on service telemetry and feedback. For any errors, please refer to the troubleshooting guide.

Next steps

- Azure VM Image Builder overview
- Getting started with Azure Container Instances
- Securing your Azure resources
- Troubleshooting guide for Azure VM Image Builder

Note: The author created this article with assistance from AI. Learn more

Last updated on 03/28/2026

Azure Policy Regulatory Compliance controls for Azure VM Image Builder

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Regulatory Compliance in Azure Policy provides initiative definitions, known as *built-ins*, for the compliance domains and security controls related to different compliance standards. Microsoft creates and manages these built-ins.

In this article, you can refer to a list of the compliance domains and security controls for Azure VM Image Builder. You can assign the built-ins for a security control individually, to help make your Azure resources compliant with the specific standard.

The title of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Policy Version** column to view the source on the Azure Policy GitHub repo.

> **Important**
>
> Each control is associated with one or more **Azure Policy** definitions. These policies might help you **assess compliance** with the control. However, there often isn't a one-to-one or complete match between a control and one or more policies. As such, **Compliant** in Azure Policy refers only to the policies themselves. This doesn't ensure that you're fully compliant with all requirements of a control. In addition, the compliance standard includes controls that aren't addressed by any Azure Policy definitions at this time. Therefore, compliance in Azure Policy is only a partial view of your overall compliance status. The associations between controls and Azure Policy Regulatory Compliance definitions for these compliance standards can change over time.

FedRAMP High

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - FedRAMP High. For more information about this compliance standard, see FedRAMP High.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17	Remote Access	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	VM Image Builder templates should use private link	1.1.0
System And Communications Protection	SC-7	Boundary Protection	VM Image Builder templates should use private link	1.1.0
System And Communications Protection	SC-7 (3)	Access Points	VM Image Builder templates should use private link	1.1.0

FedRAMP Moderate

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - FedRAMP Moderate. For more information about this compliance standard, see FedRAMP Moderate.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17	Remote Access	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	VM Image Builder templates should use private link	1.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Communications Protection	SC-7	Boundary Protection	VM Image Builder templates should use private link	1.1.0
System And Communications Protection	SC-7 (3)	Access Points	VM Image Builder templates should use private link	1.1.0

Microsoft cloud security benchmark

The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Microsoft cloud security benchmark, see the Azure Security Benchmark mapping files.

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - Microsoft cloud security benchmark.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Security	NS-2	Secure cloud services with network controls	VM Image Builder templates should use private link	1.1.0

NIST SP 800-171 R2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - NIST SP 800-171 R2. For more information about this compliance standard, see NIST SP 800-171 R2.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	VM Image Builder templates should use private link	1.1.0
Access Control	3.1.12	Monitor and control remote access sessions.	VM Image Builder templates should use private link	1.1.0
Access Control	3.1.13	Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.	VM Image Builder templates should use private link	1.1.0
Access Control	3.1.14	Route remote access via managed access control points.	VM Image Builder templates should use private link	1.1.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	VM Image Builder templates should use private link	1.1.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	VM Image Builder templates should use private link	1.1.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	VM Image Builder templates should use private link	1.1.0
System and Communications	3.13.5	Implement subnetworks for publicly accessible system components that	VM Image Builder	1.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection		are physically or logically separated from internal networks.	templates should use private link	

NIST SP 800-53 Rev. 4

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 4. For more information about this compliance standard, see NIST SP 800-53 Rev. 4.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17	Remote Access	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	VM Image Builder templates should use private link	1.1.0
System And Communications Protection	SC-7	Boundary Protection	VM Image Builder templates should use private link	1.1.0
System And Communications Protection	SC-7 (3)	Access Points	VM Image Builder templates should use private link	1.1.0

NIST SP 800-53 Rev. 5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 5. For more information about this compliance standard, see NIST SP 800-53 Rev. 5.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17	Remote Access	VM Image Builder templates should use private link	1.1.0
Access Control	AC-17 (1)	Monitoring and Control	VM Image Builder templates should use private link	1.1.0
System and Communications Protection	SC-7	Boundary Protection	VM Image Builder templates should use private link	1.1.0
System and Communications Protection	SC-7 (3)	Access Points	VM Image Builder templates should use private link	1.1.0

NL BIO Cloud Theme

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for NL BIO Cloud Theme. For more information about this compliance standard, see Baseline Information Security Government Cybersecurity - Digital Government (digitaleoverheid.nl).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	VM Image Builder templates should use private link	1.1.0

Reserve Bank of India IT Framework for Banks v2016

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - RBI ITF Banks v2016. For more information about this compliance standard, see RBI ITF Banks v2016 (PDF).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Patch/Vulnerability & Change Management		Patch/Vulnerability & Change Management-7.7	VM Image Builder templates should use private link	1.1.0

SWIFT CSP-CSCF v2021

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for SWIFT CSP-CSCF v2021. For more information about this compliance standard, see SWIFT CSP CSCF v2021.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SWIFT Environment Protection	1.1	SWIFT Environment Protection	VM Image Builder templates should use private link	1.1.0
Reduce Attack Surface and Vulnerabilities	2.3	System Hardening	VM Image Builder templates should use private link	1.1.0

SWIFT CSP-CSCF v2022

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for SWIFT CSP-

CSCF v2022. For more information about this compliance standard, see SWIFT CSP CSCF v2022.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	VM Image Builder templates should use private link	1.1.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	VM Image Builder templates should use private link	1.1.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	VM Image Builder templates should use private link	1.1.0

Next steps

- Learn more about Azure Policy Regulatory Compliance.
- See the built-ins on the Azure Policy GitHub repo.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use Azure VM Image Builder for Linux VMs to access an existing Azure virtual network

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to use Azure VM Image Builder to create a basic, customized Linux image that has access to existing resources on a virtual network. The build virtual machine (VM) you create is deployed to a new or existing virtual network that you specify in your subscription. When you use an existing Azure virtual network, VM Image Builder doesn't require public network connectivity.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Set variables and permissions

For this task, you use some pieces of information repeatedly. Create some variables to store that information.

Azure CLI

```
# set your environment variables here!!!!

# destination image resource group
imageResourceGroup=aibImageRG01

# location (see possible locations in main docs)
location=WestUS2

# your subscription
# get the current subID : 'az account show | grep id'
subscriptionID=$(az account show --query id --output tsv)

# name of the image to be created
imageName=aibCustomLinuxImg01

# image distribution metadata reference name
runOutputName=aibCustLinManImg01ro


# VNET properties (update to match your existing VNET, or leave as-is for
demo)
# VNET name
vnetName=myexistingvnet01
# subnet name
subnetName=subnet01
# VNET resource group name
# NOTE! The VNET must always be in the same region as the Azure Image
Builder service region.
vnetRgName=existingVnetRG
# Existing Subnet NSG Name or the demo will create it
nsgName=aibdemoNsg
```

Create the resource group.

Azure CLI

```
az group create -n $imageResourceGroup -l $location
```

Configure networking

If you don't have an existing virtual network, subnet, or network security group (NSG), use the following script to create one.

Azure CLI

```


# Create a resource group

az group create -n $vnetRgName -l $location
```

```
# Create VNET

az network vnet create \
    --resource-group $vnetRgName \
    --name $vnetName --address-prefix 10.0.0.0/16 \
    --subnet-name $subnetName --subnet-prefix 10.0.0.0/24

# Create base NSG to simulate an existing NSG

az network nsg create -g $vnetRgName -n $nsgName

az network vnet subnet update \
    --resource-group $vnetRgName \
    --vnet-name $vnetName \
    --name $subnetName \
    --network-security-group $nsgName

#  NOTE! The virtual network must always be in the same region as the Azure
Image Builder service region.
```

Add an NSG rule

This rule allows connectivity from the VM Image Builder load balancer to the proxy VM. Port 60001 is for Linux, and port 60000 is for Windows. The proxy VM connects to the build VM by using port 22 for Linux, or port 5986 for Windows.

Azure CLI

```
az network nsg rule create \
    --resource-group $vnetRgName \
    --nsg-name $nsgName \
    -n AzureImageBuilderNsgRule \
    --priority 400 \
    --source-address-prefixes AzureLoadBalancer \
    --destination-address-prefixes VirtualNetwork \
    --destination-port-ranges 60000-60001 --direction inbound \
    --access Allow --protocol Tcp \
    --description "Allow Image Builder Private Link Access to Proxy VM"
```

Disable private service policy on the subnet

Here's how:

Azure CLI

```
az network vnet subnet update \
  --name $subnetName \
```

```
  --resource-group $vnetRgName \
  --vnet-name $vnetName \
  --disable-private-link-service-network-policies true
```

For more information, see Azure VM Image Builder networking options.

Modify the example template and create role

After you configure networking, you can modify the example template and create a role. Here's how:

Bash

```
# download the example and configure it with your vars

curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickquickstarts/1a_Creating_a_Custom_Linux_Image_on_Existing_VNET/existingVNETLinux.json -o existingVNETLinux.json
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12_Creating_AIB_Security_Roles/aibRoleNetworking.json -o
aibRoleNetworking.json
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12_Creating_AIB_Security_Roles/aibRoleImageCreation.json -o
aibRoleImageCreation.json

sed -i -e "s/<subscriptionID>/$subscriptionID/g" existingVNETLinux.json
sed -i -e "s/<rgName>/$imageResourceGroup/g" existingVNETLinux.json
sed -i -e "s/<region>/$location/g" existingVNETLinux.json
sed -i -e "s/<imageName>/$imageName/g" existingVNETLinux.json
sed -i -e "s/<runOutputName>/$runOutputName/g" existingVNETLinux.json

sed -i -e "s/<vnetName>/$vnetName/g" existingVNETLinux.json
sed -i -e "s/<subnetName>/$subnetName/g" existingVNETLinux.json
sed -i -e "s/<vnetRgName>/$vnetRgName/g" existingVNETLinux.json

sed -i -e "s/<subscriptionID>/$subscriptionID/g" aibRoleImageCreation.json
sed -i -e "s/<rgName>/$imageResourceGroup/g" aibRoleImageCreation.json

sed -i -e "s/<subscriptionID>/$subscriptionID/g" aibRoleNetworking.json
sed -i -e "s/<vnetRgName>/$vnetRgName/g" aibRoleNetworking.json
```

Set permissions on the resource group

VM Image Builder uses the user identity provided to inject the image into Azure Compute Gallery. In this example, you create an Azure role definition that can distribute the image to the gallery. The role definition is then assigned to the user identity.

Azure CLI

```
# create user assigned identity for image builder
identityName=aibBuiUserId$(date +'%s')
az identity create -g $imageResourceGroup -n $identityName

# get identity id
imgBuilderCliId=$(az identity show -g $imageResourceGroup -n $identityName -
-query clientId -o tsv)

# get the user identity URI, needed for the template
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$imageResourceGro
up/providers/Microsoft.ManagedIdentity/userAssignedIdentities/$identityName

# update the template
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g" existingVNETLinux.json

# make role name unique, to avoid clashes in the same Azure Active Directory
domain
imageRoleDefName="Azure Image Builder Image Def"$(date +'%s')
netRoleDefName="Azure Image Builder Network Def"$(date +'%s')

# update the definitions
sed -i -e "s/Azure Image Builder Service Image Creation
Role/$imageRoleDefName/g" aibRoleImageCreation.json
sed -i -e "s/Azure Image Builder Service Networking Role/$netRoleDefName/g"
aibRoleNetworking.json
```

Instead of granting VM Image Builder lower granularity and increased privilege, you can create two roles. One role gives the builder permissions to create an image, and the other allows it to connect the build VM and load balancer to your virtual network.

Azure CLI

```
# create role definitions
az role definition create --role-definition ./aibRoleImageCreation.json
az role definition create --role-definition ./aibRoleNetworking.json

# grant role definition to the user assigned identity
az role assignment create \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup

az role assignment create \
    --assignee $imgBuilderCliId \
```

```
    --role "$netRoleDefName" \
    --scope /subscriptions/$subscriptionID/resourceGroups/$vnetRgName
```

For more information, see Configure Azure VM Image Builder permissions by using the Azure CLI or Configure Azure VM Image Builder permissions by using PowerShell.

Create the image

Submit the image configuration to VM Image Builder.

Azure CLI

```
az resource create \
    --resource-group $imageResourceGroup \
    --properties @existingVNETLinux.json \
    --is-full-object \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n existingVNETLinuxTemplate01

# Wait approximately 1-3 mins (validation, permissions etc.)
```

Start the image build.

Azure CLI

```
az resource invoke-action \
     --resource-group $imageResourceGroup \
     --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
     -n existingVNETLinuxTemplate01 \
     --action Run

# Wait approximately 15 mins
```

It can take a while to create the image and replicate it to both regions. Wait until this part is finished before moving on to creating a VM.

Create a VM

Create a VM from the image version that was created by VM Image Builder.

Azure CLI

```
az vm create \
  --resource-group $imageResourceGroup \
  --name aibImgVm0001 \
  --admin-username aibuser \
```

```
  --image $imageName \
  --location $location \
  --generate-ssh-keys
```

Use Secure Shell (SSH) to get into the VM.

Bash

```
ssh aibuser@<publicIpAddress>
```

You should see the image was customized with a *Message of the Day* as soon as your SSH connection is established!

Console

```
*******************************************************
**            This VM was built from the:            **
**      !! AZURE VM IMAGE BUILDER Custom Image !!     **
**         You have just been Customized :-)         **
*******************************************************
```

Clean up resources

If you want to recustomize the image version to create a new version of the same image, skip the next steps and go on to Use Azure VM Image Builder to create another image version.

The following deletes the image that was created, along with all of the other resource files. Make sure you are finished with this deployment before deleting the resources.

When you delete gallery resources, you need to delete all of the image versions before you can delete the image definition used to create them. To delete a gallery, you first need to have deleted all of the image definitions in the gallery.

Delete the VM Image Builder template:

Azure CLI

```
az resource delete \
    --resource-group $imageResourceGroup \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n existingVNETLinuxTemplate01
```

Delete permissions assignments, roles, and identity:

Azure CLI

```
az role assignment delete \
    --assignee $imgBuilderCliId \
    --role $imageRoleDefName \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup

az role assignment delete \
    --assignee $imgBuilderCliId \
    --role $netRoleDefName \
    --scope /subscriptions/$subscriptionID/resourceGroups/$vnetRgName


az role definition delete --name "$imageRoleDefName"
az role definition delete --name "$netRoleDefName"

az identity delete --ids $imgBuilderId
```

Delete the resource group:

Azure CLI

```
az group delete -n $imageResourceGroup
```

If you created a virtual network for this quickstart, you can delete the virtual network if it's no longer being used.

Next steps

Azure Compute Galleries

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use Azure VM Image Builder to access an existing Azure virtual network

Article • 08/22/2024

Applies to: ✔ Windows VMs

This article shows you how to use Azure VM Image Builder to create a basic, customized Windows image that has access to existing resources on a virtual network. The build virtual machine (VM) you create is deployed to a new or existing virtual network that you specify in your subscription. When you use an existing Azure virtual network, VM Image Builder doesn't require public network connectivity.

Set variables and permissions

For this task, you use some pieces of information repeatedly. Create some variables to store that information.

PowerShell

```
# Step 1: Import module
Import-Module Az.Accounts

# Step 2: get existing context
$currentAzContext = Get-AzContext

# destination image resource group
$imageResourceGroup="aibImageRG"

# location (see possible locations in main docs)
$location="westus2"

## if you need to change your subscription: Get-AzSubscription / Select-
AzSubscription -SubscriptionName

# get subscription, this will get your current subscription
$subscriptionID=$currentAzContext.Subscription.Id

# name of the image to be created
$imageName="win2019image01"

# image distribution metadata reference name
$runOutputName="win2019ManImg02ro"

# image template name
$imageTemplateName="window2019VnetTemplate03"

# distribution properties object name (runOutput), i.e. this gives you the
```

```
properties of the managed image on completion
$runOutputName="winSvrSigR01"

# VNET properties (update to match your existing virtual network, or leave
as-is for demo)
# VNET name
$vnetName="myexistingvnet01"
# subnet name
$subnetName="subnet01"
# VNET resource group name
$vnetRgName="existingVnetRG"
# Existing Subnet NSG Name or the demo will create it
$nsgName="aibdemoNsg"
# NOTE! The virtual network must always be in the same region as the VM
Image Builder service region.
```

Create the resource group.

PowerShell

```
New-AzResourceGroup -Name $imageResourceGroup -Location $location
```

Configure networking

If you don't have an existing virtual network, subnet, or network security group (NSG), use the following script to create one.

PowerShell

```
New-AzResourceGroup -Name $vnetRgName -Location $location

## Create base NSG to simulate an existing NSG
New-AzNetworkSecurityGroup -Name $nsgName -ResourceGroupName $vnetRgName -
location $location

$nsg = Get-AzNetworkSecurityGroup -Name $nsgName -ResourceGroupName
$vnetRgName

$subnet = New-AzVirtualNetworkSubnetConfig -Name $subnetName -AddressPrefix
"10.0.1.0/24" -PrivateLinkServiceNetworkPoliciesFlag "Disabled" -
NetworkSecurityGroup $nsg

New-AzVirtualNetwork -Name $vnetName -ResourceGroupName $vnetRgName -
Location $location -AddressPrefix "10.0.0.0/16" -Subnet $subnet

## NOTE! The virtual network must always be in the same region as the VM
Image Builder service region.
```

Add an NSG rule

This rule allows connectivity from the VM Image Builder load balancer to the proxy VM. Port 60001 is for Linux, and port 60000 is for Windows. The proxy VM connects to the build VM by using port 22 for Linux, or port 5986 for Windows.

PowerShell

```
Get-AzNetworkSecurityGroup -Name $nsgName -ResourceGroupName $vnetRgName  |
Add-AzNetworkSecurityRuleConfig -Name AzureImageBuilderAccess -Description
"Allow Image Builder Private Link Access to Proxy VM" -Access Allow -
Protocol Tcp -Direction Inbound -Priority 400 -SourceAddressPrefix
AzureLoadBalancer -SourcePortRange * -DestinationAddressPrefix
VirtualNetwork -DestinationPortRange 60000-60001 | Set-
AzNetworkSecurityGroup
```

Disable private service policy on the subnet

Here's how:

PowerShell

```
$virtualNetwork= Get-AzVirtualNetwork -Name $vnetName -ResourceGroupName
$vnetRgName

($virtualNetwork | Select -ExpandProperty subnets | Where-Object  {$_.Name -
eq $subnetName} ).privateLinkServiceNetworkPolicies = "Disabled"

$virtualNetwork | Set-AzVirtualNetwork
```

For more information, see Azure VM Image Builder networking options.

Modify the example template and create role

After you configure networking, you can modify the example template and create a role. Here's how:

PowerShell

```
$templateUrl="https://raw.githubusercontent.com/azure/azvmimagebuilder/maste
r/quickquickstarts/1a_Creating_a_Custom_Win_Image_on_Existing_VNET/existingV
NETWindows.json"
$templateFilePath = "existingVNETWindows.json"

$aibRoleNetworkingUrl="https://raw.githubusercontent.com/azure/azvmimagebuil
der/master/solutions/12_Creating_AIB_Security_Roles/aibRoleNetworking.json"
```

```
$aibRoleNetworkingPath = "aibRoleNetworking.json"

$aibRoleImageCreationUrl="https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12_Creating_AIB_Security_Roles/aibRoleImageCreation.json"
$aibRoleImageCreationPath = "aibRoleImageCreation.json"

# download configs
Invoke-WebRequest -Uri $templateUrl -OutFile $templateFilePath -UseBasicParsing

Invoke-WebRequest -Uri $aibRoleNetworkingUrl -OutFile $aibRoleNetworkingPath -UseBasicParsing

Invoke-WebRequest -Uri $aibRoleImageCreationUrl -OutFile $aibRoleImageCreationPath -UseBasicParsing

# update AIB image config template
((Get-Content -path $templateFilePath -Raw) -replace '<subscriptionID>',$subscriptionID) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<rgName>',$imageResourceGroup) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<region>',$location) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<runOutputName>',$runOutputName) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<imageName>',$imageName) | Set-Content -Path $templateFilePath

((Get-Content -path $templateFilePath -Raw) -replace '<vnetName>',$vnetName) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<subnetName>',$subnetName) | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace '<vnetRgName>',$vnetRgName) | Set-Content -Path $templateFilePath
```

Create a user-assigned identity and set permissions

Next, you create a user-assigned identity and set permissions. Here's how:

PowerShell

```
# setup role def names, these need to be unique
$timeInt=$(get-date -UFormat "%s")
$imageRoleDefName="Azure Image Builder Image Def"+$timeInt
$networkRoleDefName="Azure Image Builder Network Def"+$timeInt
$idenityName="aibIdentity"+$timeInt

# create user identity
## Add AZ PS module to support AzUserAssignedIdentity
```

```powershell
Install-Module -Name Az.ManagedServiceIdentity

# create identity
New-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -Name
$idenityName

$idenityNameResourceId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $idenityName).Id
$idenityNamePrincipalId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $idenityName).PrincipalId

# update template with identity
((Get-Content -path $templateFilePath -Raw) -replace
'<imgBuilderId>',$idenityNameResourceId) | Set-Content -Path
$templateFilePath

# update the role defintion names
((Get-Content -path $aibRoleImageCreationPath -Raw) -replace 'Azure Image
Builder Service Image Creation Role',$imageRoleDefName) | Set-Content -Path
$aibRoleImageCreationPath
((Get-Content -path $aibRoleNetworkingPath -Raw) -replace 'Azure Image
Builder Service Networking Role',$networkRoleDefName) | Set-Content -Path
$aibRoleNetworkingPath

# update role definitions
((Get-Content -path $aibRoleNetworkingPath -Raw) -replace
'<subscriptionID>',$subscriptionID) | Set-Content -Path
$aibRoleNetworkingPath
((Get-Content -path $aibRoleNetworkingPath -Raw) -replace
'<vnetRgName>',$vnetRgName) | Set-Content -Path $aibRoleNetworkingPath

((Get-Content -path $aibRoleImageCreationPath -Raw) -replace
'<subscriptionID>',$subscriptionID) | Set-Content -Path
$aibRoleImageCreationPath
((Get-Content -path $aibRoleImageCreationPath -Raw) -replace '<rgName>',
$imageResourceGroup) | Set-Content -Path $aibRoleImageCreationPath

# create role definitions from role configurations examples, this avoids
granting contributor to the SPN
New-AzRoleDefinition -InputFile  ./aibRoleImageCreation.json
New-AzRoleDefinition -InputFile  ./aibRoleNetworking.json

# grant role definition to image builder user identity
New-AzRoleAssignment -ObjectId $idenityNamePrincipalId -RoleDefinitionName
$imageRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
New-AzRoleAssignment -ObjectId $idenityNamePrincipalId -RoleDefinitionName
$networkRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$vnetRgName"
```

For more information, see Configure Azure VM Image Builder permissions by using the Azure CLI or Configure Azure VM Image Builder permissions by using PowerShell.

Create the image

Submit the image configuration to Azure VM Image Builder.

PowerShell

```
New-AzResourceGroupDeployment -ResourceGroupName $imageResourceGroup -
TemplateFile $templateFilePath -api-version "2020-02-14" -imageTemplateName
$imageTemplateName -svclocation $location
```

> **Note**
>
> This will take a minute, as validation is run in regards to security, dependenciec, etc.

Start the image build.

PowerShell

```
Invoke-AzResourceAction -ResourceName $imageTemplateName -ResourceGroupName
$imageResourceGroup -ResourceType
Microsoft.VirtualMachineImages/imageTemplates -ApiVersion "2020-02-14" -
Action Run -Force
```

Get build status and properties

First, you query the image template for current or last run status, and for image template settings.

powerShell

```
$managementEp = $currentAzureContext.Environment.ResourceManagerUrl

$urlBuildStatus = [System.String]::Format("
{0}subscriptions/{1}/resourceGroups/$imageResourceGroup/providers/Microsoft.
VirtualMachineImages/imageTemplates/{2}?api-version=2020-02-14",
$managementEp, $currentAzureContext.Subscription.Id,$imageTemplateName)

$buildStatusResult = Invoke-WebRequest -Method GET  -Uri $urlBuildStatus -
UseBasicParsing -Headers  @{"Authorization"= ("Bearer " + $accessToken)} -
ContentType application/json
$buildJsonStatus =$buildStatusResult.Content
$buildJsonStatus
```

The image build for this example takes approximately 50 minutes (including multiple reboots and Windows updates). When you query the status, look for `lastRunStatus`. The following code shows that the build is still running. If it had completed successfully, it would show `succeeded`.

text

```
  "lastRunStatus": {
   "startTime": "2019-08-21T00:39:40.61322415Z",
   "endTime": "0001-01-01T00:00:00Z",
   "runState": "Running",
   "runSubState": "Building",
   "message": ""
  },
```

Query the distribution properties

If you're distributing to a VHD location, need managed image location properties, or Azure Compute Gallery replications status, you need to query `runOutput`. Every time you have a distribution target, you will have a unique `runOutput`, to describe properties of the distribution type.

powerShell

```
$managementEp = $currentAzureContext.Environment.ResourceManagerUrl
$urlRunOutputStatus = [System.String]::Format("
{0}subscriptions/{1}/resourceGroups/$imageResourceGroup/providers/Microsoft.
VirtualMachineImages/imageTemplates/$imageTemplateName/runOutputs/{2}?api-
version=2020-02-14", $managementEp, $currentAzureContext.Subscription.Id,
$runOutputName)

$runOutStatusResult = Invoke-WebRequest -Method GET  -Uri
$urlRunOutputStatus -UseBasicParsing -Headers  @{"Authorization"= ("Bearer "
+ $accessToken)} -ContentType application/json
$runOutJsonStatus =$runOutStatusResult.Content
$runOutJsonStatus
```

Create a VM

Now that the build is finished, you can build a VM from the image. Use the examples from the PowerShell New-AzVM documentation.

Clean up tasks

You can now delete the image template artifact, the role assignment, and the resource groups if you want to.

Here's how to delete the image template artifact:

powerShell

```
# Get ResourceID of the Image Template
$resTemplateId = Get-AzResource -ResourceName $imageTemplateName -
ResourceGroupName $imageResourceGroup -ResourceType
Microsoft.VirtualMachineImages/imageTemplates -ApiVersion "2020-02-14"

### Delete Image Template Artifact
Remove-AzResource -ResourceId $resTemplateId.ResourceId -Force
```

Here's how to delete the role assignment:

powerShell

```
## remove role assignments
Remove-AzRoleAssignment -ObjectId $idenityNamePrincipalId  -
RoleDefinitionName $imageRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
Remove-AzRoleAssignment -ObjectId $idenityNamePrincipalId -
RoleDefinitionName $networkRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$vnetRgName"

## remove definitions
Remove-AzRoleDefinition -Id $imageRoleDefObjId -Force
Remove-AzRoleDefinition -Id $networkRoleObjId -Force

## delete identity
Remove-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -Name
$idenityName -Force
```

Here's how to delete resource groups:

powerShell

```
Remove-AzResourceGroup $imageResourceGroup -Force


# delete VNET created
# BEWARE! In this example, you have either used an existing virtual network
or created one for this example. Do not delete your existing virtual
network. If you want to delete the virtual network resource group used in
this example '$vnetRgName', modify the preceding code.
```

Next steps

Azure Compute Galleries

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure VM Image Builder networking options

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure VM Image Builder (AIB) deploys an Azure Container Instance (ACI) in the staging resource group in your subscription. The ACI must be associated with a subnet in an Azure virtual network (VNet). The AIB service code running in the ACI deploys a build virtual machine (build VM) in the staging resource group to customize your image, and the build VM must be placed on a separate subnet. To customize and validate your image, the ACI must have network connectivity to the build VM. Based on your networking requirements and organizational policies, you can configure AIB to use different network topologies. You can choose to bring both subnets, one subnet, or none. For recommendations, see Best practices for Azure VM Image Builder.

Network topologies

Don't bring your own Build VM subnet or the ACI subnet

- You can select this topology by not specifying the `vnetConfig` field in the Image Template or by specifying the field but without `subnetId` and `containerInstanceSubnetId` subfields.
- If you don't specify any subnets, AIB deploys an Azure virtual network (VNet) in the staging resource group with two subnets: one for the Azure Container Instance and one for the build VM. Both subnets are associated with a network security group (NSG) that includes default rules, while still allowing direct line-of-sight connectivity required for customization. The build VM is also deployed with a network interface resource (and other resources not directly related to networking, like Managed Disk). After the build completes, the build VM and networking resources are deleted.

Bring your own Build VM subnet and bring your own ACI subnet

- You can select this topology by specifying the `vnetConfig` field with the `subnetId` and `containerInstanceSubnetId` subfields in the image template. This option, including the `containerInstanceSubnetId` subfield, is available starting with API version 2024-02-01. You can also update existing templates to use this topology.
- In this topology, AIB deploys the build VM to the specified build VM subnet and the ACI to the specified ACI subnet. Because the subnets are already provided, AIB does not deploy a virtual network, subnets, or a network security group. This topology is useful

when quota restrictions or policies prevent deployment of these resources. The build VM can access resources that are reachable from your virtual network, and you can also create a siloed virtual network that is not connected to any other virtual network. The ACI subnet must satisfy the prerequisites for isolated image builds. For details about these fields, see the template reference.

- This topology is the recommended option for most scenarios because it gives you full control over both subnets, simplifies setup and network configuration, can reduce overall deployment costs, and helps align networking with your organization's security and governance requirements.

Bring your own build VM subnet but not your own ACI subnet

- You can select this topology by specifying the `vnetConfig` field with the `subnetId` subfield while omitting the `containerInstanceSubnetId` subfield in the image template.
- In this topology, AIB deploys a temporary virtual network in the staging resource group with two subnets, each associated with a network security group (NSG). One subnet hosts the ACI, and the other hosts the private endpoint resource. The build VM is deployed in your specified subnet. To enable communication between the ACI subnet and your build VM subnet, AIB also deploys a Private Link-based communication path in the staging resource group that includes a private endpoint, Private Link Service, Azure Load Balancer, network interfaces, and a proxy virtual machine. The exact resources and configurations vary slightly depending on whether you are customizing a Windows image or a Linux image.
- This topology is generally not recommended for most scenarios because it can increase deployment costs, require more setup and operational configuration, and introduce additional components that can make the end-to-end pipeline more sensitive to failures.

For examples of this topology, see the following articles:

- Use Azure VM Image Builder for Windows VMs allowing access to an existing Azure virtual network
- Use Azure VM Image Builder for Linux VMs allowing access to an existing Azure virtual network

What is Azure Private Link?

Azure Private Link provides private connectivity from a virtual network to Azure platform as a service (PaaS), or to customer-owned or Microsoft partner services. It simplifies network architecture and secures connectivity between Azure endpoints by eliminating data exposure to the public internet. Private Link requires an IP address from the specified virtual network and subnet. Azure currently does not support network policies on these IP addresses, so you must

disable network policies on the subnet. For more information, see the Private Link documentation.

Why deploy a proxy VM?

When a VM without a public IP is behind an internal load balancer, it does not have internet access. The load balancer used for the virtual network is internal. The proxy VM allows internet access for the build VM during builds, and AIB uses the proxy VM to send commands between the service and the build VM. You can use the associated network security groups to restrict build VM access. Traffic from the ACI goes across the private link to the load balancer. The load balancer communicates with the proxy VM on port 60001 for Linux, or port 60000 for Windows. The proxy forwards commands to the build VM on port 22 for Linux, or port 5986 for Windows. By default, the deployed proxy VM size is *Standard A1_v2*, but you can change the size. For details, see the template reference.

Checklist for using your virtual network

1. Allow Azure Load Balancer to communicate with the proxy VM in a network security group.
    - Azure CLI example
    - PowerShell example
2. Disable the private service policy on the subnet.
    - Azure CLI example
    - PowerShell example
3. Allow VM Image Builder to create a load balancer, and add VMs to the virtual network.
    - Azure CLI example
    - PowerShell example
4. Allow VM Image Builder to read and write source images, and create images.
    - Azure CLI example
    - PowerShell example
5. Ensure that you're using a virtual network in the same region as the VM Image Builder service region.

Additional considerations

Required permissions for an existing virtual network

VM Image Builder requires specific permissions to use an existing virtual network. For more information, see Configure Azure VM Image Builder permissions by using the Azure CLI or Configure Azure VM Image Builder permissions by using PowerShell.

> **Note**
>
> The virtual network must be in the same region as the VM Image Builder service region.

> **Important**
>
> The Azure VM Image Builder service modifies the WinRM connection configuration on all Windows builds to use HTTPS on port 5986 instead of the default HTTP port on 5985. This configuration change can impact workflows that rely on WinRM communication.

Connectivity model

AIB does not deploy a public IP for direct connectivity, and the AIB service component that runs in the platform subscription does not have network connectivity to the ACI or the build VM. Only the AIB component that runs in the ACI has connectivity to the build VM to perform customization.

Supported subnet combinations

You can configure both subnets (`subnetId` and `containerInstanceSubnetId`) or only the build VM subnet (`subnetId`). A configuration that specifies only the ACI subnet (`containerInstanceSubnetId`) is not supported.

Next steps

Azure VM Image Builder overview

Last updated on 03/28/2026

Configure Azure VM Image Builder permissions by using the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When you register for Azure VM Image Builder, your subscription gets access to a VM Image Builder service principal name (SPN). This registration also authorizes the service permission to create, manage, and delete a staging resource group. For the image building process, the Contributor role assignment is also required at the staging resource group.

If you want VM Image Builder to distribute images, you need to create a user-assigned identity in Azure, with permissions to read and write images. For example, you might want to distribute images to managed images or to Azure Compute Gallery. If you're accessing Azure storage, then the user-assigned identity you create needs permissions to read private or public containers.

You must set up permissions and privileges prior to building an image. The following sections detail how to configure possible scenarios by using the Azure CLI.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Create a user-assigned managed identity

VM Image Builder requires you to create an Azure user-assigned managed identity. VM Image Builder uses this identity to read images, write images, and access Azure storage accounts. You grant the identity permission to do specific actions in your subscription.

> **Note**
>
> User-assigned managed identity is the correct way to grant permissions to the image resource groups. The SPN is deprecated for this purpose.

The following example shows you how to create an Azure user-assigned managed identity. Replace the placeholder settings to set your variables.

Expand table

Setting	Description
<Resource group>	The resource group where you want to create the user-assigned managed identity.

Azure CLI

```
identityName="aibIdentity"
imageResourceGroup=<Resource group>

az identity create \
    --resource-group $imageResourceGroup \
    --name $identityName
```

For more information, see Azure user-assigned managed identity.

Allow VM Image Builder to distribute images

For VM Image Builder to distribute images, the service must be allowed to inject the images into resource groups. To grant the required permissions, create a user-assigned managed identity, and grant it rights on the resource group where the image is built. VM Image Builder doesn't have permission to access resources in other resource groups in the subscription. You need to take explicit actions to allow access, to prevent your builds from failing.

You don't need to grant the user-assigned managed identity contributor rights on the resource group to distribute images. However, the user-assigned managed identity

needs the following Azure `Actions` permissions in the distribution resource group:

```
Actions

Microsoft.Compute/images/write
Microsoft.Compute/images/read
Microsoft.Compute/images/delete
```

If you want to distribute to Azure Compute Gallery, you also need:

```
Actions

Microsoft.Compute/galleries/read
Microsoft.Compute/galleries/images/read
Microsoft.Compute/galleries/images/versions/read
Microsoft.Compute/galleries/images/versions/write
```

Permission to customize existing images

For VM Image Builder to build images from source custom images, the service must be allowed to read the images into these resource groups. To grant the required permissions, create a user-assigned managed identity, and grant it rights on the resource group where the image is located.

Here's how you build from an existing custom image:

```
Actions

Microsoft.Compute/images/read
```

Here's how you build from an existing Azure Compute Gallery version:

```
Actions

Microsoft.Compute/galleries/read
Microsoft.Compute/galleries/images/read
Microsoft.Compute/galleries/images/versions/read
```

Permission to customize images on your virtual networks

VM Image Builder has the capability to deploy and use an existing virtual network in your subscription, thus allowing customizations access to connected resources.

You don't need to grant the user-assigned managed identity contributor rights on the resource group to deploy a VM to an existing virtual network. However, the user-assigned managed identity needs the following Azure `Actions` permissions on the virtual network resource group:

Actions

```
Microsoft.Network/virtualNetworks/read
Microsoft.Network/virtualNetworks/subnets/join/action
```

Create an Azure role definition

The following examples create an Azure role definition from the actions described in the previous sections. The examples are applied at the resource group level. Evaluate and test if the examples are granular enough for your requirements.

The image actions allow read and write. Decide what is appropriate for your environment. For example, create a role to allow VM Image Builder to read images from resource group *example-rg-1*, and write images to resource group *example-rg-2*.

Custom image Azure role example

The following example creates an Azure role to use and distribute a source custom image. You then grant the custom role to the user-assigned managed identity for VM Image Builder.

To simplify the replacement of values in the example, set the following variables first. Replace the placeholder settings to set your variables.

Expand table

Setting	Description
<Subscription ID>	Your Azure subscription ID.
<Resource group>	Resource group for the custom image.

Azure CLI

```
# Subscription ID - You can get this using `az account show | grep id` or
from the Azure portal.
subscriptionID=$(az account show --query id --output tsv)
# Resource group - image builder will only support creating custom images in
the same Resource Group as the source managed image.
imageResourceGroup=<Resource group>
identityName="aibIdentity"

# Use *cURL* to download the a sample JSON description
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12
_Creating_AIB_Security_Roles/aibRoleImageCreation.json -o
aibRoleImageCreation.json

# Create a unique role name to avoid clashes in the same Azure Active
Directory domain
imageRoleDefName="Azure Image Builder Image Def"$(date +'%s')

# Update the JSON definition using stream editor
sed -i -e "s/<subscriptionID>/$subscriptionID/g" aibRoleImageCreation.json
sed -i -e "s/<rgName>/$imageResourceGroup/g" aibRoleImageCreation.json
sed -i -e "s/Azure Image Builder Service Image Creation
Role/$imageRoleDefName/g" aibRoleImageCreation.json

# Create a custom role from the sample aibRoleImageCreation.json description
file.
az role definition create --role-definition ./aibRoleImageCreation.json

# Get the user-assigned managed identity id
imgBuilderCliId=$(az identity show -g $imageResourceGroup -n $identityName -
-query clientId -o tsv)

# Grant the custom role to the user-assigned managed identity for Azure
Image Builder.
az role assignment create \
    --assignee $imgBuilderCliId \
    --role $imageRoleDefName \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup
```

Existing virtual network Azure role example

The following example creates an Azure role to use and distribute an existing virtual network image. You then grant the custom role to the user-assigned managed identity for VM Image Builder.

To simplify the replacement of values in the example, set the following variables first. Replace the placeholder settings to set your variables.

Expand table

Setting	Description
<Subscription ID>	Your Azure subscription ID.
<Resource group>	The virtual network resource group

Azure CLI

```
# Subscription ID - You can get this using `az account show | grep id` or
from the Azure portal.
subscriptionID=$(az account show --query id --output tsv)
VnetResourceGroup=<Resource group>
identityName="aibIdentity"

# Use *cURL* to download the a sample JSON description
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12
_Creating_AIB_Security_Roles/aibRoleNetworking.json -o
aibRoleNetworking.json

# Create a unique role name to avoid clashes in the same domain
netRoleDefName="Azure Image Builder Network Def"$(date +'%s')

# Update the JSON definition using stream editor
sed -i -e "s/<subscriptionID>/$subscriptionID/g" aibRoleNetworking.json
sed -i -e "s/<vnetRgName>/$VnetResourceGroup/g" aibRoleNetworking.json
sed -i -e "s/Azure Image Builder Service Networking Role/$netRoleDefName/g"
aibRoleNetworking.json

# Create a custom role from the aibRoleNetworking.json description file.
az role definition create --role-definition ./aibRoleNetworking.json

# Get the user-assigned managed identity id
imgBuilderCliId=$(az identity show -g $imageResourceGroup -n $identityName -
-query clientId -o tsv)

# Grant the custom role to the user-assigned managed identity for Azure
Image Builder.
az role assignment create \
    --assignee $imgBuilderCliId \
    --role $netRoleDefName \
    --scope /subscriptions/$subscriptionID/resourceGroups/$VnetResourceGroup
```

Using managed identity for Azure Storage access

If you want to authenticate with Azure Storage and use private containers, VM Image Builder needs a user-assigned managed identity. VM Image Builder uses the identity to authenticate with Azure Storage.

> **Note**
>
> VM Image Builder only uses the identity at the time that you submit the image template. The build VM doesn't have access to the identity during image build.

Use the Azure CLI to create the user-assigned managed identity:

Azure CLI

```
az role assignment create \
    --assignee <Image Builder client ID> \
    --role "Storage Blob Data Reader" \
    --scope /subscriptions/<Subscription ID>/resourceGroups/<Resource
group>/providers/Microsoft.Storage/storageAccounts/$scriptStorageAcc/blobSer
vices/default/containers/<Storage account container>
```

In the VM Image Builder template, provide the user-assigned managed identity:

JSON

```
    "type": "Microsoft.VirtualMachineImages/imageTemplates",
    "apiVersion": "2020-02-14",
    "location": "<Region>",
    ..
    "identity": {
    "type": "UserAssigned",
          "userAssignedIdentities": {
            "<Image Builder ID>": {}
        }
```

Replace the following placeholder settings:

Expand table

Setting	Description
<Region>	Template region
<Resource group>	Resource group
<Storage account container>	Storage account container name
<Subscription ID>	Azure subscription

For more information, see Create an image and use a user-assigned managed identity to access files in Azure Storage. You learn how to create and configure the user-assigned managed identity to access a storage account.

Next steps

Azure VM Image Builder overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Configure Azure VM Image Builder permissions by using PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When you register for Azure VM Image Builder, this grants the service permission to create, manage, and delete a staging resource group. The service also has rights to add resources to a resource group, required for the image build. During a successful registration, your subscription gets access to a VM Image Builder service principal name (SPN).

If you want VM Image Builder to distribute images, you need to create a user-assigned identity in Azure, with permissions to read and write images. For example, you might want to distribute images to managed images or to Azure Compute Gallery. If you're accessing Azure Storage, then the user-assigned identity you create needs permissions to read private or public containers.

You must set up permissions and privileges prior to building an image. The following sections detail how to configure possible scenarios by using PowerShell.

Create a user-assigned managed identity

VM Image Builder requires you to create an Azure user-assigned managed identity. VM Image Builder uses this identity to read images, write images, and access Azure Storage accounts. You grant the identity permission to do specific actions in your subscription.

> **Note**
>
> User-assigned managed identity is the correct way to grant permissions to the image resource groups. The SPN is deprecated for this purpose.

The following example shows you how to create an Azure user-assigned managed identity. Replace the placeholder settings to set your variables.

Expand table

Setting	Description
<Resource group>	The resource group where you want to create the user-assigned managed identity.

PowerShell

```
## Add AZ PS module to support AzUserAssignedIdentity
Install-Module -Name Az.ManagedServiceIdentity

$parameters = @{
    Name = 'aibIdentity'
    ResourceGroupName = '<Resource group>'
}
# create identity
New-AzUserAssignedIdentity @parameters
```

For more information, see Azure user-assigned managed identity.

Allow VM Image Builder to distribute images

For VM Image Builder to distribute images, the service must be allowed to inject the images into resource groups. To grant the required permissions, create a user-assigned managed identity, and grant it rights on the resource group where the image is built. VM Image Builder doesn't have permission to access resources in other resource groups in the subscription. You need to take explicit actions to allow access, to prevent your builds from failing.

You don't need to grant the user-assigned managed identity contributor rights on the resource group to distribute images. However, the user-assigned managed identity needs the following Azure `Actions` permissions in the distribution resource group:

Actions

```
Microsoft.Compute/images/write
Microsoft.Compute/images/read
Microsoft.Compute/images/delete
```

If you want to distribute to Azure Compute Gallery, you also need:

Actions

```
Microsoft.Compute/galleries/read
Microsoft.Compute/galleries/images/read
```

```
Microsoft.Compute/galleries/images/versions/read
Microsoft.Compute/galleries/images/versions/write
```

Permission to customize existing images

For VM Image Builder to build images from source custom images, the service must be allowed to read the images into these resource groups. To grant the required permissions, create a user-assigned managed identity, and grant it rights on the resource group where the image is located.

Here's how you build from an existing custom image:

Actions

```
Microsoft.Compute/images/read
```

Here's how you build from an existing Azure Compute Gallery version:

Actions

```
Microsoft.Compute/galleries/read
Microsoft.Compute/galleries/images/read
Microsoft.Compute/galleries/images/versions/read
```

Permission to customize images on your virtual networks

VM Image Builder has the capability to deploy and use an existing virtual network in your subscription, thus allowing customizations access to connected resources.

You don't need to grant the user-assigned managed identity contributor rights on the resource group to deploy a VM to an existing virtual network. However, the user-assigned managed identity needs the following Azure `Actions` permissions on the virtual network resource group:

Actions

```
Microsoft.Network/virtualNetworks/read
Microsoft.Network/virtualNetworks/subnets/join/action
```

Create an Azure role definition

The following examples create an Azure role definition from the actions described in the previous sections. The examples are applied at the resource group level. Evaluate and test if the examples are granular enough for your requirements.

The image actions allow read and write. Decide what is appropriate for your environment. For example, create a role to allow VM Image Builder to read images from resource group *example-rg-1*, and write images to resource group *example-rg-2*.

Custom image Azure role example

The following example creates an Azure role to use and distribute a source custom image. You then grant the custom role to the user-assigned managed identity for VM Image Builder.

To simplify the replacement of values in the example, set the following variables first. Replace the placeholder settings to set your variables.

Expand table

Setting	Description
<Subscription ID>	Your Azure subscription ID.
<Resource group>	Resource group for the custom image.

PowerShell

```
$sub_id = "<Subscription ID>"
# Resource group - image builder will only support creating custom images in
the same Resource Group as the source managed image.
$imageResourceGroup = "<Resource group>"
$identityName = "aibIdentity"

# Use a web request to download the sample JSON description
$sample_uri="https://raw.githubusercontent.com/azure/azvmimagebuilder/master
/solutions/12_Creating_AIB_Security_Roles/aibRoleImageCreation.json"
$role_definition="aibRoleImageCreation.json"

Invoke-WebRequest -Uri $sample_uri -Outfile $role_definition -
UseBasicParsing

# Create a unique role name to avoid clashes in the same Azure Active
Directory domain
$timeInt=$(get-date -UFormat "%s")
$imageRoleDefName="Azure Image Builder Image Def"+$timeInt
```

```
# Update the JSON definition placeholders with variable values
((Get-Content -path $role_definition -Raw) -replace
'<subscriptionID>',$sub_id) | Set-Content -Path $role_definition
((Get-Content -path $role_definition -Raw) -replace '<rgName>',
$imageResourceGroup) | Set-Content -Path $role_definition
((Get-Content -path $role_definition -Raw) -replace 'Azure Image Builder
Service Image Creation Role', $imageRoleDefName) | Set-Content -Path
$role_definition

# Create a custom role from the aibRoleImageCreation.json description file.
New-AzRoleDefinition -InputFile $role_definition

# Get the user-identity properties
$identityNameResourceId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $identityName).Id
$identityNamePrincipalId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $identityName).PrincipalId

# Grant the custom role to the user-assigned managed identity for Azure
Image Builder.
$parameters = @{
    ObjectId = $identityNamePrincipalId
    RoleDefinitionName = $imageRoleDefName
    Scope = '/subscriptions/' + $sub_id + '/resourceGroups/' +
$imageResourceGroup
}

New-AzRoleAssignment @parameters
```

Existing virtual network Azure role example

The following example creates an Azure role to use and distribute an existing virtual network image. You then grant the custom role to the user-assigned managed identity for VM Image Builder.

To simplify the replacement of values in the example, set the following variables first. Replace the placeholder settings to set your variables.

Expand table

Setting	Description
<Subscription ID>	Your Azure subscription ID.
<Resource group>	The virtual network resource group.

PowerShell

```
$sub_id = "<Subscription ID>"
$res_group = "<Resource group>"
$identityName = "aibIdentity"

# Use a web request to download the sample JSON description
$sample_uri="https://raw.githubusercontent.com/azure/azvmimagebuilder/master
/solutions/12_Creating_AIB_Security_Roles/aibRoleNetworking.json"
$role_definition="aibRoleNetworking.json"

Invoke-WebRequest -Uri $sample_uri -Outfile $role_definition -
UseBasicParsing

# Create a unique role name to avoid clashes in the same AAD domain
$timeInt=$(get-date -UFormat "%s")
$networkRoleDefName="Azure Image Builder Network Def"+$timeInt

# Update the JSON definition placeholders with variable values
((Get-Content -path $role_definition -Raw) -replace
'<subscriptionID>',$sub_id) | Set-Content -Path $role_definition
((Get-Content -path $role_definition -Raw) -replace '<vnetRgName>',
$res_group) | Set-Content -Path $role_definition
((Get-Content -path $role_definition -Raw) -replace 'Azure Image Builder
Service Networking Role',$networkRoleDefName) | Set-Content -Path
$role_definition

# Create a custom role from the aibRoleNetworking.json description file
New-AzRoleDefinition -InputFile $role_definition

# Get the user-identity properties
$identityNameResourceId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $identityName).Id
$identityNamePrincipalId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $identityName).PrincipalId

# Assign the custom role to the user-assigned managed identity for Azure
Image Builder
$parameters = @{
    ObjectId = $identityNamePrincipalId
    RoleDefinitionName = $networkRoleDefName
    Scope = '/subscriptions/' + $sub_id + '/resourceGroups/' + $res_group
}

New-AzRoleAssignment @parameters
```

Next steps

Azure VM Image Builder overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure VM Image Builder service DevOps task (preview)

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

In this article, you learn how to use an Azure DevOps task to inject build artifacts into a virtual machine (VM) image, so that you can install and configure your application and operating system.

> **Important**
>
> Azure DevOps task for VM Image Builder is currently in PREVIEW. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

DevOps task versions

At this time, there are two Azure VM Image Builder DevOps tasks:

- *Stable* VM Image Builder task: The latest stable build that's been tested, and reports no General Data Protection Regulation (GDPR) issues.
- *Unstable* VM Image Builder task: We offer a so-called *unstable* task so that you can test the latest updates and features before we release the task code as *stable*. After about a week, if there are no customer-reported or telemetry issues, we promote the task code to *stable*.

Prerequisites

> **Note**
>
> The VM Image Builder task doesn't currently support Windows Restart or running elevated commands as Administrator. That is, the task isn't suitable for Azure Virtual Desktop scenarios or Windows customizations that require those features. To use DevOps with VM Image Builder, nest the template within an Azure Resource Manager task, and use Azure CLI or PowerShell tasks.

Before you begin, you must:

- Install *Stable* DevOps task from Visual Studio Marketplace.
- Have an Azure DevOps Services (formerly Visual Studio Team Services, or VSTS) account, and a Build Pipeline created.
- Register and enable the VM Image Builder feature requirements in the subscription that's used by the pipelines:
  - Azure PowerShell
  - The Azure CLI
- Create a standard Azure storage account in the source image resource group. You can use other resource groups or storage accounts. The storage account is used transfer the build artifacts from the DevOps task to the image.

Azure PowerShell

```
# Azure PowerShell
$timeInt=$(get-date -UFormat "%s")
$storageAccName="aibstorage"+$timeInt
$location=westus
# Create a storage account and blob in the resource group
New-AzStorageAccount -ResourceGroupName $strResourceGroup -Name
$storageAccName -Location $location -SkuName Standard_LRS
```

Azure CLI

```
# The Azure CLI
location=westus
scriptStorageAcc=aibstordot$(date +'%s')
# Create a storage account and blob in the resource group
az storage account create -n $scriptStorageAcc -g $strResourceGroup -l
$location --sku Standard_LRS
```

Add a task to the release pipeline

1. Select **Release Pipeline** > **Edit**.
2. On the User Agent, select the plus sign (+) to add and search for **Image Builder**.
3. Select **Add**.

In the following sections, set the task properties.

Azure subscription

In the dropdown list, select the subscription that you want VM Image Builder to run. Use the subscription where your source images are stored and the images are to be distributed. You need to grant the VM Image Builder contributor access to the subscription or resource group.

Resource group

Use the resource group where the temporary image template artifact will be stored. When you create a template artifact, another temporary VM Image Builder resource group, `IT_<DestinationResourceGroup>_<TemplateName>_guid`, is created. The temporary resource group stores the image metadata, such as scripts. At the end of the task, the image template artifact and temporary VM Image Builder resource group is deleted.

Location

The location is the region where VM Image Builder will run. Only a set number of regions are supported. The source images must be present in this location. For example, if you're using Azure Compute Gallery (formerly Shared Image Gallery), a replica must exist in that region.

Managed identity (required)

VM Image Builder requires a managed identity, which it uses to read source custom images, connect to Azure Storage, and create custom images. For more information, see Learn about VM Image Builder.

Virtual network support

You can configure the created VM to be in a specific virtual network. When you configure the task, provide the resource ID of a pre-existing subnet in the **VNet Configuration (Optional)** input field. Omit the resource ID if no specific virtual network needs to be used. For more information, see Azure VM Image Builder service networking options.

Source

The source images must be of the supported VM Image Builder operating systems. You can choose existing custom images in the same region that VM Image Builder is running

from:

- Managed Image: Pass in the resource ID. For example:

  ```
  JSON

  /subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Compute/images/<imageName>
  ```

- Compute Gallery: Pass in the resource ID of the image version. For example:

  ```
  JSON

  /subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup/providers/Microsoft.Compute/galleries/$sigName/images/$imageDefName/versions/<versionNumber>
  ```

  If you need to get the latest Compute Gallery version, use an Azure PowerShell or Azure CLI task to get it and set a DevOps variable. Use the variable in the VM Image Builder DevOps task. For more information, see the examples in Get the latest image version resource ID.

- (Marketplace) Base image: Use the dropdown list of popular images, which always uses the latest version of the supported operating systems.

  If the base image isn't in the list, you can specify the exact image by using `Publisher:Offer:Sku`.

  (Optional) Base image version: You can supply the version of the image that you want to use. The default version is `latest`.

Customize

The following sections discuss various ways to customize tasks.

Provisioner

Initially, two customizers are supported, Shell and PowerShell. Only inline is supported. If you want to download scripts, you can pass inline commands to do so.

For your operating system, select PowerShell or Shell.

The Windows Update task

For Windows only, the task runs Windows Update at the end of the customizations. It also handles the required reboots.

The task runs the following Windows Update configuration:

JSON

```
    "type": "WindowsUpdate",
    "searchCriteria": "IsInstalled=0",
    "filters": [
        "exclude:$_.Title -like '*Preview*'",
        "include:$true"
```

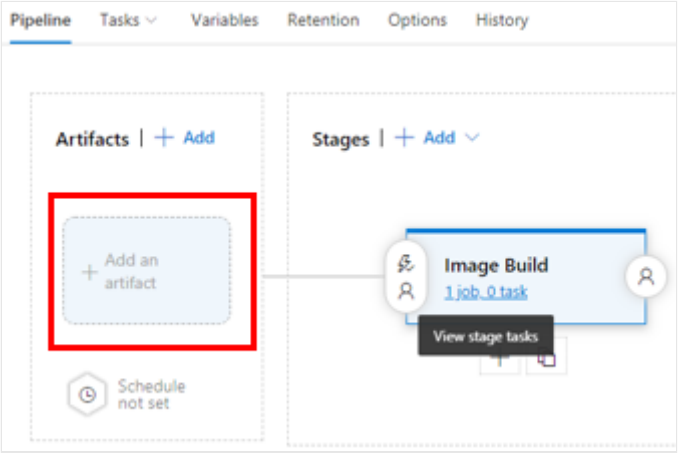
The task installs important and recommended Windows Updates that aren't *preview* versions.

Handling reboots

The DevOps task doesn't currently support rebooting Windows builds. If you try to reboot with PowerShell code, the build fails. However, you can use code to reboot Linux builds.

Build path

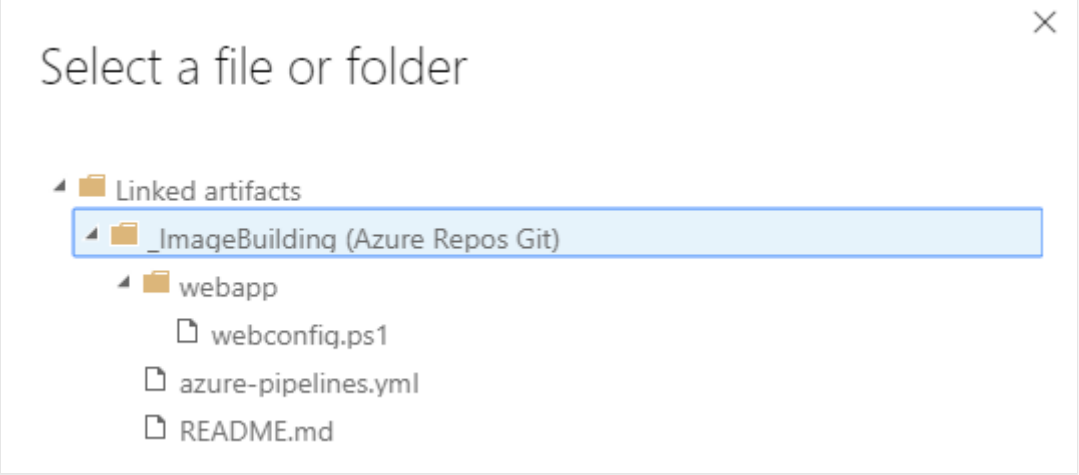
The task is designed to be able to inject DevOps Build release artifacts into the image. To make this work, you need to set up a build pipeline. In the release pipeline setup, add the repo of the build artifacts.



Select the **Build Path** button to choose the build folder that you want to be placed on the image. The VM Image Builder task copies all the files and directories within it. When the image is being created, VM Image Builder deploys the files and directories into different paths, depending on the operating system.

> **Important**
>
> When you're adding a repo artifact, you might find that the directory name is prefixed with an underscore character (_). The underscore can cause issues with the inline commands. Be sure to use the appropriate quotation marks in the commands.

The following example explains how this works:



- For Windows: Files exist in the *C:* drive. A directory named *buildArtifacts* is created, which includes the *webapp* directory.
- For Linux: Files exist in the `/tmp` directory. The `webapp` directory is created, which includes all the files and directories. Because this is a temporary directory, you must move the files out of it. Otherwise, they'll be deleted.

Inline customization script

- For Windows: You can enter PowerShell inline commands, separated by commas. If you want to run a script in your build directory, you can use:

  Azure PowerShell

  ```
  & 'c:\buildArtifacts\webapp\webconfig.ps1'
  ```

  You can reference multiple scripts or add more commands. For example:

  Azure PowerShell

  ```
  & 'c:\buildArtifacts\webapp\webconfig.ps1'
  & 'c:\buildArtifacts\webapp\installAgent.ps1'
  ```

- For Linux: The build artifacts are put into the */tmp* directory. However, on many Linux operating systems, on a reboot, the */tmp* directory contents are deleted. If you want the artifacts to exist in the image, you must create another directory and copy them over. For example:

  Bash

  ```
  sudo mkdir /lib/buildArtifacts
  sudo cp -r "/tmp/_ImageBuilding/webapp" /lib/buildArtifacts/.
  ```

  If you're OK with using the */tmp* directory, you can run the script by using the following code:

  Bash

  ```
  # Grant execute permissions to run scripts
  sudo chmod +x "/tmp/_ImageBuilding/webapp/coreConfig.sh"
  echo "running script"
  sudo . "/tmp/AppsAndImageBuilderLinux/_WebApp/coreConfig.sh"
  ```

What happens to the build artifacts after the image build?

> ⓘ **Note**
>
> VM Image Builder doesn't automatically remove the build artifacts. We strongly suggest that you always use code to remove the build artifacts.

- For Windows: VM Image Builder deploys files to the *C:\buildArtifacts* directory. Because the directory is persisted, you must remove it by running a script. For example:

  Azure PowerShell

  ```
  # Clean up buildArtifacts directory
  Remove-Item -Path "C:\buildArtifacts\*" -Force -Recurse

  # Delete the buildArtifacts directory
  Remove-Item -Path "C:\buildArtifacts" -Force
  ```

- For Linux: The build artifacts are put into the */tmp* directory. However, on many Linux operating systems, the */tmp* directory contents are deleted on reboot. We suggest that you use code to remove the contents and not rely on the operating system to remove the contents. For example:

Bash

```
sudo rm -R "/tmp/AppsAndImageBuilderLinux"
```

Total length of image build

Total length can't be changed in the DevOps pipeline task yet. It uses the default of 240 minutes. If you want to increase the buildTimeoutInMinutes, you can use an Azure CLI task in the release pipeline. Configure the task to copy a template and submit it. For an example solution, see Use environment variables and parameters with VM Image Builder, or use Azure PowerShell.

Storage account

Select the storage account you created in the prerequisites. If you don't see it in the list, VM Image Builder doesn't have permissions to it.

When the build starts, VM Image Builder creates a container called *imagebuilder-vststask*, where the build artifacts from the repo are stored.

> **Note**
>
> You need to manually delete the storage account or container after each build.

Distribute

The following three distribute types are supported.

Managed image

- Resource ID:

  Azure CLI

  ```
  /subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Compute/images/<imageName>
  ```

- Locations

Azure Compute Gallery

The Compute Gallery must already exist.

- Resource ID:

  Azure CLI

  ```
  /subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Compute/galleries/<galleryName>/images/<imageDefName>
  ```

- Regions: A list of regions, comma separated. For example, `westus`, `eastus`, `centralus`.

Virtual hard disk

You can't pass any values to this. VM Image Builder emits the virtual hard disk VHD to the temporary VM Image Builder resource group, `IT_<DestinationResourceGroup>_<TemplateName>`, in the *vhds* container. When you start the release build, VM Image Builder emits logs. When VM Image Builder has finished, it emits the VHD URL.

Optional settings

You can override the VM size setting from its default size of *Standard_D1_v2*. You might want to do so to reduce total customization time. Or you might want to create images that depend on certain VM sizes, such as GPU (graphics processing unit), HPC (high-performance computing), and so on.

How the task works

When you create the release, the task creates a container in the storage account, named *imagebuilder-vststask*. It zips (compresses) and uploads your build artifacts and creates a shared access signature token for the zip file.

The task uses the properties that are passed to the task to create the VM Image Builder template artifact. The task does the following:

- Downloads the build artifact zip file and any other associated scripts. The files are saved in a storage account in the temporary VM Image Builder resource group `IT_<DestinationResourceGroup>_<TemplateName>`.

- Creates a template that's prefixed with *t_* and a 10-digit monotonic integer. The template is saved to the resource group that you selected, and it exists for the

duration of the build in the resource group.

Example output:

Output

```
start reading task parameters...
found build at:  /home/vsts/work/r1/a/_ImageBuilding/webapp
end reading parameters
getting storage account details for aibstordot1556933914
created archive /home/vsts/work/_temp/temp_web_package_21475337782320203.zip
Source for image:  { type: 'SharedImageVersion',
  imageVersionId:
'/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft
.Compute/galleries/<galleryName>/images/<imageDefName>/versions/<imgVersionN
umber>' }
template name:  t_1556938436xxx
starting put template...
```

When the image build starts, the run status is reported in the release logs:

Output

```
starting run template...
```

When the image build finishes, the output is similar to following text:

Output

```
2019-05-06T12:49:52.0558229Z starting run template...
2019-05-06T13:36:33.8863094Z run template:  Succeeded
2019-05-06T13:36:33.8867768Z getting runOutput for  SharedImage_distribute
2019-05-06T13:36:34.6652541Z
================================================================================
==
2019-05-06T13:36:34.6652925Z ## task output variables ##
2019-05-06T13:36:34.6658728Z $(imageUri) =
/subscriptions/<subscriptionID>/resourceGroups/aibwinsig/providers/Microsoft
.Compute/galleries/my22stSIG/images/winWAppimages/versions/0.23760.13763
2019-05-06T13:36:34.6659989Z
================================================================================
==
2019-05-06T13:36:34.6663500Z deleting template t_1557146959485...
2019-05-06T13:36:34.6673713Z deleting storage blob imagebuilder-
vststask\webapp/18-1/webapp_1557146958741.zip
2019-05-06T13:36:34.9786039Z blob imagebuilder-vststask\webapp/18-
1/webapp_1557146958741.zip is deleted
2019-05-06T13:38:37.4884068Z delete template:  Succeeded
```

The image template and `IT_<DestinationResourceGroup>_<TemplateName>` are deleted.

You can take the `$(imageUri)` Azure DevOps Services (formerly Visual Studio Team Services, or VSTS) variable and use it in the next task or just use the value and build a VM.

Output DevOps variables

Here is the publisher, offer, SKU, and version of the source marketplace image:

- `$(pirPublisher)`
- `$(pirOffer)`
- `$(pirSku)`
- `$(pirVersion)`

Here's the image URI, which is the resource ID of the distributed image:

- `$(imageUri)`

FAQ

Can I use an existing image template that I've already created, outside of DevOps?

Not at this time.

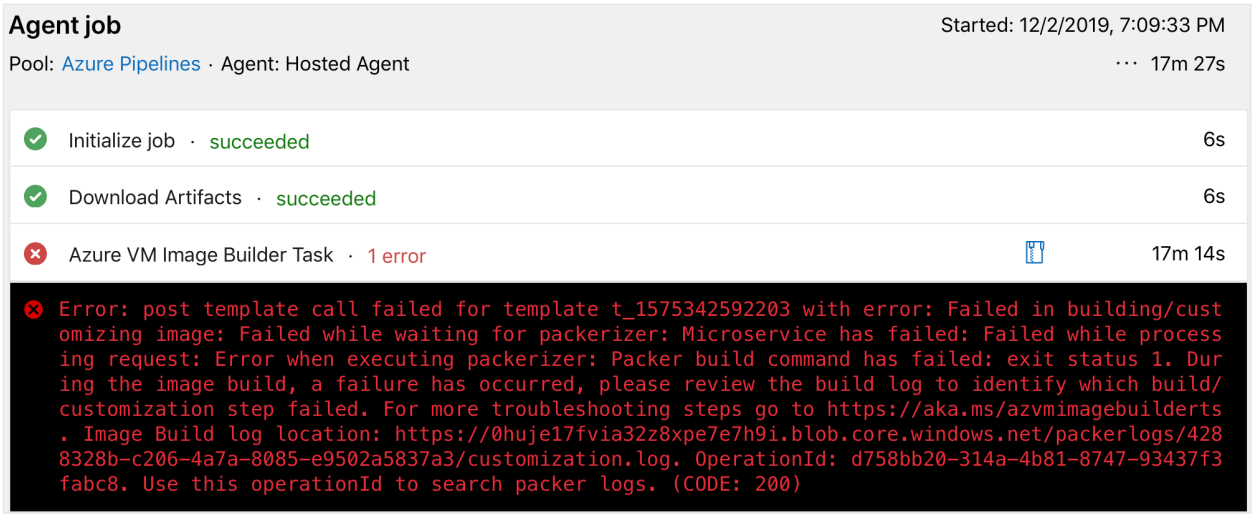
Can I specify the image template name?

No. A unique template name is used and then deleted.

The VM Image Builder task failed. How can I troubleshoot the issue?

If there's a build failure, the DevOps task doesn't delete the staging resource group. You can access the staging resource group that contains the build customization log.

You'll see an error in the DevOps log for the VM Image Builder task, and the message will contain the *customization.log* location. For example:



For more information, see Troubleshoot the VM Image Builder service.

After you've investigated the failure, you can delete the staging resource group. First, delete the VM Image Builder template resource artifact. The artifact is prefixed with *t_*, and you can find it in the DevOps task build log:

```
text

...
Source for image:  { type: 'SharedImageVersion',
  imageVersionId:
'/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft
.Compute/galleries/<galleryName>/images/<imageDefName>/versions/<imgVersionN
umber>' }
...
template name:  t_1556938436xxx
...
```

The VM Image Builder template resource artifact is in the resource group that was specified initially in the task. When you're done troubleshooting, delete the artifact. If you're deleting it by using the Azure portal, within the resource group, select **Show Hidden Types** to view the artifact.

Next steps

For more information, see VM Image Builder overview.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an Azure Image Builder Bicep or ARM template JSON template

Article • 10/03/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

Azure Image Builder uses a Bicep file or an ARM template JSON template file to pass information into the Image Builder service. In this article we go over the sections of the files, so you can build your own. For latest API versions, see template reference. To see examples of full .json files, see the Azure Image Builder GitHub.

The basic format is:

JSON

JSON

```
{
  "type": "Microsoft.VirtualMachineImages/imageTemplates",
  "location": "<region>",
  "tags": {
    "<name>": "<value>",
    "<name>": "<value>"
  },
  "identity": {},
  "properties": {
    "buildTimeoutInMinutes": <minutes>,
    "customize": [],
    "errorHandling":[],
    "distribute": [],
    "optimize": [],
    "source": {},
    "stagingResourceGroup": "/subscriptions/<subscriptionID>/resourceGroups/<stagingResourceGroupName>",
    "validate": {},
    "vmProfile": {
      "vmSize": "<vmSize>",
      "osDiskSizeGB": <sizeInGB>,
      "vnetConfig": {
        "subnetId":
"/subscriptions/<subscriptionID>/resourceGroups/<vnetRgName>/providers/Microsoft.Network/virtualNetworks/<vnetName>/subnets/<subnetName1>",
        "containerInstanceSubnetId":
"/subscriptions/<subscriptionID>/resourceGroups/<vnetRgName>/providers/Microsoft.Network/virtualNetworks/<vnetName>/subnets/<subnetName2>",
        "proxyVmSize": "<vmSize>"
      },
      "userAssignedIdentities": [

"/subscriptions/<subscriptionID>/resourceGroups/<identityRgName>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<identityName1>",

"/subscriptions/<subscriptionID>/resourceGroups/<identityRgName>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<identityName2>",

"/subscriptions/<subscriptionID>/resourceGroups/<identityRgName>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<identityName3>",
        ...
      ]
    }
  }
}
```

API version

The API version will change over time as the API changes. See What's new in Azure VM Image Builder for all major API changes and feature updates for the Azure VM Image Builder service.

Type

The `type` is the resource type, which must be `Microsoft.VirtualMachineImages/imageTemplates`.

JSON

```JSON
"type": "Microsoft.VirtualMachineImages/imageTemplates",
```

Location

The location is the region where the custom image is created. The following regions are supported:

- East US
- East US 2
- West Central US
- West US
- West US 2
- West US 3
- South Central US
- North Europe
- West Europe
- South East Asia
- Australia Southeast
- Australia East
- UK South
- UK West
- Brazil South
- Canada Central
- Central India
- Central US
- France Central
- Germany West Central
- Japan East
- North Central US
- Norway East
- Switzerland North
- Jio India West
- UAE North
- East Asia
- Korea Central
- South Africa North
- Qatar Central
- USGov Arizona (Public Preview)
- USGov Virginia (Public Preview)
- China North 3 (Public Preview)
- Sweden Central
- Poland Central
- Italy North
- Israel Central

Important

Register the feature `Microsoft.VirtualMachineImages/FairfaxPublicPreview` to access the Azure Image Builder public preview in Azure Government regions (USGov Arizona and USGov Virginia).

Important

Register the feature `Microsoft.VirtualMachineImages/MooncakePublicPreview` to access the Azure Image Builder public preview in the China North 3 region.

To access the Azure VM Image Builder public preview in the Azure Government regions (USGov Arizona and USGov Virginia), you must register the *Microsoft.VirtualMachineImages/FairfaxPublicPreview* feature. To do so, run the following command in either PowerShell or Azure CLI:

Azure PowerShell

Azure PowerShell

```
Register-AzProviderPreviewFeature -ProviderNamespace Microsoft.VirtualMachineImages -Name FairfaxPublicPreview
```

To access the Azure VM Image Builder public preview in the China North 3 region, you must register the *Microsoft.VirtualMachineImages/MooncakePublicPreview* feature. To do so, run the following command in either PowerShell or Azure CLI:

Azure PowerShell

Azure PowerShell

```
Register-AzProviderPreviewFeature -ProviderNamespace Microsoft.VirtualMachineImages -Name MooncakePublicPreview
```

JSON

JSON

```
"location": "<region>"
```

Data residency

The Azure VM Image Builder service doesn't store or process customer data outside regions that have strict single region data residency requirements when a customer requests a build in that region. If a service outage for regions that have data residency requirements, you need to create Bicep files/templates in a different region and geography.

Zone redundancy

Distribution supports zone redundancy, VHDs are distributed to a Zone Redundant Storage (ZRS) account by default and the Azure Compute Gallery (formerly known as Shared Image Gallery) version will support a ZRS storage type if specified.

Tags

Tags are key/value pairs you can specify for the image that's generated.

Identity

There are two ways to add user assigned identities explained below.

User-assigned identity for Azure Image Builder image template resource

Required - For Image Builder to have permissions to read/write images, and read in scripts from Azure Storage, you must create an Azure user-assigned identity that has permissions to the individual resources. For details on how Image Builder permissions work, and relevant steps, see Create an image and use a user-assigned managed identity to access files in an Azure storage account.

JSON

JSON

```
"identity": {
    "type": "UserAssigned",
    "userAssignedIdentities": {
        "<imgBuilderId>": {}
    }
}
```

The Image Builder service User Assigned Identity:

- Supports a single identity only.
- Doesn't support custom domain names.

To learn more, see What is managed identities for Azure resources?. For more information on deploying this feature, see Configure managed identities for Azure resources on an Azure VM using Azure CLI.

User-assigned identity for the Image Builder Build VM

This property is only available in API versions `2021-10-01` or newer.

Optional - The Image Builder Build VM that is created by the Image Builder service in your subscription is used to build and customize the image. For the Image Builder Build VM to have permissions to authenticate with other services like Azure Key Vault in your subscription, you must create one or more Azure User Assigned Identities that have permissions to the individual resources. Azure Image Builder can then associate these User Assigned Identities with the Build VM. Customizer scripts running inside the Build VM can then fetch tokens for these identities and interact with other Azure resources as needed. Be aware, the user assigned identity for Azure Image Builder must have the "Managed Identity Operator" role assignment on all the user assigned identities for Azure Image Builder to be able to associate them to the build VM.

> **Note**
>
> Be aware that multiple identities can be specified for the Image Builder Build VM, including the identity you created for the image template resource. By default, the identity you created for the image template resource won't automatically be added to the build VM.

JSON

JSON

```
"properties": {
  "vmProfile": {
    "userAssignedIdentities": [

"/subscriptions/<subscriptionID>/resourceGroups/<identityRgName>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<identityName>"
    ]
  }
}
```

The Image Builder Build VM User Assigned Identity:

- Supports a list of one or more user assigned managed identities to be configured on the VM.
- Supports cross subscription scenarios (identity created in one subscription while the image template is created in another subscription under the same tenant).
- Doesn't support cross tenant scenarios (identity created in one tenant while the image template is created in another tenant).

To learn more, see:

- How to use managed identities for Azure resources on an Azure VM to acquire an access token
- How to use managed identities for Azure resources on an Azure VM for sign-in

Properties: buildTimeoutInMinutes

Maximum duration to wait while building the image template (includes all customizations, validations, and distributions).

If you don't specify the property or set the value to 0, the default value is used, which is 240 minutes or four hours. The minimum value is 6 minutes, and the maximum value is 960 minutes or 16 hours. When the timeout value is hit (whether or not the image build is complete), you see an error similar to:

```text
[ERROR] Failed while waiting for packerizer: Timeout waiting for microservice to
[ERROR] complete: 'context deadline exceeded'
```

For Windows, we don't recommend setting `buildTimeoutInMinutes` below 60 minutes. If you find you're hitting the timeout, review the logs to see if the customization step is waiting on something like user input. If you find you need more time for customizations to complete, increase the `buildTimeoutInMinutes` value. But, don't set it too high because you might have to wait for it to time out before seeing an error.

Properties: customize

Image Builder supports multiple "customizers", which are functions used to customize your image, such as running scripts, or rebooting servers.

When using `customize`:

- You can use multiple customizers.
- Customizers execute in the order specified in the template.
- If one customizer fails, then the whole customization component will fail and report back an error.
- Test the scripts thoroughly before using them in a template. Debugging the scripts by themselves is easier.
- Don't put sensitive data in the scripts. Inline commands can be viewed in the image template definition. If you have sensitive information (including passwords, SAS token, authentication tokens, etc.), it should be moved into scripts in Azure Storage, where access requires authentication.
- The script locations need to be publicly accessible, unless you're using MSI.

The `customize` section is an array. The supported customizer types are: File, PowerShell, Shell, WindowsRestart, and WindowsUpdate.

JSON

```json
"customize": [
  {
    "type": "File",
    "destination": "string",
    "sha256Checksum": "string",
    "sourceUri": "string"
  },
  {
    "type": "PowerShell",
    "inline": [ "string" ],
    "runAsSystem": "bool",
    "runElevated": "bool",
    "scriptUri": "string",
    "sha256Checksum": "string",
    "validExitCodes": [ "int" ]
  },
  {
    "type": "Shell",
    "inline": [ "string" ],
    "scriptUri": "string",
    "sha256Checksum": "string"
  },
  {
    "type": "WindowsRestart",
    "restartCheckCommand": "string",
    "restartCommand": "string",
    "restartTimeout": "string"
  },
  {
    "type": "WindowsUpdate",
    "filters": [ "string" ],
    "searchCriteria": "string",
    "updateLimit": "int"
```

```
  }
]
```

Shell customizer

The `Shell` customizer supports running shell scripts on Linux. The shell scripts must be publicly accessible or you must have configured an MSI for Image Builder to access them.

JSON

```
JSON
```

```
"customize": [
  {
    "type": "Shell",
    "name": "<name>",
    "scriptUri": "<link to script>",
    "sha256Checksum": "<sha256 checksum>"
  }
],
"customize": [
  {
    "type": "Shell",
    "name": "<name>",
    "inline": "<commands to run>"
  }
]
```

Customize properties:

- **type** – Shell.
- **name** - name for tracking the customization.
- **scriptUri** - URI to the location of the file.
- **inline** - array of shell commands, separated by commas.
- **sha256Checksum** - Value of sha256 checksum of the file, you generate this value locally, and then Image Builder will checksum and validate.

  To generate the sha256Checksum, using a terminal on Mac/Linux run: `sha256sum <fileName>`

> ⓘ **Note**
>
> Inline commands are stored as part of the image template definition, you can see these when you dump out the image definition. If you have sensitive commands or values (including passwords, SAS token, authentication tokens etc), it's recommended these are moved into scripts, and use a user identity to authenticate to Azure Storage.

Super user privileges

Prefix the commands with `sudo` to run them with super user privileges. You can add the commands into scripts or use it inline commands, for example:

JSON

```
JSON
```

```
"type": "Shell",
"name": "setupBuildPath",
"inline": [
    "sudo mkdir /buildArtifacts",
    "sudo cp /tmp/index.html /buildArtifacts/index.html"
]
```

Example of a script using sudo that you can reference using scriptUri:

```bash
#!/bin/bash -e

echo "Telemetry: creating files"
mkdir /myfiles

echo "Telemetry: running sudo 'as-is' in a script"
sudo touch /myfiles/somethingElevated.txt
```

Windows restart customizer

The `WindowsRestart` customizer allows you to restart a Windows VM and wait for the VM come back online, this customizer allows you to install software that requires a reboot.

JSON

```json
"customize": [
  {
    "type": "WindowsRestart",
    "restartCommand": "shutdown /r /f /t 0",
    "restartCheckCommand": "echo Azure-Image-Builder-Restarted-the-VM  >
c:\\buildArtifacts\\azureImageBuilderRestart.txt",
    "restartTimeout": "5m"
  }
]
```

Customize properties:

- **Type**: WindowsRestart.
- **restartCommand** - Command to execute the restart (optional). The default is `'shutdown /r /f /t 0 /c \"packer restart\"'`.
- **restartCheckCommand** – Command to check if restart succeeded (optional).
- **restartTimeout** - Restart time out specified as a string of magnitude and unit. For example, `5m` (5 minutes) or `2h` (2 hours). The default is: `5m`.

> **Note**
>
> There is no Linux restart customizer.

PowerShell customizer

The `PowerShell` customizer supports running PowerShell scripts and inline command on Windows, the scripts must be publicly accessible for the IB to access them.

JSON

```json
"customize": [
  {
    "type": "PowerShell",
    "name":   "<name>",
    "scriptUri": "<path to script>",
    "runElevated": <true false>,
    "runAsSystem": <true false>,
    "sha256Checksum": "<sha256 checksum>"
  },
  {
    "type": "PowerShell",
    "name": "<name>",
    "inline": "<PowerShell syntax to run>",
    "validExitCodes": [<exit code>],
    "runElevated": <true or false>,
```

```
    "runAsSystem": <true or false>
  }
]
```

Customize properties:

- **type** – PowerShell.
- **scriptUri** - URI to the location of the PowerShell script file.
- **inline** – Inline commands to be run, separated by commas.
- **validExitCodes** – Optional, valid codes that can be returned from the script/inline command. The property avoids reported failure of the script/inline command.
- **runElevated** – Optional, boolean, support for running commands and scripts with elevated permissions.
- **runAsSystem** - Optional, boolean, determines whether the PowerShell script should be run as the System user.
- **sha256Checksum** - generate the SHA256 checksum of the file locally, update the checksum value to lowercase, and Image Builder will validate the checksum during the deployment of the image template.

  To generate the sha256Checksum, use the Get-FileHash cmdlet in PowerShell.

File customizer

The `File` customizer lets Image Builder download a file from a GitHub repo or Azure storage. The customizer supports both Linux and Windows. If you have an image build pipeline that relies on build artifacts, you can set the file customizer to download from the build share, and move the artifacts into the image.

JSON

JSON

```
"customize": [
  {
    "type": "File",
    "name": "<name>",
    "sourceUri": "<source location>",
    "destination": "<destination>",
    "sha256Checksum": "<sha256 checksum>"
  }
]
```

File customizer properties:

- **sourceUri** - an accessible storage endpoint, this endpoint can be GitHub or Azure storage. You can only download one file, not an entire directory. If you need to download a directory, use a compressed file, then uncompress it using the Shell or PowerShell customizers.

  > ⓘ Note
  >
  > If the sourceUri is an Azure Storage Account, irrespective if the blob is marked public, you'll need to grant the Managed User Identity permissions to read access on the blob. See this **example** to set the storage permissions.

- **destination** – the full destination path and file name. Any referenced path and subdirectories must exist, use the Shell or PowerShell customizers to set up these paths up beforehand. You can use the script customizers to create the path.

This customizer is supported by Windows directories and Linux paths, but there are some differences:

- Linux – the only path Image builder can write to is /tmp.
- Windows – No path restriction, but the path must exist.

If there's an error trying to download the file, or put it in a specified directory, then customize step fails, and this error will be in the customization.log.

> **Note**
>
> The file customizer is only suitable for small file downloads, < 20MB. For larger file downloads, use a script or inline command, then use code to download files, such as, Linux `wget` or `curl`, Windows, `Invoke-WebRequest`. For files that are in Azure storage, ensure that you assign an identity with permissions to view that file to the build VM by following the documentation here: User Assigned Identity for the Image Builder Build VM. Any file that is not stored in Azure must be publicly accessible for Azure Image Builder to be able to download it.

- **sha256Checksum** - generate the SHA256 checksum of the file locally, update the checksum value to lowercase, and Image Builder will validate the checksum during the deployment of the image template.

  To generate the sha256Checksum, use the Get-FileHash cmdlet in PowerShell.

Windows update customizer

The `WindowsUpdate` customizer is built on the community Windows Update Provisioner for Packer, which is an open source project maintained by the Packer community. Microsoft tests and validate the provisioner with the Image Builder service, and will support investigating issues with it, and work to resolve issues, however the open source project isn't officially supported by Microsoft. For detailed documentation on and help with the Windows Update Provisioner, see the project repository.

JSON

```
JSON
"customize": [
  {
    "type": "WindowsUpdate",
    "searchCriteria": "IsInstalled=0",
    "filters": [
      "exclude:$_.Title -like '*Preview*'",
      "include:$true"
    ],
    "updateLimit": 20
  }
]
```

Customizer properties:

- **type** – WindowsUpdate.
- **searchCriteria** - Optional, defines which type of updates are installed (like Recommended or Important), BrowseOnly=0 and IsInstalled=0 (Recommended) is the default.
- **filters** – Optional, allows you to specify a filter to include or exclude updates.
- **updateLimit** – Optional, defines how many updates can be installed, default 1000.

> **Note**
>
> The Windows Update customizer can fail if there are any outstanding Windows restarts, or application installations still running, typically you may see this error in the customization.log, `System.Runtime.InteropServices.COMException (0x80240016): Exception from HRESULT: 0x80240016`. We strongly advise you consider adding in a Windows Restart, and/or allowing applications enough time to complete their installations using sleep or wait commands in the inline commands or scripts before running Windows Update.

Generalize

By default, Azure Image Builder will also run `deprovision` code at the end of each image customization phase, to generalize the image. Generalizing is a process where the image is set up so it can be reused to create multiple VMs. For Windows VMs, Azure Image Builder uses Sysprep. For Linux, Azure Image Builder runs `waagent -deprovision`.

The commands Image Builder users to generalize may not be suitable for every situation, so Azure Image Builder allows you to customize this command, if needed.

If you're migrating existing customization, and you're using different Sysprep/waagent commands, you can use the Image Builder generic commands, and if the VM creation fails, use your own Sysprep or waagent commands.

If Azure Image Builder creates a Windows custom image successfully, and you create a VM from it, then find that the VM creation fails or doesn't complete successfully, you need to review the Windows Server Sysprep documentation or raise a support request with the Windows Server Sysprep Customer Services Support team, who can troubleshoot and advise on the correct Sysprep usage.

Default Sysprep command

Azure PowerShell

```
Write-Output '>>> Waiting for GA Service (RdAgent) to start ...'
while ((Get-Service RdAgent).Status -ne 'Running') { Start-Sleep -s 5 }
Write-Output '>>> Waiting for GA Service (WindowsAzureTelemetryService) to start ...'
while ((Get-Service WindowsAzureTelemetryService) -and ((Get-Service WindowsAzureTelemetryService).Status -ne 'Running'))
{ Start-Sleep -s 5 }
Write-Output '>>> Waiting for GA Service (WindowsAzureGuestAgent) to start ...'
while ((Get-Service WindowsAzureGuestAgent).Status -ne 'Running') { Start-Sleep -s 5 }
if( Test-Path $Env:SystemRoot\system32\Sysprep\unattend.xml ) {
  Write-Output '>>> Removing Sysprep\unattend.xml ...'
  Remove-Item $Env:SystemRoot\system32\Sysprep\unattend.xml -Force
}
if (Test-Path $Env:SystemRoot\Panther\unattend.xml) {
  Write-Output '>>> Removing Panther\unattend.xml ...'
  Remove-Item $Env:SystemRoot\Panther\unattend.xml -Force
}
Write-Output '>>> Sysprepping VM ...'
& $Env:SystemRoot\System32\Sysprep\Sysprep.exe /oobe /generalize /quiet /quit
while($true) {
  $imageState = (Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup\State).ImageState
  Write-Output $imageState
  if ($imageState -eq 'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { break }
  Start-Sleep -s 5
}
Write-Output '>>> Sysprep complete ...'
```

Default Linux deprovision command

Bash

```
WAAGENT=/usr/sbin/waagent
waagent -version 1> /dev/null 2>&1
if [ $? -eq 0 ]; then
  WAAGENT=waagent
fi
$WAAGENT -force -deprovision+user && export HISTSIZE=0 && sync
```

Overriding the Commands

To override the commands, use the PowerShell or Shell script provisioners to create the command files with the exact file name, and put them in the correct directories:

- Windows: c:\DeprovisioningScript.ps1
- Linux: /tmp/DeprovisioningScript.sh

Image Builder reads these commands, these commands are written out to the AIB logs, `customization.log`. See troubleshooting on how to collect logs.

Properties: errorHandling

The `errorHandling` property allows you to configure how errors are handled during image creation.

JSON

JSON

```
{
  "errorHandling": {
    "onCustomizerError": "abort",
    "onValidationError": "cleanup"
```

```
    }
  }
```

The `errorHandling` property allows you to configure how errors are handled during image creation. It has two properties:

- **onCustomizerError** - Specifies the action to take when an error occurs during the customizer phase of image creation.
- **onValidationError** - Specifies the action to take when an error occurs during validation of the image template.

The `errorHandling` property also has two possible values for handling errors during image creation:

- **cleanup** - Ensures that temporary resources created by Packer are cleaned up even if Packer or one of the customizations/validations encounters an error. This maintains backwards compatibility with existing behavior.
- **abort** - In case Packer encounters an error, the Azure Image Builder (AIB) service skips the clean up of temporary resources. As the owner of the AIB template, you are responsible for cleaning up these resources from your subscription. These resources may contain useful information such as logs and files left behind in a temporary VM, which can aid in investigating the error encountered by Packer.

Properties: distribute

Azure Image Builder supports three distribution targets:

- **ManagedImage** - Managed image.
- **sharedImage** - Azure Compute Gallery.
- **VHD** - VHD in a storage account.

You can distribute an image to both of the target types in the same configuration.

> ⓘ **Note**
>
> The default AIB sysprep command doesn't include "/mode:vm", however this property maybe required when create images that will have the HyperV role installed. If you need to add this command argument, you must override the sysprep command.

Because you can have more than one target to distribute to, Image Builder maintains a state for every distribution target that can be accessed by querying the `runOutputName`. The `runOutputName` is an object you can query post distribution for information about that distribution. For example, you can query the location of the VHD, or regions where the image version was replicated to, or SIG Image version created. This is a property of every distribution target. The `runOutputName` must be unique to each distribution target. Here's an example for querying an Azure Compute Gallery distribution:

Azure CLI

```
subscriptionID=<subcriptionID>
imageResourceGroup=<resourceGroup of image template>
runOutputName=<runOutputName>

az resource show \
  --ids
"/subscriptions/$subscriptionID/resourcegroups/$imageResourceGroup/providers/Microsoft.VirtualMachineImages/imageTemplates
/ImageTemplateLinuxRHEL77/runOutputs/$runOutputName" \
--api-version=2023-07-01
```

Output:

Output

```
{
  "id":
"/subscriptions/xxxxxx/resourcegroups/rheltest/providers/Microsoft.VirtualMachineImages/imageTemplates/ImageTemplateLinuxR
HEL77/runOutputs/rhel77",
  "identity": null,
  "kind": null,
  "location": null,
  "managedBy": null,
  "name": "rhel77",
  "plan": null,
  "properties": {
    "artifactId":
"/subscriptions/xxxxxx/resourceGroups/aibDevOpsImg/providers/Microsoft.Compute/galleries/devOpsSIG/images/rhel/versions/0.
24105.52755",
```

```
    "provisioningState": "Succeeded"
  },
  "resourceGroup": "rheltest",
  "sku": null,
  "tags": null,
  "type": "Microsoft.VirtualMachineImages/imageTemplates/runOutputs"
}
```

Distribute: managedImage

The image output is a managed image resource.

JSON

JSON

```
{
  "type":"ManagedImage",
  "imageId": "<resource ID>",
  "location": "<region>",
  "runOutputName": "<name>",
  "artifactTags": {
      "<name>": "<value>",
      "<name>": "<value>"
  }
}
```

Distribute properties:

- **type** – ManagedImage
- **imageId** – Resource ID of the destination image, expected format: /subscriptions/<subscriptionId>/resourceGroups/<destinationResourceGroupName>/providers/Microsoft.Compute/images/<imageName>
- **location** - location of the managed image.
- **runOutputName** – unique name for identifying the distribution.
- **artifactTags** - Optional user specified key\value tags.

> ⓘ **Note**
>
> The destination resource group must exist. If you want the image distributed to a different region, it will increase the deployment time.

Distribute: sharedImage

The Azure Compute Gallery is a new Image Management service that allows managing of image region replication, versioning and sharing custom images. Azure Image Builder supports distributing with this service, so you can distribute images to regions supported by Azure Compute Galleries.

an Azure Compute Gallery is made up of:

- **Gallery** - Container for multiple images. A gallery is deployed in one region.
- **Image definitions** - a conceptual grouping for images.
- **Image versions** - an image type used for deploying a VM or scale set. Image versions can be replicated to other regions where VMs need to be deployed.

Before you can distribute to the gallery, you must create a gallery and an image definition, see Create a gallery.

> ⓘ **Note**
>
> The image version ID needs to be distinct or different from any image versions that are in the existing Azure Compute Gallery.

JSON

JSON

```
{
  "type": "SharedImage",
  "galleryImageId": "<resource ID>",
  "runOutputName": "<name>",
  "artifactTags": {
      "<name>": "<value>",
      "<name>": "<value>"
  }
}
```

The following JSON is an example of how to use the `replicationRegions` field to distribute to an Azure Compute Gallery.

JSON

```
  "replicationRegions": [
      "<region where the gallery is deployed>",
      "<region>"
  ]
```

> **Note**
>
> `replicationRegions` is deprecated for gallery distributions as `targetRegions` is updated property. For more information, see targetRegions.

Distribute: targetRegions

The following JSON is an example of how to use the targetRegions field to distribute to an Azure Compute Gallery.

JSON

```
"distribute": [
      {
        "type": "SharedImage",
        "galleryImageId": "<resource ID>",
        "runOutputName": "<name>",
        "artifactTags": {
          "<name>": "<value>",
          "<name>": "<value>"
        },
        "targetRegions": [
            {
             "name": "eastus",
             "replicaCount": 2,
             "storageAccountType": "Standard_ZRS"
            },
            {
             "name": "eastus2",
             "replicaCount": 3,
             "storageAccountType": "Premium_LRS"
            }
          ]
      },
  ]
```

Distribute properties for galleries:

- **type** - sharedImage
- **galleryImageId** – ID of the Azure Compute Gallery, this property can be specified in two formats:
  - Automatic versioning - Image Builder generates a monotonic version number for you, this property is useful for when you want to keep rebuilding images from the same template: The format is:

`/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Compute/galleries/<sharedImageGalleryName>/images/<imageGalleryName>`.
    - Explicit versioning - You can pass in the version number you want image builder to use. The format is: `/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Compute/galleries/<sharedImageGalName>/images/<imageDefName>/versions/<version - for example: 1.1.1>`

- **runOutputName** – unique name for identifying the distribution.
- **artifactTags** - optional user specified key\value tags.
- **replicationRegions** - array of regions for replication. One of the regions must be the region where the Gallery is deployed. Adding regions mean an increase of build time, as the build doesn't complete until the replication has completed. This field is deprecated as of API version 2022-07-01, please use `targetRegions` when distributing a "SharedImage" type.
- **targetRegions** - an array of regions for replication. It's newly introduced as part of the 2022-07-01 API and applies only to the `SharedImage` type distribute.
- **excludeFromLatest** (optional) - allows you to mark the image version you create not be used as the latest version in the gallery definition, the default is 'false'.
- **storageAccountType** (optional) - AIB supports specifying these types of storage for the image version that is to be created:
    - "Standard_LRS"
    - "Standard_ZRS","

> ⓘ **Note**
>
> If the image template and referenced `image definition` aren't in the same location, you'll see additional time to create images. Image Builder currently doesn't have a `location` parameter for the image version resource, we take it from its parent `image definition`. For example, if an image definition is in `westus` and you want the image version replicated to `eastus`, a blob is copied to `westus`, an image version resource in `westus` is created, and then replicate to `eastus`. To avoid the additional replication time, ensure the `image definition` and image template are in the same location.

versioning

The **versioning** property is for the `sharedImage` distribute type only. It's an enum with two possible values:

- **latest** - New strictly increasing schema per design
- **source** - Schema based upon the version number of the source image.

The default version numbering schema is `latest`. The latest schema has an additional property, “major” which specifies the major version under which to generate the latest version.

> ⓘ **Note**
>
> The existing version generation logic for `sharedImage` distribution is deprecated. Two new options are provided: monotonically increasing versions that are always the latest version in a gallery, and versions generated based on the version number of the source image. The enum specifying the version generation schema allows for expansion in the future with additional version generation schemas.

JSON

```
    "distribute": [
        "versioning": {
            "scheme": "Latest",
            "major": 1
        }
    ]
```

---

versioning properties:

- **scheme** - Generate new version number for distribution. `Latest` or `Source` are two possible values.

- **major** - Specifies the major version under which to generate the latest version. Only applicable when the `scheme` is set to `Latest`. For example, in a gallery with the following versions published: 0.1.1, 0.1.2, 1.0.0, 1.0.1, 1.1.0, 1.1.1, 1.2.0, 2.0.0, 2.0.1, 2.1.0
  - With major not set or major set to 2, The `Latest` scheme generates version 2.1.1
  - With major set to 1, the Latest scheme generates version 1.2.1
  - With major set to 0, the Latest scheme generates version 0.1.3

Distribute: VHD

You can output to a VHD. You can then copy the VHD, and use it to publish to Azure MarketPlace, or use with Azure Stack.

JSON

```
{
  "type": "VHD",
  "runOutputName": "<VHD name>",
  "artifactTags": {
      "<name>": "<value>",
      "<name>": "<value>"
  }
}
```

OS Support: Windows and Linux

Distribute VHD parameters:

- **type** - VHD.
- **runOutputName** – unique name for identifying the distribution.
- **tags** - Optional user specified key value pair tags.

Azure Image Builder doesn't allow the user to specify a storage account location, but you can query the status of the `runOutputs` to get the location.

Azure CLI

```
az resource show \
  --ids
"/subscriptions/$subscriptionId/resourcegroups/<imageResourceGroup>/providers/Microsoft.VirtualMachineImages/imageTemplates/<imageTemplateName>/runOutputs/<runOutputName>"  | grep artifactUri
```

> **Note**
>
> Once the VHD has been created, copy it to a different location, as soon as possible. The VHD is stored in a storage account in the temporary resource group created when the image template is submitted to the Azure Image Builder service. If you delete the image template, then you'll lose the VHD.

The following JSON distributes the image as a VHD to a custom storage account.

JSON

```
"distribute": [
  {
    "type": "VHD",
    "runOutputName": "<VHD name>",
    "artifactTags": {
        "<name>": "<value>",
        "<name>": "<value>"
    },
    "uri": "<replace with Azure storage URI>"
  }
]
```

VHD distribute properties:

uri - Optional Azure Storage URI for the distributed VHD blob. Omit to use the default (empty string) in which case VHD would be published to the storage account in the staging resource group.

Properties: optimize

The `optimize` property can be enabled while creating a VM image and allows VM optimization to improve image creation time.

JSON

```
"optimize": {
      "vmBoot": {
        "state": "Enabled"
      }
    }
```

- **vmBoot**: A configuration related to the booting process of the virtual machine (VM), used to control optimizations that can improve boot time or other performance aspects.
- state: The state of the boot optimization feature within `vmBoot`, with the value `Enabled` indicating that the feature is turned on to improve image creation time.

To learn more, see VM optimization for gallery images with Azure VM Image Builder.

Properties: source

The `source` section contains information about the source image that will be used by Image Builder. Azure Image Builder only supports generalized images as source images, specialized images aren't supported at this time.

The API requires a `SourceType` that defines the source for the image build, currently there are three types:

- PlatformImage - indicated the source image is a Marketplace image.
- ManagedImage - used when starting from a regular managed image.
- SharedImageVersion - used when you're using an image version in an Azure Compute Gallery as the source.

> **Note**
>
> When using existing Windows custom images, you can run the Sysprep command up to three times on a single Windows 7 or Windows Server 2008 R2 image, or 1001 times on a single Windows image for later versions; for more information, see the sysprep documentation.

PlatformImage source

Azure Image Builder supports Windows Server and client, and Linux Azure Marketplace images, see Learn about Azure Image Builder for the full list.

JSON

```
"source": {
  "type": "PlatformImage",
  "publisher": "Canonical",
  "offer": "UbuntuServer",
  "sku": "18.04-LTS",
  "version": "latest"
}
```

The properties here are the same that are used to create VM's, using AZ CLI, run the below to get the properties:

Azure CLI

```
az vm image list -l westus -f UbuntuServer -p Canonical --output table --all
```

You can use `latest` in the version, the version is evaluated when the image build takes place, not when the template is submitted. If you use this functionality with the Azure Compute Gallery destination, you can avoid resubmitting the template, and rerun the image build at intervals, so your images are recreated from the most recent images.

Support for market place plan information

You can also specify plan information, for example:

JSON

JSON

```
"source": {
  "type": "PlatformImage",
  "publisher": "RedHat",
  "offer": "rhel-byos",
  "sku": "rhel-lvm75",
  "version": "latest",
  "planInfo": {
    "planName": "rhel-lvm75",
    "planProduct": "rhel-byos",
    "planPublisher": "redhat"
  }
}
```

ManagedImage source

Sets the source image as an existing managed image of a generalized VHD or VM.

> **Note**
>
> The source managed image must be of a supported OS and the image must reside in the same subscription and region as your Azure Image Builder template.

JSON

JSON

```
"source": {
  "type": "ManagedImage",
  "imageId":
"/subscriptions/<subscriptionId>/resourceGroups/{destinationResourceGroupName}/providers/Microsoft.Compute/images/<ima
geName>"
}
```

The `imageId` should be the ResourceId of the managed image. Use `az image list` to list available images.

SharedImageVersion source

Sets the source image as an existing image version in an Azure Compute Gallery.

> **Note**
>
> The source shared image version must be of a supported OS and the image version must reside in the same region as your Azure Image Builder template, if not, replicate the image version to the Image Builder Template region.

JSON

JSON

```
"source": {
  "type": "SharedImageVersion",
  "imageVersionId":
"/subscriptions/<subscriptionId>/resourceGroups/<resourceGroup>/providers/Microsoft.Compute/galleries/<sharedImageGall
eryName>/images/<imageDefinitionName/versions/<imageVersion>"
}
```

- imageVersionId - ARM template resource ID of the image version. When image version name is 'latest', the version is evaluated when the image build takes place. The `imageVersionId` should be the `ResourceId` of the image version. Use az sig image-version list to list image versions.

The following JSON sets the source image as an image stored in a Direct Shared Gallery.

> **Note**
>
> The Direct Shared Gallery is currently in preview availability.

JSON

JSON

```
    source: {
      "type": "SharedImageVersion",
      "imageVersionId": "<replace with resourceId of the image stored in the Direct Shared Gallery>"
    },
```

The following JSON sets the source image as the latest image version for an image stored in an Azure Compute Gallery.

JSON

JSON

```
"properties": {
    "source": {
        "type": "SharedImageVersion",
        "imageVersionId":
"/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Compute/galleries/<azureComput
eGalleryName>/images/<imageDefinitionName>/versions/latest"
    }
},
```

SharedImageVersion properties:

imageVersionId - ARM template resource ID of the image version. When the image version name is 'latest', the version is evaluated when the image build takes place.

Properties: stagingResourceGroup

The `stagingResourceGroup` property contains information about the staging resource group that the Image Builder service creates for use during the image build process. The `stagingResourceGroup` is an optional property for anyone who wants more control over the resource group created by Image Builder during the image build process. You can create your own resource group and specify it in the `stagingResourceGroup` section or have Image Builder create one on your behalf.

> **Important**
>
> The staging resource group specified cannot be associated with another image template, must be empty (no resources inside), in the same region as the image template, and have either "Contributor" or "Owner" RBAC applied to the identity assigned to the Azure Image Builder image template resource. Image Builder checks the tags on the staging resource group with keys `imageTemplateResourceGroupName` and `imageTemplateName` to determine if any image template uses the staging resource group. If these

tags exist before the image template submission or they do not match the running image template during the image build, the operation will fail.

JSON

```
"properties": {
  "stagingResourceGroup": "/subscriptions/<subscriptionID>/resourceGroups/<stagingResourceGroupName>"
}
```

Template creation scenarios

- **The stagingResourceGroup property is left empty**

  If the `stagingResourceGroup` property isn't specified or specified with an empty string, the Image Builder service creates a staging resource group with the default name convention "IT_***". The staging resource group has the default tags applied to it: `createdBy`, `imageTemplateName`, `imageTemplateResourceGroupName`. Also, the default RBAC is applied to the identity assigned to the Azure Image Builder template resource, which is "Contributor".

- **The stagingResourceGroup property is specified with a resource group that exists**

  If the `stagingResourceGroup` property is specified with a resource group that does exist, then the Image Builder service checks to make sure the resource group isn't associated with another image template, is empty (no resources inside), in the same region as the image template, and has either "Contributor" or "Owner" RBAC applied to the identity assigned to the Azure Image Builder image template resource. If any of the aforementioned requirements aren't met, an error is thrown. The staging resource group has the following tags added to it: `usedBy`, `imageTemplateName`, `imageTemplateResourceGroupName`. Pre-existing tags aren't deleted.

> **Important**
>
> You will need to assign the contributor role to the resource group for the service principal corresponding to Azure Image Builder's first party app when trying to specify a pre-existing resource group and VNet to the Azure Image Builder service with a Windows source image. For the CLI command and portal instructions on how to assign the contributor role to the resource group see the following documentation [Troubleshoot VM Azure Image Builder: Authorization error creating disk]

- **The stagingResourceGroup property is specified with a resource group that doesn't exist**

  If the `stagingResourceGroup` property is specified with a resource group that doesn't exist, then the Image Builder service creates a staging resource group with the name provided in the `stagingResourceGroup` property. There will be an error if the given name doesn't meet Azure naming requirements for resource groups. The staging resource group has the default tags applied to it: `createdBy`, `imageTemplateName`, `imageTemplateResourceGroupName`. By default the identity assigned to the Azure Image Builder image template resource has the "Contributor" RBAC applied to it in the resource group.

Template deletion

Any staging resource group created by the Image Builder service will be deleted after the image template is deleted. The deletion includes staging resource groups that were specified in the `stagingResourceGroup` property, but didn't exist prior to the image build.

If Image Builder didn't create the staging resource group, but the resources inside of the resource group, those resources will be deleted after the image template is deleted, given the Image Builder service has the appropriate permissions or role required to delete resources.

Properties: validate

You can use the `validate` property to validate platform images and any customized images you create regardless of if you used Azure Image Builder to create them.

Azure Image Builder supports a 'Source-Validation-Only' mode that can be set using the `sourceValidationOnly` property. If the `sourceValidationOnly` property is set to true, the image specified in the `source` section will directly be validated. No separate build will be run to generate and then validate a customized image.

The `inVMValidations` property takes a list of validators that will be performed on the image. Azure Image Builder supports File, PowerShell and Shell validators.

The `continueDistributeOnFailure` property is responsible for whether the output image(s) will be distributed if validation fails. By default, if validation fails and this property is set to false, the output image(s) won't be distributed. If validation fails and this property is set to true, the output image(s) will still be distributed. Use this option with caution as it may result in failed images being distributed for use. In either case (true or false), the end to end image run will be reported as a failed if a validation failure. This property has no effect on whether validation succeeds or not.

When using `validate`:

- You can use multiple validators.
- Validators execute in the order specified in the template.
- If one validator fails, then the whole validation component will fail and report back an error.
- It's advised you test the script thoroughly before using it in a template. Debugging the script on your own VM will be easier.
- Don't put sensitive data in the scripts.
- The script locations need to be publicly accessible, unless you're using MSI.

How to use the `validate` property to validate Windows images:

JSON

JSON

```
{
   "properties":{
      "validate":{
         "continueDistributeOnFailure":false,
         "sourceValidationOnly":false,
         "inVMValidations":[
            {
               "type":"File",
               "destination":"string",
               "sha256Checksum":"string",
               "sourceUri":"string"
            },
            {
               "type":"PowerShell",
               "name":"test PowerShell validator inline",
               "inline":[
                  "<command to run inline>"
               ],
               "validExitCodes":"<exit code>",
               "runElevated":"<true or false>",
               "runAsSystem":"<true or false>"
            },
            {
               "type":"PowerShell",
               "name":"<name>",
               "scriptUri":"<path to script>",
               "runElevated":"<true false>",
               "sha256Checksum":"<sha256 checksum>"
            }
         ]
      }
   }
}
```

`inVMValidations` properties:

- **type** – PowerShell.
- **name** - name of the validator
- **scriptUri** - URI of the PowerShell script file.
- **inline** – array of commands to be run, separated by commas.
- **validExitCodes** – Optional, valid codes that can be returned from the script/inline command, this avoids reported failure of the script/inline command.
- **runElevated** – Optional, boolean, support for running commands and scripts with elevated permissions.

- **sha256Checksum** - Value of sha256 checksum of the file, you generate this locally, and then Image Builder will checksum and validate.

  To generate the sha256Checksum, using a PowerShell on Windows Get-Hash

How to use the `validate` property to validate Linux images:

JSON

JSON

```
{
  "properties": {
    "validate": {
      "continueDistributeOnFailure": false,
      "sourceValidationOnly": false,
      "inVMValidations": [
        {
          "type": "Shell",
          "name": "<name>",
          "inline": [
            "<command to run inline>"
          ]
        },
        {
          "type": "Shell",
          "name": "<name>",
          "scriptUri": "<path to script>",
          "sha256Checksum": "<sha256 checksum>"
        },
        {
          "type": "File",
          "destination": "string",
          "sha256Checksum": "string",
          "sourceUri": "string"
        }
      ]
    }
  }
}
```

`inVMValidations` properties:

- **type** – Shell or File specified as the validation type to be performed.
- **name** - name of the validator
- **scriptUri** - URI of the script file
- **inline** - array of commands to be run, separated by commas.
- **sha256Checksum** - Value of sha256 checksum of the file, you generate this locally, and then Image Builder will checksum and validate.

  To generate the sha256Checksum, using a terminal on Mac/Linux run: `sha256sum <fileName>`
- **destination** - Destination of the file.
- **sha256Checksum** - Specifies the SHA256 checksum of the file.
- **sourceUri** - The source URI of the file.

Properties: vmProfile

vmSize (optional)

Image Builder uses a default SKU size of `Standard_D1_v2` for Gen1 images and `Standard_D2ds_v4` for Gen2 images. The generation is defined by the image you specify in the `source`. You can override vmSize for these reasons:

- Performing customizations that require increased memory, CPU and handling large files (GBs).

- Running Windows builds, you should use "Standard_D2_v2" or equivalent VM size.
- Require VM isolation.
- Customize an image that requires specific hardware. For example, for a GPU VM, you need a GPU VM size.
- Require end to end encryption at rest of the build VM, you need to specify the support build VM size that doesn't use local temporary disks.

osDiskSizeGB

By default, Image Builder doesn't change the size of the image, it uses the size from the source image. You can optionally **only** increase the size of the OS Disk (Win and Linux), and a value of 0 means leaving the same size as the source image. You can't reduce the OS Disk size to smaller than the size from the source image.

JSON

JSON

```
{
  "osDiskSizeGB": 100
}
```

vnetConfig (optional)

If you don't specify any VNet properties, Image Builder creates its own VNet, Public IP, and network security group (NSG). The Public IP is used for the service to communicate with the build VM. If you don't want to have a Public IP or you want Image Builder to have access to your existing VNet resources, such as configuration servers (DSC, Chef, Puppet, Ansible), file shares, then you can specify a VNet. For more information, review the networking documentation.

JSON

JSON

```
"vnetConfig": {
  "subnetId":
"/subscriptions/<subscriptionID>/resourceGroups/<vnetRgName>/providers/Microsoft.Network/virtualNetworks/<vnetName>/su
bnets/<subnetName1>",
  "containerInstanceSubnetId":
"/subscriptions/<subscriptionID>/resourceGroups/<vnetRgName>/providers/Microsoft.Network/virtualNetworks/<vnetName>/su
bnets/<subnetName2>",
  "proxyVmSize": "<vmSize>"
}
```

subnetId

Resource ID of a pre-existing subnet on which the build VM and validation VM is deployed.

containerInstanceSubnetId (optional)

Resource ID of a pre-existing subnet on which Azure Container Instance (ACI) is deployed for Isolated Builds. If this field isn't specified, then a temporary Virtual Network, along with subnets and Network Security Groups, is deployed in the staging resource group in addition to other networking resources (Private Endpoint, Private Link Service, Azure Load Balancer, and the Proxy VM) to enable communication between the ACI and the build VM.

[This property is only available in API versions `2024-02-01` or newer though existing templates created using earlier API versions can be updated to specify this property.]

This field can be specified only if `subnetId` is also specified and must meet the following requirements:

- This subnet must be on the same Virtual Network as the subnet specified in `subnetId`.
- This subnet must not be the same subnet as the one specified in `subnetId`.
- This subnet must be delegated to the ACI service so that it can be used to deploy ACI resources. You can read more about subnet delegation for Azure services here. ACI specific subnet delegation information is available here.

- This subnet must allow outbound access to the Internet and to the subnet specified in `subnetId`. These accesses are required so that the ACI can be provisioned and it can communicate with the build VM to perform customizations/validations. On the other end, the subnet specified in `subnetId` must allow inbound access from this subnet. In general, default security rules of Azure Network Security Groups (NSGs) allow these accesses. However, if you add more security rules to your NSGs then the following accesses must still be allowed:

    1. Outbound access from the subnet specified in `containerInstanceSubnetId` to:
        a. To the Internet on port 443 (*for provisioning the container image*).
        b. To the Internet on port 445 (*for mounting file share from Azure Storage*).
        c. To the subnet specified in `subnetId` on port 22 (for ssh/Linux) and port 5986 (for WinRM/Windows) (*for connecting to the build VM*).
    2. Inbound access to the subnet specified in `subnetId`:
        a. To Port 22 (for ssh/Linux) and Port 5986 (for WinRM/Windows) from the subnet specified in `containerInstanceSubnetId` (*for ACI to connect to the build VM*).

- The template identity must have permission to perform 'Microsoft.Network/virtualNetworks/subnets/join/action' action on this subnet's scope. You can read more about Azure permissions for Networking here.

proxyVmSize (optional)

Size of the proxy virtual machine used to pass traffic to the build VM and validation VM. This field must not be specified if `containerInstanceSubnetId` is specified because no proxy virtual machine is deployed in that case. Omit or specify empty string to use the default (Standard_A1_v2).

Properties: autoRun

You can use the `autoRun` property to control whether the image template build process should automatically start when the template is created. It's an enum with two possible values:

- **Enabled** - Auto run is enabled, so your image template build process will automatically start when the template is created.
- **Disabled** - Auto run is disabled, so you will have to manually start the image build process after the template is created.

JSON

```
"properties": {
    "autoRun": {
        "state": "Enabled"
    }
 }
```

> **Note**
>
> When you set `autoRun` to "Enabled," the image build process runs **once** upon template creation. It ensures that the initial image build occurs seamlessly. However, it does not provide consistent and ongoing image builds. For consistent and ongoing image builds that run once an image template is updated, see How to use Azure Image Builder triggers to set up an automatic image build.
>
> Unlike `autoRun`, automatic image creation via the Azure Image Builder trigger resource ensures that image builds occur consistently. Whenever there are changes to the template, the Azure Image Builder service will automatically trigger the image build process.
>
> Choose `autoRun` for immediate image builds upon template creation. Opt for automatic image creation when you need ongoing consistency in image builds. Consider your specific requirements and use the appropriate option based on your workflow.

Properties: managedResourceTags

You can use the `managedResourceTags` property to apply tags to the resources that the Azure Image Builder service creates in the staging resource group during the image build. For more information on the staging resource group, see Azure Image Builder Overview

JSON

```
"properties": {
        "managedResourceTags": {
            "tag1": "value1",
```

```
                "tag2": "value2"
            }
}
```

Image Template Operations

Starting an Image Build

To start a build, you need to invoke 'Run' on the Image Template resource, examples of `run` commands:

Azure PowerShell

```
Invoke-AzResourceAction -ResourceName $imageTemplateName -ResourceGroupName $imageResourceGroup -ResourceType
Microsoft.VirtualMachineImages/imageTemplates -ApiVersion "2023-07-01" -Action Run -Force
```

Azure CLI

```
az resource invoke-action \
  --resource-group $imageResourceGroup \
  --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
  -n helloImageTemplateLinux01 \
  --action Run
```

Cancelling an Image Build

If you're running an image build that you believe is incorrect, waiting for user input, or you feel will never complete successfully, then you can cancel the build.

The build can be canceled anytime. If the distribution phase has started you can still cancel, but you need to clean up any images that may not be completed. The cancel command doesn't wait for cancel to complete, monitor `lastrunstatus.runstate` for canceling progress, using these status commands.

Examples of `cancel` commands:

Azure PowerShell

```
Invoke-AzResourceAction -ResourceName $imageTemplateName -ResourceGroupName $imageResourceGroup -ResourceType
Microsoft.VirtualMachineImages/imageTemplates -ApiVersion "2023-07-01" -Action Cancel -Force
```

Azure CLI

```
az resource invoke-action \
  --resource-group $imageResourceGroup \
  --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
  -n helloImageTemplateLinux01 \
  --action Cancel
```

Next steps

There are sample .json files for different scenarios in the Azure Image Builder GitHub.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a Linux image and distribute it to an Azure Compute Gallery

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how you can use the Azure Image Builder, and the Azure CLI, to create an image version in an Azure Compute Gallery (formerly known as Shared Image Gallery), then distribute the image globally. You can also do this using Azure PowerShell.

We'll be using a sample .json template to configure the image. The .json file we're using is here: helloImageTemplateforSIG.json .

To distribute the image to an Azure Compute Gallery, the template uses sharedImage as the value for the `distribute` section of the template.

Register the providers

To use Azure Image Builder, you need to register the feature.

Check your registration.

Azure CLI

```
az provider show -n Microsoft.VirtualMachineImages | grep registrationState
az provider show -n Microsoft.KeyVault | grep registrationState
az provider show -n Microsoft.Compute | grep registrationState
az provider show -n Microsoft.Storage | grep registrationState
az provider show -n Microsoft.Network | grep registrationState
az provider show -n Microsoft.ContainerInstance | grep registrationState
```

If they don't say registered, run the following:

Azure CLI

```
az provider register -n Microsoft.VirtualMachineImages
az provider register -n Microsoft.Compute
az provider register -n Microsoft.KeyVault
az provider register -n Microsoft.Storage
az provider register -n Microsoft.Network
az provider register -n Microsoft.ContainerInstance
```

Set variables and permissions

We'll be using some pieces of information repeatedly, so we'll create some variables to store that information.

Image Builder only supports creating custom images in the same Resource Group as the source managed image. Update the resource group name in this example to be the same resource group as your source managed image.

Azure CLI

```
# Resource group name - we are using ibLinuxGalleryRG in this example
sigResourceGroup=ibLinuxGalleryRG
# Datacenter location - we are using West US 2 in this example
location=westus2
# Additional region to replicate the image to - we are using East US in this example
additionalregion=eastus
# name of the Azure Compute Gallery - in this example we are using myGallery
sigName=myIbGallery
# name of the image definition to be created - in this example we are using myImageDef
imageDefName=myIbImageDef
# image distribution metadata reference name
runOutputName=aibLinuxSIG
```

Create a variable for your subscription ID.

Azure CLI

```
subscriptionID=$(az account show --query id --output tsv)
```

Create the resource group.

Azure CLI

```
az group create -n $sigResourceGroup -l $location
```

Create a user-assigned identity and set permissions on the resource group

Image Builder uses the user-identity provided to inject the image into the Azure Compute Gallery. In this example, you'll create an Azure role definition that has the

granular actions to perform distributing the image to the gallery. The role definition will then be assigned to the user-identity.

Azure CLI

```
# create user assigned identity for image builder to access the storage
account where the script is located
identityName=aibBuiUserId$(date +'%s')
az identity create -g $sigResourceGroup -n $identityName

# get identity id
imgBuilderCliId=$(az identity show -g $sigResourceGroup -n $identityName --
query clientId -o tsv)

# get the user identity URI, needed for the template
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$sigResourceGroup
/providers/Microsoft.ManagedIdentity/userAssignedIdentities/$identityName

# this command will download an Azure role definition template, and update
the template with the parameters specified earlier.
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12
_Creating_AIB_Security_Roles/aibRoleImageCreation.json -o
aibRoleImageCreation.json

imageRoleDefName="Azure Image Builder Image Def"$(date +'%s')

# update the definition
sed -i -e "s/<subscriptionID>/$subscriptionID/g" aibRoleImageCreation.json
sed -i -e "s/<rgName>/$sigResourceGroup/g" aibRoleImageCreation.json
sed -i -e "s/Azure Image Builder Service Image Creation
Role/$imageRoleDefName/g" aibRoleImageCreation.json

# create role definitions
az role definition create --role-definition ./aibRoleImageCreation.json

# grant role definition to the user assigned identity
az role assignment create \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope /subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup
```

Create an image definition and gallery

To use Image Builder with an Azure Compute Gallery, you need to have an existing gallery and image definition. Image Builder won't create the gallery and image definition for you.

If you don't already have a gallery and image definition to use, start by creating them. First, create a gallery.

Azure CLI

```
az sig create \
    -g $sigResourceGroup \
    --gallery-name $sigName
```

Then, create an image definition.

Azure CLI

```
az sig image-definition create \
   -g $sigResourceGroup \
   --gallery-name $sigName \
   --gallery-image-definition $imageDefName \
   --publisher myIbPublisher \
   --offer myOffer \
   --sku 18.04-LTS \
   --os-type Linux
```

Download and configure the .json

Download the .json template and configure it with your variables.

Azure CLI

```
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickquickst
arts/1_Creating_a_Custom_Linux_Shared_Image_Gallery_Image/helloImageTemplate
forSIG.json -o helloImageTemplateforSIG.json
sed -i -e "s/<subscriptionID>/$subscriptionID/g"
helloImageTemplateforSIG.json
sed -i -e "s/<rgName>/$sigResourceGroup/g" helloImageTemplateforSIG.json
sed -i -e "s/<imageDefName>/$imageDefName/g" helloImageTemplateforSIG.json
sed -i -e "s/<sharedImageGalName>/$sigName/g" helloImageTemplateforSIG.json
sed -i -e "s/<region1>/$location/g" helloImageTemplateforSIG.json
sed -i -e "s/<region2>/$additionalregion/g" helloImageTemplateforSIG.json
sed -i -e "s/<runOutputName>/$runOutputName/g" helloImageTemplateforSIG.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g" helloImageTemplateforSIG.json
```

Create the image version

This next part will create the image version in the gallery.

Submit the image configuration to the Azure Image Builder service.

Azure CLI

```
az resource create \
    --resource-group $sigResourceGroup \
    --properties @helloImageTemplateforSIG.json \
    --is-full-object \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateforSIG01
```

Start the image build.

Azure CLI

```
az resource invoke-action \
     --resource-group $sigResourceGroup \
     --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
     -n helloImageTemplateforSIG01 \
     --action Run
```

Creating the image and replicating it to both regions can take a while. Wait until this part is finished before moving on to creating a VM.

Create the VM

Create a VM from the image version that was created by Azure Image Builder.

Azure CLI

```
az vm create \
  --resource-group $sigResourceGroup \
  --name myAibGalleryVM \
  --admin-username aibuser \
  --location $location \
  --image
"/subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup/providers/M
icrosoft.Compute/galleries/$sigName/images/$imageDefName/versions/latest" \
  --generate-ssh-keys
```

SSH into the VM.

Azure CLI

```
ssh aibuser@<publicIpAddress>
```

You should see the image was customized with a *Message of the Day* as soon as your SSH connection is established!

Console

```
**************************************************
**            This VM was built from the:       **
**      !! AZURE VM IMAGE BUILDER Custom Image !!    **
**         You have just been Customized :-)      **
**************************************************
```

Clean up resources

If you want to now try recustomizing the image version to create a new version of the same image, skip the next steps and go on to Use Azure Image Builder to create another image version.

This deletes the image that was created, along with all of the other resource files. Make sure you're finished with this deployment before deleting the resources.

When deleting gallery resources, you need delete all of the image versions before you can delete the image definition used to create them. To delete a gallery, you first need to have deleted all of the image definitions in the gallery.

Delete the image builder template.

Azure CLI

```azure-cli
az resource delete \
    --resource-group $sigResourceGroup \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateforSIG01
```

Delete permissions assignments, roles and identity

Azure CLI

```azure-cli
az role assignment delete \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope /subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup

az role definition delete --name "$imageRoleDefName"

az identity delete --ids $imgBuilderId
```

Get the image version created by image builder, this always starts with `0.`, and then delete the image version

Azure CLI

```
sigDefImgVersion=$(az sig image-version list \
   -g $sigResourceGroup \
   --gallery-name $sigName \
   --gallery-image-definition $imageDefName \
   --subscription $subscriptionID --query [].'name' -o json | grep 0. | tr -
d '"')
az sig image-version delete \
   -g $sigResourceGroup \
   --gallery-image-version $sigDefImgVersion \
   --gallery-name $sigName \
   --gallery-image-definition $imageDefName \
   --subscription $subscriptionID
```

Delete the image definition.

Azure CLI

```
az sig image-definition delete \
   -g $sigResourceGroup \
   --gallery-name $sigName \
   --gallery-image-definition $imageDefName \
   --subscription $subscriptionID
```

Delete the gallery.

Azure CLI

```
az sig delete -r $sigName -g $sigResourceGroup
```

Delete the resource group.

Azure CLI

```
az group delete -n $sigResourceGroup -y
```

Next steps

Learn more about Azure Compute Galleries.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a Windows image and distribute it to an Azure Compute Gallery

Article • 08/22/2024

Applies to: ✔ Windows VMs

In this article, you learn how to use Azure VM Image Builder and Azure PowerShell to create an image version in an Azure Compute Gallery (formerly Shared Image Gallery) and then distribute the image globally. You can also do this by using the Azure CLI.

To configure the image, this article uses a JSON template, which you can find at armTemplateWinSIG.json. You'll download and edit a local version of the template, so you'll also use a local PowerShell session.

To distribute the image to an Azure Compute Gallery, the template uses sharedImage as the value for the `distribute` section of the template.

VM Image Builder automatically runs `Sysprep` to generalize the image. The command is a generic `Sysprep` command, and you can override it if you need to.

Be aware of the number of times you layer customizations. You can run the `Sysprep` command a limited number of times on a single Windows image. After you've reached the `Sysprep` limit, you must re-create your Windows image. For more information, see Limits on how many times you can run Sysprep.

Register the providers

To use VM Image Builder, you need to register the providers.

1. Check your provider registrations. Make sure that each one returns *Registered*.

PowerShell

```
Get-AzResourceProvider -ProviderNamespace
Microsoft.VirtualMachineImages | Format-table -Property
ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.Storage | Format-
table -Property ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.Compute | Format-
table -Property ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.KeyVault | Format-
table -Property ResourceTypes,RegistrationState
Get-AzResourceProvider -ProviderNamespace Microsoft.Network | Format-
table -Property ResourceTypes,RegistrationState
```

```
Get-AzResourceProvider -ProviderNamespace Microsoft.ContainerInstance |
Format-table -Property ResourceTypes,RegistrationState
```

2. If they don't return *Registered*, register the providers by running the following commands:

```PowerShell
Register-AzResourceProvider -ProviderNamespace
Microsoft.VirtualMachineImages
Register-AzResourceProvider -ProviderNamespace Microsoft.Storage
Register-AzResourceProvider -ProviderNamespace Microsoft.Compute
Register-AzResourceProvider -ProviderNamespace Microsoft.KeyVault
Register-AzResourceProvider -ProviderNamespace
Microsoft.ContainerInstance
```

3. Install PowerShell modules:

```powerShell
'Az.ImageBuilder', 'Az.ManagedServiceIdentity' | ForEach-Object
{Install-Module -Name $_ -AllowPrerelease}
```

Create variables

Because you'll be using some pieces of information repeatedly, create some variables to store that information.

Replace the values for the variables, such as `username` and `vmpassword`, with your own information.

```PowerShell
# Get existing context
$currentAzContext = Get-AzContext

# Get your current subscription ID.
$subscriptionID=$currentAzContext.Subscription.Id

# Destination image resource group
$imageResourceGroup="aibwinsig"

# Location
$location="westus"

# Image distribution metadata reference name
$runOutputName="aibCustWinManImg02ro"
```

```
# Image template name
$imageTemplateName="helloImageTemplateWin02ps"

# Distribution properties object name (runOutput).
# This gives you the properties of the managed image on completion.
$runOutputName="winclientR01"

# Create a resource group for the VM Image Builder template and Azure
Compute Gallery
New-AzResourceGroup `
   -Name $imageResourceGroup `
   -Location $location
```

Create a user-assigned identity and set permissions on the resource group

VM Image Builder uses the provided user-identity to inject the image into Azure Compute Gallery. In this example, you create an Azure role definition with specific actions for distributing the image. The role definition is then assigned to the user identity.

PowerShell

```
# setup role def names, these need to be unique
$timeInt=$(get-date -UFormat "%s")
$imageRoleDefName="Azure Image Builder Image Def"+$timeInt
$identityName="aibIdentity"+$timeInt

## Add an Azure PowerShell module to support AzUserAssignedIdentity
Install-Module -Name Az.ManagedServiceIdentity

# Create an identity
New-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -Name
$identityName

$identityNameResourceId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $identityName).Id
$identityNamePrincipalId=$(Get-AzUserAssignedIdentity -ResourceGroupName
$imageResourceGroup -Name $identityName).PrincipalId
```

Assign permissions for the identity to distribute the images

Use this command to download an Azure role definition template, and then update it with the previously specified parameters.

PowerShell

```
$aibRoleImageCreationUrl="https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutions/12_Creating_AIB_Security_Roles/aibRoleImageCreation.json"
$aibRoleImageCreationPath = "aibRoleImageCreation.json"

# Download the configuration
Invoke-WebRequest -Uri $aibRoleImageCreationUrl -OutFile $aibRoleImageCreationPath -UseBasicParsing

((Get-Content -path $aibRoleImageCreationPath -Raw) -replace '<subscriptionID>',$subscriptionID) | Set-Content -Path $aibRoleImageCreationPath
((Get-Content -path $aibRoleImageCreationPath -Raw) -replace '<rgName>', $imageResourceGroup) | Set-Content -Path $aibRoleImageCreationPath
((Get-Content -path $aibRoleImageCreationPath -Raw) -replace 'Azure Image Builder Service Image Creation Role', $imageRoleDefName) | Set-Content -Path $aibRoleImageCreationPath

# Create a role definition
New-AzRoleDefinition -InputFile  ./aibRoleImageCreation.json

# Grant the role definition to the VM Image Builder service principal
New-AzRoleAssignment -ObjectId $identityNamePrincipalId -RoleDefinitionName $imageRoleDefName -Scope "/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
```

> **Note**
>
> If you receive the error "New-AzRoleDefinition: Role definition limit exceeded. No more role definitions can be created," see Troubleshoot Azure RBAC (role-based access control).

Create an Azure Compute Gallery

To use VM Image Builder with an Azure Compute Gallery, you need to have an existing gallery and image definition. VM Image Builder doesn't create the gallery and image definition for you.

If you don't already have a gallery and image definition to use, start by creating them.

PowerShell

```
# Gallery name
$sigGalleryName= "myIBSIG"
```

```
# Image definition name
$imageDefName ="winSvrimage"

# Additional replication region
$replRegion2="eastus"

# Create the gallery
New-AzGallery `
   -GalleryName $sigGalleryName `
   -ResourceGroupName $imageResourceGroup  `
   -Location $location

# Create the image definition
New-AzGalleryImageDefinition `
   -GalleryName $sigGalleryName `
   -ResourceGroupName $imageResourceGroup `
   -Location $location `
   -Name $imageDefName `
   -OsState generalized `
   -OsType Windows `
   -Publisher 'myCompany' `
   -Offer 'WindowsServer' `
   -Sku 'WinSrv2019'
```

Download and configure the template

Download the JSON template and configure it with your variables.

PowerShell

```
$templateFilePath = "armTemplateWinSIG.json"

Invoke-WebRequest `
   -Uri
"https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickquickstarts/1_Creating_a_Custom_Win_Shared_Image_Gallery_Image/armTemplateWinSIG.json" `
   -OutFile $templateFilePath `
   -UseBasicParsing

(Get-Content -path $templateFilePath -Raw ) `
   -replace '<subscriptionID>',$subscriptionID | Set-Content -Path
$templateFilePath
(Get-Content -path $templateFilePath -Raw ) `
   -replace '<rgName>',$imageResourceGroup | Set-Content -Path
$templateFilePath
(Get-Content -path $templateFilePath -Raw ) `
   -replace '<runOutputName>',$runOutputName | Set-Content -Path
$templateFilePath
(Get-Content -path $templateFilePath -Raw ) `
   -replace '<imageDefName>',$imageDefName | Set-Content -Path
```

```
$templateFilePath
(Get-Content -path $templateFilePath -Raw ) `
   -replace '<sharedImageGalName>',$sigGalleryName | Set-Content -Path
$templateFilePath
(Get-Content -path $templateFilePath -Raw ) `
   -replace '<region1>',$location | Set-Content -Path $templateFilePath
(Get-Content -path $templateFilePath -Raw ) `
   -replace '<region2>',$replRegion2 | Set-Content -Path $templateFilePath
((Get-Content -path $templateFilePath -Raw) -replace
'<imgBuilderId>',$identityNameResourceId) | Set-Content -Path
$templateFilePath
```

Create the image version

Your template must be submitted to the service. The following commands will download any dependent artifacts, such as scripts, and store them in the staging resource group, which is prefixed with *IT_*.

PowerShell

```
New-AzResourceGroupDeployment `
   -ResourceGroupName $imageResourceGroup `
   -TemplateFile $templateFilePath `
   -ApiVersion "2022-02-14" `
   -imageTemplateName $imageTemplateName `
   -svclocation $location
```

To build the image, invoke 'Run' on the template.

PowerShell

```
Invoke-AzResourceAction `
   -ResourceName $imageTemplateName `
   -ResourceGroupName $imageResourceGroup `
   -ResourceType Microsoft.VirtualMachineImages/imageTemplates `
   -ApiVersion "2022-02-14" `
   -Action Run
```

Creating the image and replicating it to both regions can take a few moments. Before you begin creating a VM, wait until this part is finished.

PowerShell

```
Get-AzImageBuilderTemplate -ImageTemplateName $imageTemplateName -
ResourceGroupName $imageResourceGroup |
```

```
  Select-Object -Property Name, LastRunStatusRunState, LastRunStatusMessage,
ProvisioningState
```

Create the VM

Create a VM from the image version that you created with VM Image Builder.

1. Get the image version that you created:

   PowerShell

   ```
   $imageVersion = Get-AzGalleryImageVersion `
   -ResourceGroupName $imageResourceGroup `
   -GalleryName $sigGalleryName `
   -GalleryImageDefinitionName $imageDefName
   $imageVersionId = $imageVersion.Id
   ```

2. Create the VM in the second region, where the image was replicated:

   PowerShell

   ```
   $vmResourceGroup = "myResourceGroup"
   $vmName = "myVMfromImage"

   # Create user object
   $cred = Get-Credential -Message "Enter a username and password for the
   virtual machine."

   # Create a resource group
   New-AzResourceGroup -Name $vmResourceGroup -Location $replRegion2

   # Network pieces
   $subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -
   AddressPrefix 192.168.1.0/24
   $vnet = New-AzVirtualNetwork -ResourceGroupName $vmResourceGroup -
   Location $replRegion2 `
   -Name MYvNET -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
   $pip = New-AzPublicIpAddress -ResourceGroupName $vmResourceGroup -
   Location $replRegion2 `
   -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -
   IdleTimeoutInMinutes 4
   $nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name
   myNetworkSecurityGroupRuleRDP  -Protocol Tcp `
   -Direction Inbound -Priority 1000 -SourceAddressPrefix * -
   SourcePortRange * -DestinationAddressPrefix * `
   -DestinationPortRange 3389 -Access Deny
   $nsg = New-AzNetworkSecurityGroup -ResourceGroupName $vmResourceGroup -
   Location $replRegion2 `
   -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
   $nic = New-AzNetworkInterface -Name myNic -ResourceGroupName
   ```

```
$vmResourceGroup -Location $replRegion2 `
-SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -
NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration using $imageVersion.Id to
specify the image
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1_v2 | `
Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential
$cred | `
Set-AzVMSourceImage -Id $imageVersion.Id | `
Add-AzVMNetworkInterface -Id $nic.Id

# Create a virtual machine
New-AzVM -ResourceGroupName $vmResourceGroup -Location $replRegion2 -VM
$vmConfig
```

Verify the customization

Create a Remote Desktop connection to the VM by using the username and password that you set when you created the VM. In the VM, open a Command Prompt window and run the following command:

Console

```
dir c:\
```

You should see a directory named `buildActions` that was created during image customization.

Clean up your resources

> **Note**
>
> If you now want to try to recustomize the image version to create a new version of the same image, *skip the step outlined here* and go to Use VM Image Builder to create another image version.

If you no longer need the resources that you created as you followed the process in this article, you can delete them.

The following process deletes both the image that you created and all the other resource files. Make sure that you've finished this deployment before you delete the resources.

Delete the resource group template first. Otherwise, the staging resource group (*IT_*) that VM Image Builder uses won't be cleaned up.

1. Get the ResourceID of the image template.

```powerShell
$resTemplateId = Get-AzResource -ResourceName $imageTemplateName -
ResourceGroupName $imageResourceGroup -ResourceType
Microsoft.VirtualMachineImages/imageTemplates -ApiVersion "2022-02-14"
```

2. Delete image template.

```powerShell
Remove-AzResource -ResourceId $resTemplateId.ResourceId -Force
```

3. Delete the role assignment.

```powerShell
Remove-AzRoleAssignment -ObjectId $identityNamePrincipalId -
RoleDefinitionName $imageRoleDefName -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
```

4. Remove the definitions.

```powerShell
Remove-AzRoleDefinition -Name "$identityNamePrincipalId" -Force -Scope
"/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup"
```

5. Delete the identity.

```powerShell
Remove-AzUserAssignedIdentity -ResourceGroupName $imageResourceGroup -
Name $identityName -Force
```

6. Delete the resource group.

```powerShell
Remove-AzResourceGroup $imageResourceGroup -Force
```

Next steps

To update the image version that you created in this article, see Use VM Image Builder to create another image version.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a new VM image from an existing image by using Azure VM Image Builder in Linux

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

In this article, you learn how to update an existing image version in an Azure Compute Gallery (formerly Shared Image Gallery) and publish it to the gallery as a new image version.

To configure the image, you use a sample JSON template, helloImageTemplateforSIGfromSIG.json.

Register the providers

To use VM Image Builder, you need to register the providers.

1. Check your provider registrations. Make sure that each one returns *Registered*.

   Azure CLI

   ```
   az provider show -n Microsoft.VirtualMachineImages | grep
   registrationState
   az provider show -n Microsoft.KeyVault | grep registrationState
   az provider show -n Microsoft.Compute | grep registrationState
   az provider show -n Microsoft.Storage | grep registrationState
   az provider show -n Microsoft.Network | grep registrationState
   az provider show -n Microsoft.ContainerInstance | grep
   registrationState
   ```

2. If they don't return *Registered*, register the providers by running the following commands:

   Azure CLI

   ```
   az provider register -n Microsoft.VirtualMachineImages
   az provider register -n Microsoft.Compute
   az provider register -n Microsoft.KeyVault
   az provider register -n Microsoft.Storage
   az provider register -n Microsoft.Network
   az provider register -n Microsoft.ContainerInstance
   ```

Set variables and permissions

If you've already created an Azure Compute Gallery by using Create an image and distribute it to an Azure Compute Gallery, you've already created some of the variables you need.

1. If you haven't already created the variables, run the following commands:

   Console

   ```
   # Resource group name
   sigResourceGroup=ibLinuxGalleryRG
   # Gallery location
   location=westus2
   # Additional region to replicate the image version to
   additionalregion=eastus
   # Name of the Azure Compute Gallery
   sigName=myIbGallery
   # Name of the image definition to use
   imageDefName=myIbImageDef
   # image distribution metadata reference name
   runOutputName=aibSIGLinuxUpdate
   ```

2. Create a variable for your subscription ID:

   Console

   ```
   subscriptionID=$(az account show --query id --output tsv)
   ```

3. Get the image version that you want to update:

   Azure CLI

   ```
   sigDefImgVersionId=$(az sig image-version list \
     -g $sigResourceGroup \
     --gallery-name $sigName \
     --gallery-image-definition $imageDefName \
     --subscription $subscriptionID --query [].'id' -o tsv)
   ```

Create a user-assigned identity and set permissions on the resource group

You've set up the user identity in an earlier example, so now you need to get the resource ID, which will be appended to the template.

Azure CLI

```
#get identity used previously
imgBuilderId=$(az identity list -g $sigResourceGroup --query "[?
contains(name, 'aibBuiUserId')].id" -o tsv)
```

If you already have an Azure Compute Gallery but didn't set it up by following an earlier example, you need to assign permissions for VM Image Builder to access the resource group so that it can access the gallery. For more information, see Create an image and distribute it to an Azure Compute Gallery.

Modify the helloImage example

You can review the JSON example you're about to use at helloImageTemplateforSIGfromSIG.json. For information about the JSON file, see Create an Azure VM Image Builder template.

1. Download the JSON example, as shown in Create a Linux image and distribute it to an Azure Compute Gallery by using the Azure CLI.

2. Configure the JSON with your variables:

   Console

   ```
   curl
   https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickqu
   ickstarts/8_Creating_a_Custom_Linux_Shared_Image_Gallery_Image_from_SIG
   /helloImageTemplateforSIGfromSIG.json -o
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<subscriptionID>/$subscriptionID/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<rgName>/$sigResourceGroup/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<imageDefName>/$imageDefName/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<sharedImageGalName>/$sigName/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s%<sigDefImgVersionId>%$sigDefImgVersionId%g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<region1>/$location/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<region2>/$additionalregion/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s/<runOutputName>/$runOutputName/g"
   helloImageTemplateforSIGfromSIG.json
   sed -i -e "s%<imgBuilderId>%$imgBuilderId%g"
   helloImageTemplateforSIGfromSIG.json
   ```

Create the image

1. Submit the image configuration to the VM Image Builder service:

   Azure CLI

   ```
   az resource create \
       --resource-group $sigResourceGroup \
       --properties @helloImageTemplateforSIGfromSIG.json \
       --is-full-object \
       --resource-type Microsoft.VirtualMachineImages/imageTemplates \
       -n helloImageTemplateforSIGfromSIG01
   ```

2. Start the image build:

   Azure CLI

   ```
   az resource invoke-action \
       --resource-group $sigResourceGroup \
       --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
       -n helloImageTemplateforSIGfromSIG01 \
       --action Run
   ```

Wait for the image to be built and replicated before you move along to the next step.

Create the VM

1. Create the VM by doing the following:

   Azure CLI

   ```
   az vm create \
   --resource-group $sigResourceGroup \
   --name aibImgVm001 \
   --admin-username azureuser \
   --location $location \
   --image
   "/subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup/providers/Microsoft.Compute/galleries/$sigName/images/$imageDefName/versions/latest" \
   --generate-ssh-keys
   ```

2. Create a Secure Shell (SSH) connection to the VM by using the public IP address of the VM.

   Console

```
ssh azureuser@<pubIp>
```

After the SSH connection is established, you should receive a "Message of the Day" saying that the image was customized:

Output

```
*********************************************************
**            This VM was built from the:              **
**      !! AZURE VM IMAGE BUILDER Custom Image !!      **
**         You have just been Customized :-)           **
*********************************************************
```

3. Type `exit` to close the SSH connection.

4. To list the image versions that are now available in your gallery, run:

Azure CLI

```
az sig image-version list -g $sigResourceGroup -r $sigName -i
$imageDefName -o table
```

Next steps

To learn more about the components of the JSON file that you used in this article, see Create an Azure VM Image Builder template.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a new Windows VM image from an existing image by using Azure VM Image Builder

Article • 08/22/2024

Applies to: ✔ Windows VMs

In this article, you learn how to update an existing Windows image version in an Azure Compute Gallery (formerly Shared Image Gallery) and publish it to the gallery as a new image version.

To configure the image, you use a sample JSON template, helloImageTemplateforSIGfromWinSIG.json .

Register the providers

To use VM Image Builder, you need to register the providers.

1. Check your provider registrations. Make sure that each one returns *Registered*.

   Azure CLI

   ```
   az provider show -n Microsoft.VirtualMachineImages | grep
   registrationState
   az provider show -n Microsoft.KeyVault | grep registrationState
   az provider show -n Microsoft.Compute | grep registrationState
   az provider show -n Microsoft.Storage | grep registrationState
   az provider show -n Microsoft.Network | grep registrationState
   az provider show -n Microsoft.ContainerInstance | grep
   registrationState
   ```

2. If they don't return *Registered*, register the providers by running the following commands:

   Azure CLI

   ```
   az provider register -n Microsoft.VirtualMachineImages
   az provider register -n Microsoft.Compute
   az provider register -n Microsoft.KeyVault
   az provider register -n Microsoft.Storage
   az provider register -n Microsoft.Network
   az provider register -n Microsoft.ContainerInstance
   ```

Set variables and permissions

If you've already created an Azure Compute Gallery by using Create an image and distribute it to an Azure Compute Gallery, you've already created some of the variables you need.

> **Note**
>
> VM Image Builder supports creating custom images only in the same resource group that the source-managed image is in. In the following example, update the resource group name, *ibsigRG*, with the name of resource group that your source-managed image is in.

1. If you haven't already created the variables, run the following commands:

   Azure CLI

   ```
   # Resource group name - we are using ibsigRG in this example
   sigResourceGroup=myIBWinRG
   # Datacenter location - we are using West US 2 in this example
   location=westus
   # Additional region to replicate the image to - we are using East US in this example
   additionalregion=eastus
   # name of the Azure Compute Gallery - in this example we are using myGallery
   sigName=my22stSIG
   # name of the image definition to be created - in this example we are using myImageDef
   imageDefName=winSvrimages
   # image distribution metadata reference name
   runOutputName=w2019SigRo
   # User name and password for the VM
   username="user name for the VM"
   vmpassword="password for the VM"
   ```

2. Create a variable for your subscription ID:

   Azure CLI

   ```
   subscriptionID=$(az account show --query id --output tsv)
   ```

3. Get the image version that you want to update:

   Azure CLI

```
sigDefImgVersionId=$(az sig image-version list \
-g $sigResourceGroup \
--gallery-name $sigName \
--gallery-image-definition $imageDefName \
--subscription $subscriptionID --query [].'id' -o tsv)
```

Create a user-assigned identity and set permissions on the resource group

You've set up the user identity in an earlier example, so now you need to get the resource ID, which will be appended to the template.

Azure CLI

```
#get identity used previously
imgBuilderId=$(az identity list -g $sigResourceGroup --query "[?
contains(name, 'aibBuiUserId')].id" -o tsv)
```

If you already have an Azure Compute Gallery but didn't set it up by following an earlier example, you need to assign permissions for VM Image Builder to access the resource group so that it can access the gallery. For more information, see Create an image and distribute it to an Azure Compute Gallery.

Modify the helloImage example

You can review the JSON example you're about to use at helloImageTemplateforSIGfromSIG.json. For information about the JSON file, see Create an Azure VM Image Builder template.

1. Download the JSON example, as shown in Create a user-assigned identity and set permissions on the resource group.

2. Configure the JSON with your variables:

Azure CLI

```
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickqu
ickstarts/8_Creating_a_Custom_Win_Shared_Image_Gallery_Image_from_SIG/h
elloImageTemplateforSIGfromWinSIG.json -o
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<subscriptionID>/$subscriptionID/g"
helloImageTemplateforSIGfromWinSIG.json
```

```
sed -i -e "s/<rgName>/$sigResourceGroup/g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<imageDefName>/$imageDefName/g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<sharedImageGalName>/$sigName/g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s%<sigDefImgVersionId>%$sigDefImgVersionId%g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<region1>/$location/g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<region2>/$additionalregion/g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s/<runOutputName>/$runOutputName/g"
helloImageTemplateforSIGfromWinSIG.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g"
helloImageTemplateforSIGfromWinSIG.json
```

Create the image

1. Submit the image configuration to the VM Image Builder service:

Azure CLI

```
az resource create \
    --resource-group $sigResourceGroup \
    --location $location \
    --properties @helloImageTemplateforSIGfromWinSIG.json \
    --is-full-object \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n imageTemplateforSIGfromWinSIG01
```

2. Start the image build:

Azure CLI

```
az resource invoke-action \
    --resource-group $sigResourceGroup \
    --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
    -n imageTemplateforSIGfromWinSIG01 \
    --action Run
```

Wait for the image to be built and replicated before you move along to the next step.

Create the VM

Create the VM by doing the following:

Azure CLI

```
az vm create \
--resource-group $sigResourceGroup \
--name aibImgWinVm002 \
--admin-username $username \
--admin-password $vmpassword \
--image
"/subscriptions/$subscriptionID/resourceGroups/$sigResourceGroup/providers/Microsoft.Compute/galleries/$sigName/images/$imageDefName/versions/latest" \
--location $location
```

Verify the customization

Create a Remote Desktop connection to the VM by using the username and password you set when you created the VM. Inside the VM, open a Command Prompt window, and then run:

Console

```
dir c:\
```

You should now see two directories:

- *buildActions*: Created in the first image version.
- *buildActions2*: Created when you updated the first image version to create the second image version.

Next steps

To learn more about the components of the JSON file that you used in this article, see Create an Azure VM Image Builder template.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to enable Automatic Image Creation with Azure Image Builder triggers

Article • 08/29/2024

You can use triggers in Azure Image Builder (AIB) to enable automatic image creation when certain criteria are met in your build pipeline.

> **Important**
>
> Please be informed that there exists a restriction on the number of triggers allowable per region, specifically 100 per region per subscription.

> **Note**
>
> Currently, we only support setting a trigger for a new source image, but we do expect to support different kinds of triggers in the future.

> **Note**
>
> To prevent unnecessary build failures, automatic image creation via triggers will be deactivated if the image template build has failed multiple times consecutively (either manually or automatically triggered). You can still manually build the image template, and once a manual build succeeds, the automatic triggers will be reactivated.

Prerequisites

Before setting up your first trigger, ensure you're using Azure Image Builder API version **2022-07-01**.

How to set up a trigger in Azure Image Builder

Register the providers

To use VM Image Builder with triggers, you need to register the below providers. Check your registration by running the following commands:

Azure CLI

```
az provider show -n Microsoft.VirtualMachineImages -o json | grep
registrationState
az provider show -n Microsoft.KeyVault -o json | grep registrationState
az provider show -n Microsoft.Compute -o json | grep registrationState
az provider show -n Microsoft.Storage -o json | grep registrationState
az provider show -n Microsoft.Network -o json | grep registrationState
az provider show -n Microsoft.ContainerInstance -o json | grep
registrationState
```

If the output doesn't say registered, run the following commands:

Azure CLI

```
az provider register -n Microsoft.VirtualMachineImages
az provider register -n Microsoft.Compute
az provider register -n Microsoft.KeyVault
az provider register -n Microsoft.Storage
az provider register -n Microsoft.Network
az provider register -n Microsoft.ContainerInstance
```

Register the auto image build triggers feature:

Azure CLI

```
az feature register --namespace Microsoft.VirtualMachineImages --name
Triggers
```

Set variables

First, you need to set some variables that you'll repeatedly use in commands.

Azure CLI

```
# Resource group name - ibTriggersTestRG in this example
resourceGroupName=ibTriggersRG
# Datacenter location - West US 2 in this example
location=westus2
# Additional region to replicate the image to - East US in this example
additionalregion=eastus2
# Name of the Azure Compute Gallery - ibTriggersGallery in this example
acgName=ibTriggersGallery
# Name of the image definition to be created - ibTriggersImageDef in this
example
```

```
imageDefName=ibTriggersImageDef
# Name of the Trigger to be created - ibTrigger in this example
ibTriggerName=ibTrigger
# Name of the image template to be created - ibTriggersImageTemplate in this
example
imageTemplateName=ibTriggersImageTemplate
# Reference name in the image distribution metadata
runOutputName=ibTriggersTestRun
# Create a variable for your subscription ID
subscriptionID=$(az account show --query id --output tsv)
```

Create resource group

Now, you need to create a resource group where you can store your image template. Use the following command to make your resource group:

Azure CLI

```
az group create -n $resourceGroupName -l $location
```

Create managed identity for the service

You'll also need to create a managed identity that will be used for the image template (and potentially the Azure Image Builder build VM). In this example, we create a managed identity with "Contributor" access, but you can refine the permissions or role assigned to the managed identity as you like as long as you include the permissions needed for the Azure Image Builder service to function properly.

For more information on the permissions needed for the Azure Image Builder service, see the following documentation: Configure Azure VM Image Builder permissions by using the Azure CLI

For more information on how managed identities can be assigned and used in Azure Image Builder, see the following documentation: VM Image Builder template reference: Identity

Use the following command to create the managed identity that will be used for the image template:

Azure CLI

```
# Create user-assigned identity for VM Image Builder to access the storage
account where the script is stored
identityName=aibBuiUserId$(date +'%s')
az identity create -g $resourceGroupName -n $identityName
```

```
# Get the identity client and principal ID
imgBuilderCliId=$(az identity show -g $resourceGroupName -n $identityName --
query clientId -o tsv)

# Get the user identity URI that's needed for the template
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$resourceGroupNam
e/providers/Microsoft.ManagedIdentity/userAssignedIdentities/$identityName

# Grant "Contributor" access to the user-assigned identity
az role assignment create \
    --assignee $imgBuilderCliId \
    --role "Contributor" \
    --scope /subscriptions/$subscriptionID/resourceGroups/$resourceGroupName
```

Create gallery and image definition

To use VM Image Builder with Azure Compute Gallery, you need to have an existing gallery and image definition. VM Image Builder doesn't create the gallery and image definition for you.

If you don't already have a gallery and image definition to use, start by creating them.

First, create a gallery:

Azure CLI

```
az sig create \
    -g $resourceGroupName \
    --gallery-name $acgName
```

Then, create an image definition:

Azure CLI

```
az sig image-definition create \
   -g $resourceGroupName \
   --gallery-name $acgName \
   --gallery-image-definition $imageDefName \
   --publisher myIbPublisher \
   --offer myOffer \
   --sku 18.04-LTS \
   --os-type Linux
```

Create the image template

Download the example JSON template and configure it with your variables. The following image template uses a Platform Image as its source, but you can change the source to an Azure Compute Gallery image if you'd like to enable automatic image creation anytime there's a new image version in your Azure Compute Gallery.

Azure CLI

```
curl
https://raw.githubusercontent.com/Azure/azvmimagebuilder/main/quickquickstar
ts/9_Setting_up_a_Trigger_with_a_Custom_Linux_Image/helloImageTemplate.json
-o helloImageTemplateforTriggers.json
sed -i -e "s/<subscriptionID>/$subscriptionID/g"
helloImageTemplateforTriggers.json
sed -i -e "s/<rgName>/$resourceGroupName/g"
helloImageTemplateforTriggers.json
sed -i -e "s/<imageDefName>/$imageDefName/g"
helloImageTemplateforTriggers.json
sed -i -e "s/<acgName>/$acgName/g" helloImageTemplateforTriggers.json
sed -i -e "s/<region1>/$location/g" helloImageTemplateforTriggers.json
sed -i -e "s/<region2>/$additionalregion/g"
helloImageTemplateforTriggers.json
sed -i -e "s/<runOutputName>/$runOutputName/g"
helloImageTemplateforTriggers.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g"
helloImageTemplateforTriggers.json
```

Image template requirements:

- The `source` must be either a Platform image or Azure Compute Gallery image (only these two sources are allowed currently)
- If you're using a Platform Image, then the version in the source needs to be `Latest`. For an Azure Compute Gallery image the last part of the resource ID that has the version name needs to be set to `Latest`.
- You can't specify a version if you're distributing the image to an Azure Compute Gallery. The version is automatically incremented.
- When source is set to an Azure Compute Gallery image and distribute is set to an Azure Compute Gallery, the source gallery image and the distribute gallery image can't be the same. The Azure Compute Gallery image definition ID can't be the same for both the source gallery image and the distribute gallery image.
- The image template should have "Succeeded" in the `provisioningState`, meaning the template was created without any issues. If the template isn't provisioned successfully, you won't be able to create a trigger.

After configuring your template use the following command to submit the image configuration to the Azure Image Builder service:

Azure CLI

```
az image builder create -g $resourceGroupName -n $imageTemplateName --image-template helloImageTemplateforTriggers.json
```

You can use the following command to check to make sure the image template was created successfully:

Azure CLI

```
az image builder show --name $imageTemplateName --resource-group $resourceGroupName
```

> **Note**
>
> When running the command above the `provisioningState` should say "Succeeded", which means the template was created without any issues. If the `provisioningState` does not say succeeded, you will not be able to make a trigger use the image template.

Create source trigger

Download the example trigger template and configure it with your variables. The following trigger starts a new image build anytime the source image is updated.

Azure CLI

```
curl https://raw.githubusercontent.com/kof-f/azvmimagebuilder/main/quickquickstarts/9_Setting_up_a_Trigger_with_a_Custom_Linux_Image/trigger.json -o trigger.json
sed -i -e "s/<region1>/$location/g" trigger.json
```

Trigger requirements:

- The location in the trigger needs to be the same as the location in the image template. This is a requirement of the `az resource create` cmdlet.
- We currently support one `kind` of trigger, which is a "SourceImage"
- We only support one "SourceImage" trigger per image. If you already have a "SourceImage" trigger on the image, then you can't create a new one.
- You can't update the `kind` field to another type of trigger. You have to delete the trigger and recreate it or create another trigger with the appropriate configuration.

Use the following command to add the trigger to your resource group.

Azure CLI

```
az image builder trigger create --name $ibTriggerName --resource-group
$resourceGroupName --image-template-name $imageTemplateName --kind
SourceImage
```

You can also use the following command to check that the trigger was created successfully:

Azure CLI

```
az image builder trigger show --name $ibTriggerName --image-template-name
$imageTemplateName --resource-group $resourceGroupName
```

> **Note**
>
> When running the command above the `provisioningState` should say `Succeeded`, which means the trigger was created without any issues. In `status`, the code should say `Healthy` and the message should say `Trigger is active.`

Clean up your resources

Deleting the trigger

Use the following command to delete the trigger:

Azure CLI

```
az image builder trigger delete --name $ibTriggerName --image-template-name
$imageTemplateName --resource-group $resourceGroupName
```

Deleting the image template

Use the following command to delete the image template:

Azure CLI

```
az image builder delete --name $imageTemplateName --resource-group
$resourceGroupName
```

Next steps

Check out the Image Builder template reference for more information.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an image and use a user-assigned managed identity to access files in an Azure storage account

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows how to create a customized image by using Azure VM Image Builder. The service uses a user-assigned managed identity to access files in an Azure storage account, and it can achieve blocking unauthenticated access to the storage account.

Azure VM Image Builder supports using scripts and copying files from GitHub, Azure storage accounts, and other locations. If you want to use the locations, they must be externally accessible to VM Image Builder.

In the following example, you'll create two resource groups, one for the custom image and the other to host an Azure storage account that contains a script file. This example simulates a real-life scenario, where you might have build artifacts or image files in various storage accounts. You'll create a user-assigned identity and then grant the identity read permissions on the script file, but you won't allow public access to the file. You'll then use the shell customizer to download and run a script from the storage account.

Create a resource group

1. Because you'll be using some pieces of information repeatedly, create some variables to store that information.

   Console

   ```
   # Image resource group name
   imageResourceGroup=aibmdimsi
   # Storage resource group
   strResourceGroup=aibmdimsistor
   # Location
   location=WestUS2
   # Name of the image to be created
   imageName=aibCustLinuxImgMsi01
   # Image distribution metadata reference name
   runOutputName=u1804ManImgMsiro
   ```

2. Create a variable for your subscription ID:

Console

```
subscriptionID=$(az account show --query id --output tsv)
```

3. Create resource groups for both the image and the script storage:

Console

```
# Create a resource group for the image template
az group create -n $imageResourceGroup -l $location
# Create a resource group for the script storage
az group create -n $strResourceGroup -l $location
```

4. Create a user-assigned identity, and set permissions on the resource group:

   VM Image Builder uses the provided user identity to inject the image into the resource group. In this example, you create an Azure role definition with specific actions for distributing the image. The role definition is then assigned to the user identity.

Console

```
# Create a user-assigned identity for VM Image Builder to access the
storage account where the script is located
identityName=aibBuiUserId$(date +'%s')
az identity create -g $imageResourceGroup -n $identityName

# Get an identity ID
imgBuilderCliId=$(az identity show -g $imageResourceGroup -n
$identityName --query clientId -o tsv)

# Get the user-identity URI, which is needed for the template
imgBuilderId=/subscriptions/$subscriptionID/resourcegroups/$imageResour
ceGroup/providers/Microsoft.ManagedIdentity/userAssignedIdentities/$ide
ntityName

# Download the preconfigured role definition example
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/solutio
ns/12_Creating_AIB_Security_Roles/aibRoleImageCreation.json -o
aibRoleImageCreation.json

# Update the definition
sed -i -e "s/<subscriptionID>/$subscriptionID/g"
aibRoleImageCreation.json
sed -i -e "s/<rgName>/$imageResourceGroup/g" aibRoleImageCreation.json

# Create role definitions
az role definition create --role-definition ./aibRoleImageCreation.json
```

```
# Grant the role definition to the user-assigned identity
az role assignment create \
    --assignee $imgBuilderCliId \
    --role "Azure Image Builder Service Image Creation Role" \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup
```

5. Create the storage account, and copy the sample script into it from GitHub:

Azure CLI

```
# Script storage account
scriptStorageAcc=aibstorscript$(date +'%s')

# Script container
scriptStorageAccContainer=scriptscont$(date +'%s')

# Script URL
scriptUrl=https://$scriptStorageAcc.blob.core.windows.net/$scriptStorag
eAccContainer/customizeScript.sh

# Create the storage account and blob in the resource group
az storage account create -n $scriptStorageAcc -g $strResourceGroup -l
$location --sku Standard_LRS

az storage container create -n $scriptStorageAccContainer --fail-on-
exist --account-name $scriptStorageAcc

# Copy in an example script from the GitHub repo
az storage blob copy start \
    --destination-blob customizeScript.sh \
    --destination-container $scriptStorageAccContainer \
    --account-name $scriptStorageAcc \
    --source-uri
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickqu
ickstarts/customizeScript.sh
```

6. Give VM Image Builder permission to create resources in the image resource group. The `--assignee` value is the user-identity ID.

Azure CLI

```
az role assignment create \
    --assignee $imgBuilderCliId \
    --role "Storage Blob Data Reader" \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$strResourceGroup/provide
rs/Microsoft.Storage/storageAccounts/$scriptStorageAcc/blobServices/def
ault/containers/$scriptStorageAccContainer
```

Modify the example

Download the example JSON file and configure it with the variables you created earlier.

Console

```
curl
https://raw.githubusercontent.com/azure/azvmimagebuilder/master/quickquickst
arts/7_Creating_Custom_Image_using_MSI_to_Access_Storage/helloImageTemplateM
si.json -o helloImageTemplateMsi.json
sed -i -e "s/<subscriptionID>/$subscriptionID/g" helloImageTemplateMsi.json
sed -i -e "s/<rgName>/$imageResourceGroup/g" helloImageTemplateMsi.json
sed -i -e "s/<region>/$location/g" helloImageTemplateMsi.json
sed -i -e "s/<imageName>/$imageName/g" helloImageTemplateMsi.json
sed -i -e "s%<scriptUrl>%$scriptUrl%g" helloImageTemplateMsi.json
sed -i -e "s%<imgBuilderId>%$imgBuilderId%g" helloImageTemplateMsi.json
sed -i -e "s%<runOutputName>%$runOutputName%g" helloImageTemplateMsi.json
```

Create the image

1. Submit the image configuration to the VM Image Builder service:

   Azure CLI

   ```
   az resource create \
       --resource-group $imageResourceGroup \
       --properties @helloImageTemplateMsi.json \
       --is-full-object \
       --resource-type Microsoft.VirtualMachineImages/imageTemplates \
       -n helloImageTemplateMsi01
   ```

2. Start the image build:

   Azure CLI

   ```
   az resource invoke-action \
       --resource-group $imageResourceGroup \
       --resource-type  Microsoft.VirtualMachineImages/imageTemplates \
       -n helloImageTemplateMsi01 \
       --action Run
   ```

The build can take about 15 minutes to finish.

Create a VM

1. Create a VM from the image:

Azure CLI

```
az vm create \
--resource-group $imageResourceGroup \
--name aibImgVm00 \
--admin-username aibuser \
--image $imageName \
--location $location \
--generate-ssh-keys
```

2. After the VM has been created, start a Secure Shell (SSH) session with it.

Console

```
ssh aibuser@<publicIp>
```

After the SSH connection is established, you should receive a "Message of the Day" saying that the image was customized:

Output

```
*******************************************************
**            This VM was built from the:            **
**      !! AZURE VM IMAGE BUILDER Custom Image !!    **
**         You have just been Customized :-)         **
*******************************************************
```

Clean up your resources

If you no longer need the resources that were created during this process, you can delete them by running the following code:

Azure CLI

```
az role definition delete --name "$imageRoleDefName"
```azurecli-interactive
az role assignment delete \
    --assignee $imgBuilderCliId \
    --role "$imageRoleDefName" \
    --scope
/subscriptions/$subscriptionID/resourceGroups/$imageResourceGroup
az identity delete --ids $imgBuilderId
az resource delete \
```
```

```
    --resource-group $imageResourceGroup \
    --resource-type Microsoft.VirtualMachineImages/imageTemplates \
    -n helloImageTemplateMsi01
az group delete -n $imageResourceGroup
az group delete -n $strResourceGroup
```

Next steps

If you have any problems using VM Image Builder, see Troubleshoot Azure VM Image Builder.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What's new in Azure VM Image Builder (AIB)

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article contains all major API changes and feature updates for the Azure VM Image Builder service.

Updates

March 2026

Breaking Change: Virtual Network Default Behavior

After March 31, 2026, new virtual networks in Azure will default to creation of private subnets which can no longer use default outbound access connectivity. This change may affect customers of AIB that specify their own subnets in their image templates (using `subnetId` field or `containerInstanceSubnetId` field.) If the build or validation VMs needs outbound access then the subnet specified in the `subnetId` field must have outbound access enabled. Also, the subnet specified in `containerInstanceSubnetId` field must have outbound access enabled as explained in the template reference. For complete details, see the official update.

Retirement: Azure unmanaged disks

Azure is retiring Azure unmanaged disks. This change affects a subset of AIB customers who use AIB to distribute images as VHDs and then use those VHDs to provision new VMs. After the unmanaged disks retirement, you will still be able to use AIB to create VHD artifacts, but those VHDs will not be able to provision new VMs. For complete details — including scope, timelines, and the recommended mitigation steps — see the official documentation: Migrate your Azure unmanaged disks by March 31, 2026

September 2024

Automatic image creation via triggers is deactivated if the image template build fails multiple times consecutively. This deactivation avoids unnecessary build failures.

You can still manually build the image template. After a manual build succeeds, the automatic triggers are reactivated.

This behavior is the same regardless of which API version you use for the image template resource.

May 2024

Breaking change: Case sensitivity

As of May 21, 2024, VM Image Builder API version 2024-02-01 and later enforces case sensitivity for all fields. The capitalization of letters in your API requests must match exactly with the expected format.

> **Important**
>
> If you're an existing user of VM Image Builder, this change does *not* affect your existing resources. The enforcement of case sensitivity applies only to newly created resources that use API version 2024-02-01 and later. Your existing resources continue to function as expected without any changes.
>
> If you encounter any problems related to case sensitivity, refer to the updated VM Image Builder API documentation for guidance.

Previously, the VM Image Builder API was more forgiving in terms of case. Moving forward, precision is crucial. When you make API calls, ensure that you use the correct capitalization for field names, parameters, and values. For example, if a field is named `vmBoot`, you must use `vmBoot` (not `VMBoot` or `vmboot`).

If you send an API request to VM Image Builder API version 2024-02-01 or later with incorrect case or unrecognized fields, the service rejects it. You receive an error response indicating that the request is invalid. The error looks something like this example:

```
Unmarshalling entity encountered error: unmarshalling type *v2024_02_01.ImageTemplate: struct field Properties: unmarshalling type *v2024_02_01.ImageTemplateProperties: struct field Optimize: unmarshalling type *v2024_02_01.ImageTemplatePropertiesOptimize: unmarshalling type *v2024_02_01.ImageTemplatePropertiesOptimize, unknown field \"vmboot\". There is an issue with the syntax with the JSON template you are submitting. Please check the JSON template for syntax and grammar. For more information on the syntax and grammar of the JSON template, visit http://aka.ms/azvmimagebuildertmplref.
```

The error message mentions an "unknown field" and directs you to the official documentation: Create an Azure VM Image Builder Bicep or Azure Resource Manager JSON template.

> **Note**
>
> When you're making API calls to the VM Image Builder service, always reference the Swagger documentation. This documentation serves as the definitive source of truth for VM Image Builder API specifications. Although the public documentation was updated to include the proper capitalization and field names ahead of the API release, the Swagger definition contains precise details about each VM Image Builder API. These details help ensure that you're making calls to the service correctly.

The following documentation changes were made to match the field names in API version 2024-02-01.

In the Create an Azure VM Image Builder Bicep or Azure Resource Manager JSON template documentation:

- Fields updated:
  - Replaced several mentions of `vmboot` with `vmBoot`.
  - Replaced one mention of `imageVersionID` with `imageVersionId`.
- Field removed:
  - `apiVersion`: We recommend avoiding the inclusion of this field in your requests because it's not explicitly specified in the API. Including it in your JSON template might lead to errors in your image build.

In the Azure VM Image Builder networking options documentation:

- Field updated:
  - Replaced one mention of `VirtualNetworkConfig` with `vnetConfig`.
- Fields removed:
  - `subnetName` in the `vnetConfig` property: This field is deprecated. The new field is `subnetId`.
  - `resourceGroupName` in the `vnetConfig` property: This field is deprecated. The new field is `subnetId`.

Pinning to an older VM Image Builder API version

If you want to avoid making changes to the properties in your image templates due to the new case-sensitivity rules, you have the option to pin your Azure VM Image Builder API calls to a

previous API version. This pinning allows you to continue using the familiar behavior without any modifications.

> **ⓘ Important**
>
> Pinning to an older VM Image Builder API version can provide compatibility with your existing templates, but we don't recommend it due to the following factors:
>
> - Older API versions might eventually be deprecated.
> - By pinning to an older API version, you miss the latest features and improvements introduced in newer versions. These enhancements often improve performance, security, and functionality.

To ensure compatibility with your existing templates when you're creating or updating an image template, specify the desired API version by including the `api-version` parameter in your call to the service. For example:

HTTP

HTTP

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.VirtualMachineImages/imageTemplates/{imageTemplateName}?api-version=2022-07-01
```

Testing your code

After you pin to the older API version, test your code to verify that it behaves as expected. Ensure that your existing templates continue to function correctly.

November 2023

VM Image Builder is enabling Isolated Image Builds via Azure Container Instances in a phased manner. The rollout is expected finish by early 2024. Your existing image templates continue to work, and there's no change in the way you create or build new image templates.

You might observe a different set of transient Azure resources appearing temporarily in the staging resource group. It doesn't affect your actual builds or the way you interact with VM Image Builder. For more information, see Isolated Image Builds.

To use Isolated Image Builds, make sure that:

- Your subscription is registered for the `Microsoft.ContainerInstance` provider.
- There are no policies blocking deployment of Azure Container Instances resources.
- Quota is available for Azure Container Instances resources.

April 2023

New portal functionality was added for VM Image Builder. Search for **Image Templates** in the Azure portal, and then select **Create**. You can also use this template configuration to get started with building and validating custom images inside the portal.

API releases

Version 2024-02-01

Improvements

- You can use the new `autoRun` property to run the image build on template creation or update. For more information, see Properties: autoRun.
- You can use the new `managedResourceTags` property to apply tags to the resources that the VM Image Builder service creates in the staging resource group during the image build. For more information, see Properties: managedResourceTags.
- You can use the new `containerInstanceSubnetId` property to specify a subnet on which Azure Container Instances will be deployed for Isolated Image Builds. You can specify this field only if you specify `subnetId`. This field must be on the same virtual network as the subnet specified in `subnetId`. For more information, see Bring your own build VM subnet and bring your own Container Instances subnet.
- This version adds support for updating the `vmProfile` property, including the following fields:
  - `vmSize`
  - `osDiskSizeGB`
  - `userAssignedIdentities`
  - `vnetConfig`
    - `subnetId`
    - `containerInstanceSubnetId`

For more information on the `vmProfile` property, see vmProfile.

Changes

API version 2024-02-01 introduces a breaking change that enforces case sensitivity for all fields. The capitalization of letters in your API requests must match exactly with the expected format. If you send an API request to VM Image Builder API version 2024-02-01 or later with an incorrect case or unrecognized fields, the service rejects it. You receive an error response indicating that the request is invalid. For more information, see Breaking change: Case sensitivity in this article.

Version 2023-07-01

Changes

The new `errorHandling` property gives you more control over how errors are handled during the image building process. For more information, see errorHandling.

Version 2022-07-01

Improvements

- This version adds support to use the latest image version stored in Azure Compute Gallery as the source for the image template.
- This version adds `versioning` to support generating version numbers for image distributions. For more information, see Properties: versioning.
- This version adds support for per-region configuration when you're distributing to Azure Compute Gallery. For more information, see Distribute: targetRegions.
- This version adds a new `File` validation type. For more information, see Properties: validate.
- You can now distribute virtual hard disks (VHDs) to a custom blob or container in a custom storage account. For more information, see Distribute: VHD.
- This version adds support for using a direct shared gallery image as the source for the image template.

Changes

- `replicationRegions` is now deprecated for gallery distributions. Instead, use gallery-replication-regions.

- You can now distribute VHDs to a custom blob or container in a custom storage account.
- This version adds the `targetRegions` array, which applies only to the `SharedImage` type of distribution. For more information on `targetRegions`, see Store and share resources in Azure Compute Gallery.
- This version adds support for using a direct shared gallery image as the source for the image template. Direct shared galleries are currently in preview.
- Triggers are now available in preview to set up automatic image builds. For more information, see How to enable automatic image creation with Azure VM Image Builder triggers.

Version 2022-02-14

Improvements

- Validation support
  - Shell (Linux): Script or inline
  - PowerShell (Windows): Script or inline, run elevated, run as system
  - Source-validation-only mode
- Customized support for staging resource groups

Version 2021-10-01

Breaking change

API version 2021-10-01 introduces a change to the error schema that will be part of every future API release. If you have any Azure VM Image Builder automations, be aware of the new error output when you switch to API version 2021-10-01 or later.

We recommend, after you switch to the latest API version, that you don't revert to an earlier version. If you revert, you'll have to change your automation again to produce the earlier error schema. We don't expect the error schema to change again in future releases.

Error output for version 2020-02-14 and earlier

```
{
  "code": "ValidationFailed",
  "message": "Validation failed: 'ImageTemplate.properties.source': Field 'imageId'
has a bad value:
'/subscriptions/subscriptionID/resourceGroups/resourceGroupName/providers/Microsoft
```

```
.Compute/images/imageName'. Please review  http://aka.ms/azvmimagebuildertmplref
for details on fields requirements in the Image Builder Template."
}
```

Error output for version 2021-10-01 and later

```
{
  "error": {
    "code": "ValidationFailed",
    "message": "Validation failed: 'ImageTemplate.properties.source': Field
'imageId' has a bad value:
'/subscriptions/subscriptionID/resourceGroups/resourceGroupName/providers/Microsoft
.Compute/images/imageName'. Please review  http://aka.ms/azvmimagebuildertmplref
for details on fields requirements in the Image Builder Template."
  }
}
```

Improvements

- Added support for managed identities for the build VM.
- Added support for customization of proxy VM size.

Version 2020-02-14

Improvements

- Added support for creating images from the following sources:
  - Managed image
  - Azure Compute Gallery
  - Platform Image Repository (including Platform Image Purchase Plan)
- Added support for the following customizations:
  - Shell (Linux): Script or inline
  - PowerShell (Windows): Script or inline, run elevated, run as system
  - File (Linux and Windows)
  - Windows restart (Windows)
  - Windows Update (Windows): Search criteria, filters, and update limit
- Added support for the following distribution types:
  - VHD
  - Managed image
  - Azure Compute Gallery

- Added support for customers to use their own virtual network.
- Added support for customers to customize the build VM (VM size, operating system disk size).
- Added support for user-assigned managed identities (for customize/distribute steps).
- Added support for Generation 2 images.

Preview API

The following API is deprecated but still supported:

- Version 2019-05-01-preview

Related content

- Azure VM Image Builder overview

Last updated on 02/11/2026

Troubleshoot Azure VM Image Builder

Article • 05/12/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Use this article to troubleshoot and resolve common problems that you might encounter when you're using Azure VM Image Builder.

VM Image Builder failures can happen in two areas:

- During image template submission
- During image building

Prerequisites

These prerequisites are for creating a build:

- The VM Image Builder service communicates to the build virtual machine (VM) by using Windows Remote Management (WinRM) or Secure Shell (SSH). *Don't* disable these settings as part of the build.

- VM Image Builder creates resources in the staging resource group as part of the builds. The exact list of resources depends on the network configuration that the image template specifies. Be sure to verify that Azure Policy doesn't prevent VM Image Builder from creating or using necessary resources.
  - Create an `IT_` resource group.
  - Create a storage account without a firewall.
  - Deploy Azure Container Instances.
  - Deploy Azure Virtual Network resources (and subnets within them).
  - Deploy Azure Private Endpoint resources.
  - Deploy Azure Files.

- Ensure that Azure Policy does not install unintended features on the build VM or other staging resources, such as Azure extensions or tag modifications.

- Ensure that VM Image Builder has the correct permissions to read/write images and to connect to the storage account. For more information, review the permissions documentation for the Azure CLI or Azure PowerShell.

- VM Image Builder fails the build if the scripts or inline commands fail with errors (nonzero exit codes). Ensure that you tested the custom scripts and verified that they run without error (exit code `0`) or require user input. For more information, see Create an Azure Virtual Desktop image by using VM Image Builder and PowerShell.

- Ensure that your subscription has a sufficient quota for Azure Container Instances.

  Each image build might deploy up to one temporary Azure Container Instances resource (of four standard cores) in the staging resource group. These resources are required for Isolated Image Builds.

> **Note**
>
> CIS Hardened Images (Linux or Windows) on Azure Marketplace, managed by the Center for Internet Security (CIS), can cause build failures with the VM Image Builder service due to their configurations. For instance:
>
> - CIS Hardened Images for Windows might disrupt WinRM connectivity, a prerequisite for VM Image Builder builds.
> - CIS Hardened Images for Linux can fail due to `chmod +x` permission issues.

Troubleshoot image template submission errors

Image template submission errors are returned at submission only. No error log exists for image template submission errors. If there's an error during submission, you can return the error by checking the status of the template. Specifically, review `ProvisioningState` and `ProvisioningErrorMessage`/`provisioningError`.

Here's the command for the Azure CLI:

Azure CLI

```
az image builder show --name $imageTemplateName  --resource-group
$imageResourceGroup
```

Here's the command for Azure PowerShell. To use Azure PowerShell, you need to install the VM Image Builder PowerShell modules.

Azure PowerShell

```
Get-AzImageBuilderTemplate -ImageTemplateName  <imageTemplateName> -
ResourceGroupName <imageTemplateResourceGroup> | Select-Object ProvisioningState,
ProvisioningErrorMessage
```

Here's the error output for version 2020-02-14 and earlier:

Output

```
{
  "code": "ValidationFailed",
  "message": "Validation failed: 'ImageTemplate.properties.source': Field
'imageId' has a bad value:
'/subscriptions/subscriptionID/resourceGroups/resourceGroupName/providers/Microsof
t.Compute/images/imageName'. Please review  http://aka.ms/azvmimagebuildertmplref
for details on fields requirements in the Image Builder Template."
}
```

Here's the error output for version 2021-10-01 and later:

Output

```
{
  "error": {
    "code": "ValidationFailed",
    "message": "Validation failed: 'ImageTemplate.properties.source': Field
'imageId' has a bad value:
'/subscriptions/subscriptionID/resourceGroups/resourceGroupName/providers/Microsof
t.Compute/images/imageName'. Please review  http://aka.ms/azvmimagebuildertmplref
for details on fields requirements in the Image Builder Template."
  }
}
```

> **ⓘ Important**
>
> API version 2021-10-01 introduces a change to the error schema that will be part of every future API release. If you have any Azure VM Image Builder automations, be aware of the new error output when you switch to API version 2021-10-01 or later.
>
> We recommend, after you switch to the latest API version, that you don't revert to an earlier version. If you revert, you'll have to change your automation again to produce the earlier error schema. We don't expect the error schema to change again in future releases.

The following sections present problem resolution guidance for common image template submission errors.

Update or upgrade of image templates is currently not supported

Error

Output

```
'Conflict'. Details: Update/Upgrade of image templates is currently not supported
```

Cause

The template already exists.

Solution

If you submit an image configuration template and the submission fails, a failed template artifact still exists. Delete the failed template.

Assigned managed identity on an image template can't be used

Error

Output

```
The assigned managed identity cannot be used. Please remove the existing one and
re-assign a new identity. For more troubleshooting steps go to
https://aka.ms/azvmimagebuilderts.
```

Cause

There are cases where a created managed identity assigned to the image template can't be used.

One possible cause is that the VM Image Builder template uses a customer-provided staging resource group, and the managed identity is deleted before the image template is deleted. (This is a staging resource group scenario.)

Solution

Use the Azure CLI to reset the managed identity on the image template. Be sure to update Azure CLI to the 2.45.0 version or later.

Confirm the managed identity from the target VM Image Builder template:

Azure CLI

```
az image builder identity show -g <template resource group> -n <template name>
```

Remove the managed identity from the target VM Image Builder template:

Azure CLI

```
az image builder identity remove -g <template resource group> -n <template name> --user-assigned <identity resource id>
```

Assign a new identity to the target VM Image Builder template:

Azure CLI

```
az image builder identity assign -g <template rg> -n <template name> --user-assigned <identity resource id>
```

For more information about configuring permissions, see Configure VM Image Builder permissions by using the Azure CLI or Configure VM Image Builder permissions by using PowerShell.

Assigned managed identity isn't authorized to access a resource

Error

Output

```
Not authorized to access the resource: <resource-not-able-to-access>. Please check the user assigned identity has the correct permissions. For more details, go to https://aka.ms/azvmimagebuilderts.
```

Cause

The created managed identity assigned to the image template doesn't have all permissions to access the resource shared in the error message.

Solution

Confirm the managed identity from the target VM Image Builder template:

Azure CLI

```
az image builder identity show -g <template resource group> -n <template name>
```

Review the role assignments for the identity:

Azure CLI

```
az role assignment list --assignee <identity_client_id_or_principal_id>
```

Assign the required role. If necessary, create your role with the required permissions.

For more information about configuring permissions, see Configure VM Image Builder permissions by using the Azure CLI or Configure VM Image Builder permissions by using PowerShell.

Resource operation finished with a terminal provisioning state of "Failed"

Error

Output

```
Microsoft.VirtualMachineImages/imageTemplates 'helloImageTemplateforSIG01' failed
with message '{
  "status": "Failed",
  "error": {
    "code": "ResourceDeploymentFailure",
    "message": "The resource operation completed with terminal provisioning state
'Failed'.",
    "details": [
      {
        "code": "InternalOperationError",
        "message": "Internal error occurred."
```

Cause

In most cases, the error about resource deployment failure occurs because of missing permissions. A conflict with the staging resource group might also cause this error.

Solution

Depending on your scenario, VM Image Builder might need permissions to:

- The source image or Azure Compute Gallery (formerly Shared Image Gallery) resource group.
- The distribution image or Azure Compute Gallery resource.
- The storage account, container, or blob that the `File` customizer is accessing.

Also, ensure that the staging resource group name is uniquely specified for each image template.

For more information about configuring permissions, see Configure VM Image Builder permissions by using the Azure CLI or Configure VM Image Builder permissions by using PowerShell.

Error occurs with getting a managed image

Error

Output

```
Build (Managed Image) step failed: Error getting Managed Image
'/subscriptions/.../providers/Microsoft.Compute/images/mymanagedmg1': Error
getting managed image (...): compute.
ImagesClient#Get: Failure responding to request: StatusCode=403 -- Original Error:
autorest/azure: Service returned an error.
Status=403 Code="AuthorizationFailed" Message="The client '......' with object id
'......' doesn't have authorization to perform action
'Microsoft.Compute/images/read' over scope
```

Cause

Permissions are missing.

Solution

Depending on your scenario, VM Image Builder might need permissions to:

- The source image or Azure Compute Gallery resource group.
- The distribution image or Azure Compute Gallery resource.
- The storage account, container, or blob that the `File` customizer is accessing.

For more information about configuring permissions, see Configure VM Image Builder permissions by using the Azure CLI or Configure VM Image Builder permissions by using PowerShell.

Build step failed for the image version

Error

Output

```
Build (Shared Image Version) step failed for Image Version
'/subscriptions/.../providers/Microsoft.Compute/galleries/.../images/...
/versions/0.23768.4001': Error getting Image Version
'/subscriptions/.../resourceGroups/<rgName>/providers/Microsoft.Compute/galleries/
.../images/.../versions/0.23768.4001': Error getting image version '...
:0.23768.4001': compute.GalleryImageVersionsClient#Get: Failure responding to
request: StatusCode=404 -- Original Error: autorest/azure: Service returned an
error.
Status=404 Code="ResourceNotFound" Message="The Resource
'Microsoft.Compute/galleries/.../images/.../versions/0.23768.4001' under resource
group '<rgName>' was not found."
```

Cause

VM Image Builder can't locate the source image.

Solution

Ensure that the source image is correct and exists in the location of VM Image Builder.

Error occurs with downloading an external file to a local file

Error

Output

```
Downloading external file (<myFile>) to local file (xxxxx.0.customizer.fp)
[attempt 1 of 10] failed: Error downloading '<myFile>' to
'xxxxx.0.customizer.fp'..
```

Cause

The file name or location is incorrect, or the location isn't reachable.

Solution

Ensure that the file is reachable. Verify that the name and location are correct.

Certain file repositories might use unsupported cipher suites and cause download errors with VM Image Builder. Store files and scripts in an Azure storage account to help ensure secure cipher suites and accessibility by VM Image Builder. For more information on how to store your files in Azure storage accounts, see Storage account overview.

Authorization error occurs with creating a disk

The VM Image Builder build fails with an authorization error that looks like the following example.

Error

Output

```
Attempting to deploy created Image template in Azure fails with an 'The client
'6df325020-fe22-4e39-bd69-10873965ac04' with object id '6df325020-fe22-4e39-bd69-
10873965ac04' does not have authorization to perform action
'Microsoft.Compute/disks/write' over scope
'/subscriptions/<subscriptionID>/resourceGroups/<resourceGroupName>/providers/Micr
osoft.Compute/disks/proxyVmDiskWin_<timestamp>' or the scope is invalid. If access
was recently granted, please refresh your credentials.'
```

Cause

This error happens when you try to specify a preexisting resource group and virtual network to the VM Image Builder service with a Windows source image.

Solution

You need to assign the Contributor role to the resource group for the service principal that corresponds to VM Image Builder first-party app. Use the following Azure CLI commands or Azure portal instructions.

First, validate that the service principal is associated with the VM Image Builder first-party app by using the following Azure CLI command:

Azure CLI

```
az ad sp show --id {servicePrincipalName, or objectId}
```

Then, to implement this solution by using the Azure CLI, use the following command:

Azure CLI

```
az role assignment create -g {ResourceGroupName} --assignee {AibrpSpOid} --role
Contributor
```

To implement this solution in the portal, follow the instructions in Assign Azure roles using the Azure portal:

- For Step 1: Identify the needed scope, the needed scope is your resource group.
- For Step 3: Select the appropriate role, the role is Contributor.
- For Step 4: Select who needs access, select the **Azure Virtual Machine Image Builder** member.
- In Step 7: Assign role, you assign the role.

Troubleshoot build failures

For image build failures, get the error from `lastRunStatus`, and then review the details in the `customization.log` file.

Azure CLI

```
az image builder show --name $imageTemplateName  --resource-group
$imageResourceGroup
```

Azure PowerShell

```
Get-AzImageBuilderTemplate -ImageTemplateName  <imageTemplateName> -
ResourceGroupName <imageTemplateResourceGroup> | Select-Object LastRunStatus,
LastRunStatusMessage
```

Customization log

Access live logs during image build

To effectively monitor the progress of your image build, you can access the live logs that VM Image Builder generates in Azure Container Instances. These logs provide real-time insights into the build process. They help you identify any problems or confirm that the build is proceeding as expected.

To locate and view the live logs:

1. Start the image build process.
2. Go to the Azure portal and select **Resource Groups**. Filter by the subscription where you initiated the image build.
3. Find and select the staging resource group that's associated with the image build. This resource group contains the build resources for the VM Image Builder service. For more information on the staging resource group, see Properties: stagingResourceGroup.
4. Within this resource group, look for the resource named `vmimagebuilder-build-container-**********`. If it's not visible, wait a few minutes and refresh the page.
5. On the left pane, under **Settings**, select **Containers**.
6. Go to the **Logs** tab to view the live logs during the image build process.

If you don't see any logs, try refreshing the container after a few minutes.

Download the customization and/or validation log after image build

After the image build finishes, the customization and validation logs are stored in a container within the storage account in the staging resource group that the VM Image Builder service created. For more information on the staging resource group, see Properties: stagingResourceGroup.

To locate and download the `customization.log` or `validation.log` file:

1. In the Azure portal, go to the relevant storage account by filtering for storage accounts within the staging resource group that the VM Image Builder service created.
2. Under the storage account, go to **Data Storage**.
3. Select the **Container** option, and then choose the `packerlogs` container.
4. Within the `packerlogs` container, multiple folders appear if the image build ran multiple times. These folders are arranged from the oldest build to the most recent. Select the folder that corresponds to the build that you're interested in.
5. Inside the selected folder, select the `customization.log` and/or `validation.log` file, and then select **Download** to download its contents.

Stages of the customization log

The log is verbose. It covers the image build, including any problems with the image distribution, such as Azure Compute Gallery replication. These problems are surfaced in the error message of the image template status.

The `customization.log` file includes the following stages:

1. *Deploy the build VM and dependencies by using Azure Resource Manager templates to the `IT_` staging resource group*. This stage includes multiple `POST` requests to the VM Image Builder resource provider:

   Output

   ```
   Azure request method="POST"
   request="https://management.azure.com/subscriptions/<subID>/resourceGroups/IT
   _aibImageRG200_window2019VnetTemplate01_dec33089-1cc3-cccc-cccc-
   ccccccc/providers/Microsoft.Storage/storageAccounts
   ..
   PACKER OUT ==> azure-arm: Deploying deployment template ...
   ..
   ```

2. *Get the status of the deployments*. This stage includes the status of each resource deployment:

   Output

   ```
   PACKER ERR 2020/04/30 23:28:50 packer: 2020/04/30 23:28:50 Azure request
   method="GET"
   request="https://management.azure.com/subscriptions/<subID>/resourcegroups/IT
   _aibImageRG200_window2019VnetTemplate01_dec33089-1cc3-4505-ae28-
   6661e43fac48/providers/Microsoft.Resources/deployments/pkrdp51lc0339jg/operat
   ionStatuses/08586133176207523519?[REDACTED]" body=""
   ```

3. *Connect to the build VM*.

   In Windows, VM Image Builder connects by using WinRM:

   Output

   ```
   PACKER ERR 2020/04/30 23:30:50 packer: 2020/04/30 23:30:50 Waiting for WinRM,
   up to timeout: 10m0s
   ..
   PACKER OUT     azure-arm: WinRM connected.
   ```

   In Linux, VM Image Builder connects by using SSH:

   Output

```
PACKER OUT ==> azure-arm: Waiting for SSH to become available...
PACKER ERR 2019/12/10 17:20:51 packer: 2020/04/10 17:20:51 [INFO] Waiting for
SSH, up to timeout: 20m0s
PACKER OUT ==> azure-arm: Connected to SSH!
```

4. *Run customizations*. When customizations run, you can identify them by reviewing the `customization.log` file. Search for `(telemetry)`:

```text
(telemetry) Starting provisioner windows-update
(telemetry) ending windows-update
(telemetry) Starting provisioner powershell
(telemetry) ending powershell
(telemetry) Starting provisioner file
(telemetry) ending file
(telemetry) Starting provisioner windows-restart
(telemetry) ending windows-restart

(telemetry) Finalizing. - This means the build has finished
```

5. *Deprovision*. VM Image Builder adds a hidden customizer. This step is responsible for preparing the VM for deprovisioning. In Windows, it runs `Sysprep` (by using `c:\DeprovisioningScript.ps1`). In Linux, it runs `waagent-deprovision` (by using `/tmp/DeprovisioningScript.sh`).

   For example:

```text
PACKER ERR 2020/03/04 23:05:04 [INFO] (telemetry) Starting provisioner
powershell
PACKER ERR 2020/03/04 23:05:04 packer: 2020/03/04 23:05:04 Found command: if(
TEST-PATH c:\DeprovisioningScript.ps1 ){cat c:\DeprovisioningScript.ps1} else
{echo "Deprovisioning script [c:\DeprovisioningScript.ps1] could not be
found. Image build may fail or the VM created from the Image may not boot.
Please make sure the deprovisioning script is not accidentally deleted by a
Customizer in the Template."}
```

6. *Clean up*. After the build finishes, the VM Image Builder resources are deleted:

```text
PACKER ERR ==> azure-arm: Deleting individual resources ...
...
PACKER ERR 2020/02/04 02:04:23 packer: 2020/02/04 02:04:23 Azure request
method="DELETE"
```

```
request="https://management.azure.com/subscriptions/<subId>/resourceGroups/IT
_aibDevOpsImg_t_vvvvvvv_yyyyyy-de5f-4f7c-92f2-
xxxxxxxx/providers/Microsoft.Network/networkInterfaces/pkrnijamvpo08eo?
[REDACTED]" body=""
...
PACKER ERR ==> azure-arm: The resource group was not created by Packer, not
deleting ...
```

Troubleshoot script or inline customization

The following tips can help you troubleshoot script or inline customization:

- Test the code before you supply it to VM Image Builder.
- Ensure that Azure Policy and Azure Firewall allow connectivity to remote resources.
- Send output comments to the console by using `Write-Host` or `echo`. Doing so lets you search the `customization.log` file.

Troubleshoot common build errors

Template deployment failed because of a policy violation

Error

text

```
{
  "statusCode": "BadRequest",
  "serviceRequestId": null,
  "statusMessage": "{\"error\":
{\"code\":\"InvalidTemplateDeployment\",\"message\":\"The template deployment
failed because of policy violation. Please see details for more
information.\",\"details\":[{\"code\":\"RequestDisallowedByPolicy\",\"target\":\"
<target_name>\",\"message\":\"Resource '<resource_name>' was disallowed by policy.
Policy identifiers: '[{\\\"policyAssignment\\\":{\\\"name\\\":\\\"[Initiative]
KeyVault
(Microsoft.KeyVault)\\\",\\\"id\\\":\\\"/providers/Microsoft.Management/management
Groups/<managementGroup_name>/providers/Microsoft.Authorization/policyAssignments/
Microsoft.KeyVault\\\"},\\\"policyDefinition\\\":{\\\"name\\\":\\\"Azure Key Vault
should disable public network
access\\\",\\\"id\\\":\\\"/providers/Microsoft.Management/managementGroups/<manage
mentGroup_name>/providers/Microsoft.Authorization/policyDefinitions/KeyVault.disab
lePublicNetworkAccess_deny_deny\\\"},\\\"policySetDefinition\\\":
{\\\"name\\\":\\\"[Initiative] KeyVault
(Microsoft.KeyVault)\\\",\\\"id\\\":\\\"/providers/Microsoft.Management/management
Groups/<managementGroup_name>/providers/Microsoft.Authorization/policySetDefinitio
ns/Microsoft.KeyVault\\\"}}]'.\",\"additionalInfo\":
```

```
[{\"type\":\"PolicyViolation\"}]}]}}",
  "eventCategory": "Administrative",
  "entity":
"/subscriptions/<subscription_ID>/<resourcegroups>/<resourcegroupname>/providers/M
icrosoft.Resources/deployments/<deployment_name>",
  "message": "Microsoft.Resources/deployments/validate/action",
  "hierarchy": "
<subscription_ID>/<resourcegroupname>/<policy_name>/<managementGroup_name>/<deploy
ment_ID>"
}
```

Cause

The preceding policy violation error is a result of using an Azure key vault with public access disabled. At this time, VM Image Builder doesn't support this configuration.

Solution

You must create the key vault with public access enabled.

Packer build command fails

Error

text

```
  "provisioningState": "Succeeded",
  "lastRunStatus": {
   "startTime": "2020-04-30T23:24:06.756985789Z",
   "endTime": "2020-04-30T23:39:14.268729811Z",
   "runState": "Failed",
   "message": "Failed while waiting for packerizer: Microservice has failed:
Failed while processing request: Error when executing packerizer: Packer build
command has failed: exit status 1. During the image build, a failure has occurred,
please review the build log to identify which build/customization step failed. For
more troubleshooting steps go to https://aka.ms/azvmimagebuilderts. Image Build
log location: https://xxxxxxxxxx.blob.core.windows.net/packerlogs/xxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxx/customization.log. OperationId: xxxxxx-5a8c-4379-xxxx-
8d85493bc791. Use this operationId to search packer logs."
```

Cause

The failure of a packer build command is a customization failure.

Solution

Review the log to locate customizer failures. Search for `(telemetry)`.

For example:

```
text

(telemetry) Starting provisioner windows-update
(telemetry) ending windows-update
(telemetry) Starting provisioner powershell
(telemetry) ending powershell
(telemetry) Starting provisioner file
(telemetry) ending file
(telemetry) Starting provisioner windows-restart
(telemetry) ending windows-restart

(telemetry) Finalizing. - This means the build has finished
```

Deployment fails due to timeout

Error

```
text

Deployment failed. Correlation ID: xxxxx-xxxx-xxxx-xxxx-xxxxxxxxx. Failed in
building/customizing image: Failed while waiting for packerizer: Timeout waiting
for microservice to complete: 'context deadline exceeded'
```

Cause

The build exceeded the timeout. This error appears in `lastRunStatus`.

Solution

1. Review the `customization.log` file. Identify the last customizer to run. Search for `(telemetry)`, starting from the bottom of the log.

2. Check script customizations. The customizations might not be suppressing user interaction for commands, such as `quiet` options. For example, `apt-get install -y` results in the script execution waiting for user interaction.

3. If you're using the `File` customizer to download artifacts greater than 20 MB, see the workarounds section.

4. Review errors and dependencies in the script that might cause the script to wait.

5. If you expect that the customizations need more time, increase the value of buildTimeoutInMinutes. The default is 4 hours.

6. If you have resource-intensive actions, such as downloading gigabytes of files, consider the size of the underlying build VM.

   The service uses a Standard_D1_v2 VM. The VM has one vCPU and 3.5 GB of memory. If you're downloading 50 GB, you'll likely exhaust the VM resources and cause communication failures between VM Image Builder and the build VM. Retry the build with a larger-memory VM by setting VM_size.

File download time is long

Error

text

```
[086cf9c4-0457-4e8f-bfd4-908cfe3fe43c] PACKER OUT
myBigFile.zip 826 B / 826000 B  1.00%
[086cf9c4-0457-4e8f-bfd4-908cfe3fe43c] PACKER OUT
myBigFile.zip 1652 B / 826000 B  2.00%
[086cf9c4-0457-4e8f-bfd4-908cfe3fe43c] PACKER OUT
..
hours later...
..
myBigFile.zip 826000 B / 826000 B  100.00%
[086cf9c4-0457-4e8f-bfd4-908cfe3fe43c] PACKER OUT
```

Cause

The `File` customizer is downloading a large file.

Solution

The `File` customizer is suitable only for small (less than 20 MB) file downloads. For larger file downloads, use a script or inline command. For example, in Linux, you can use `wget` or `curl`. In Windows, you can use `Invoke-WebRequest`.

Windows restart continually fails with error code 1190

Error

Output

```
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO]
(telemetry) Starting provisioner windows-restart
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
plugin-azure plugin: 2023/06/13 08:28:58 [INFO] starting remote command: shutdown
/r /f /t 10
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
plugin-azure plugin: 2023/06/13 08:28:58 [INFO] command 'shutdown /r /f /t 10'
exited with code: 0
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER OUT ==> azure-arm: A system shutdown
has already been scheduled.(1190)
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
plugin-azure plugin: 2023/06/13 08:28:58 [INFO] RPC endpoint: Communicator ended
with: 0
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] 0
bytes written for 'stdout'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] 0
bytes written for 'stderr'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] RPC
client: Communicator ended with: 0
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] RPC
endpoint: Communicator ended with: 0
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: [INFO] 0 bytes written for 'stdout'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: [INFO] 0 bytes written for 'stderr'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: [INFO] RPC client: Communicator ended with: 0
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: Check if machine is rebooting...
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
plugin-azure plugin: 2023/06/13 08:28:58 [INFO] starting remote command: shutdown
/r /f /t 60 /c "packer restart test"
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
plugin-azure plugin: 2023/06/13 08:28:58 [INFO] command 'shutdown /r /f /t 60 /c
"packer restart test"' exited with code: 1190
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
plugin-azure plugin: 2023/06/13 08:28:58 [INFO] RPC endpoint: Communicator ended
with: 1190
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] 52
bytes written for 'stderr'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] 0
bytes written for 'stdout'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] RPC
client: Communicator ended with: 1190
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 [INFO] RPC
endpoint: Communicator ended with: 1190
```

```
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: [INFO] 52 bytes written for 'stderr'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: [INFO] 0 bytes written for 'stdout'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: [INFO] RPC client: Communicator ended with:
1190
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:28:58 packer-
provisioner-windows-restart plugin: Reboot already in progress, waiting...
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:29:08 packer-
provisioner-windows-restart plugin: Check if machine is rebooting...
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:29:09 [INFO] 0
bytes written for 'stderr'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:29:09 packer-
provisioner-windows-restart plugin: [INFO] 0 bytes written for 'stderr'
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:29:09 packer-
provisioner-windows-restart plugin: Waiting for machine to reboot with timeout:
15m0s
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:29:09 packer-
provisioner-windows-restart plugin: Waiting for machine to become available...
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER OUT ==> Some builds didn't complete
successfully and had errors:
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:46:26 machine
readable: azure-arm,error []string{"Timeout waiting for machine to restart."}
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER OUT --> azure-arm: Timeout waiting
for machine to restart.
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR ==> Builds finished but no
artifacts were created.
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER OUT
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER ERR 2023/06/13 08:46:26 [INFO]
(telemetry) Finalizing.
[864c0337-b300-48ab-8e8e-7894bc695b7c] PACKER OUT ==> Builds finished but no
artifacts were created.
```

Cause

The Windows Update step is declared prematurely in images based on Windows Server 2016.

Solution

Increase `restartTimeout` from 15 minutes to 30 minutes.

Error occurs with waiting for Azure Compute Gallery

Error

```
text
```

```
Deployment failed. Correlation ID: XXXXXX-XXXX-XXXXXX-XXXX-XXXXXX. Failed in
distributing 1 images out of total 1: {[Error 0] [Distribute 0] Error publishing
MDI to Azure Compute
Gallery:/subscriptions/<subId>/resourceGroups/xxxxxx/providers/Microsoft.Compute/g
alleries/xxxxx/images/xxxxxx, Location:eastus. Error: Error returned from SIG
client while publishing MDI to Azure Compute Gallery for dstImageLocation: eastus,
dstSubscription: <subId>, dstResourceGroupName: XXXXXX, dstGalleryName: XXXXXX,
dstGalleryImageName: XXXXXX. Error: Error waiting on Azure Compute Gallery future
for resource group: XXXXXX, gallery name: XXXXXX, gallery image name:
XXXXXX.Error: Future#WaitForCompletion: context has been cancelled: StatusCode=200
-- Original Error: context deadline exceeded}
```

Cause

VM Image Builder timed out while waiting for the image to be added and replicated to Azure Compute Gallery.

If the image is being injected into the gallery, you can assume that the image build was successful. However, the overall process failed because VM Image Builder was waiting for Azure Compute Gallery to complete the replication.

Even though the build failed, the replication continues. You can get the properties of the image version by checking the distribution `runOutput`:

Azure CLI

```
$runOutputName=<distributionRunOutput>
az resource show \
    --ids
"/subscriptions/$subscriptionID/resourcegroups/$imageResourceGroup/providers/Micro
soft.VirtualMachineImages/imageTemplates/$imageTemplateName/runOutputs/$runOutputN
ame"  \
    --api-version=2020-02-14
```

Solution

Increase the value of `buildTimeoutInMinutes`.

Information events show low Windows resources

Error

text

```
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT     azure-arm: Waiting for
operation to complete (system performance: 1% cpu; 37% memory)...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT     azure-arm: Waiting for
operation to complete (system performance: 51% cpu; 35% memory)...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT     azure-arm: Waiting for
operation to complete (system performance: 21% cpu; 37% memory)...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT     azure-arm: Waiting for
operation to complete (system performance: 21% cpu; 36% memory)...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT     azure-arm: Waiting for
operation to complete (system performance: 90% cpu; 32% memory)...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT     azure-arm: Waiting for the
Windows Modules Installer to exit...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer:
2020/04/30 23:38:58 [INFO] command 'PowerShell -ExecutionPolicy Bypass -
OutputFormat Text -File C:/Windows/Temp/packer-windows-update-elevated.ps1' exited
with code: 101
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT ==> azure-arm: Restarting the
machine...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer:
2020/04/30 23:38:58 [INFO] RPC endpoint: Communicator ended with: 101
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] 1672
bytes written for 'stdout'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] 0
bytes written for 'stderr'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] RPC
client: Communicator ended with: 101
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] RPC
endpoint: Communicator ended with: 101
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT ==> azure-arm: Waiting for
machine to become available...
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer-
provisioner-windows-update: 2020/04/30 23:38:58 [INFO] 1672 bytes written for
'stdout'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer-
provisioner-windows-update: 2020/04/30 23:38:58 [INFO] 0 bytes written for
'stderr'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer-
provisioner-windows-update: 2020/04/30 23:38:58 [INFO] RPC client: Communicator
ended with: 101
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer:
2020/04/30 23:38:58 [INFO] starting remote command: shutdown.exe -f -r -t 0 -c
"packer restart"
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer:
2020/04/30 23:38:58 [INFO] command 'shutdown.exe -f -r -t 0 -c "packer restart"'
exited with code: 0
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer:
2020/04/30 23:38:58 [INFO] RPC endpoint: Communicator ended with: 0
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] 0
bytes written for 'stderr'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] 0
bytes written for 'stdout'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT ==> azure-arm: A system shutdown
is in progress.(1115)
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] RPC
```

```
client: Communicator ended with: 0
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 [INFO] RPC
endpoint: Communicator ended with: 0
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer-
provisioner-windows-update: 2020/04/30 23:38:58 [INFO] 0 bytes written for
'stdout'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer-
provisioner-windows-update: 2020/04/30 23:38:58 [INFO] 0 bytes written for
'stderr'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:58 packer-
provisioner-windows-update: 2020/04/30 23:38:58 [INFO] RPC client: Communicator
ended with: 0
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer:
2020/04/30 23:38:59 [INFO] starting remote command: shutdown.exe -f -r -t 60 -c
"packer restart test"
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer:
2020/04/30 23:38:59 [INFO] command 'shutdown.exe -f -r -t 60 -c "packer restart
test"' exited with code: 1115
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer:
2020/04/30 23:38:59 [INFO] RPC endpoint: Communicator ended with: 1115
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 [INFO] 0
bytes written for 'stdout'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 [INFO] 40
bytes written for 'stderr'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 [INFO] RPC
client: Communicator ended with: 1115
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 [INFO] RPC
endpoint: Communicator ended with: 1115
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer-
provisioner-windows-update: 2020/04/30 23:38:59 [INFO] 40 bytes written for
'stderr'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer-
provisioner-windows-update: 2020/04/30 23:38:59 [INFO] 0 bytes written for
'stdout'
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer-
provisioner-windows-update: 2020/04/30 23:38:59 [INFO] RPC client: Communicator
ended with: 1115
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER ERR 2020/04/30 23:38:59 packer-
provisioner-windows-update: 2020/04/30 23:38:59 Retryable error: Machine not yet
available (exit status 1115)
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT Build 'azure-arm' errored:
unexpected EOF
[45f485cf-5a8c-4379-9937-8d85493bc791] PACKER OUT
```

Cause

The cause is resource exhaustion. This problem commonly happens when Windows Update runs with the default build VM size, D1_V2.

Solution

Increase the build VM size.

Build finished but no artifacts were created

Warning

text

```
[<log_id>] PACKER 2023/09/14 19:01:18 ui: Build 'azure-arm' finished after 3
minutes 13 seconds.
[<log_id>] PACKER 2023/09/14 19:01:18 ui:
[<log_id>] PACKER ==> Wait completed after 3 minutes 13 seconds
[<log_id>] PACKER 2023/09/14 19:01:18 ui:
[<log_id>] PACKER ==> Builds finished but no artifacts were created.
[<log_id>] PACKER 2023/09/14 19:01:18 [INFO] (telemetry) Finalizing.
[<log_id>] PACKER 2023/09/14 19:01:19 waiting for all plugin processes to
complete...
[<log_id>] PACKER 2023/09/14 19:01:19 /aib/packerInput/packer-plugin-azure: plugin
process exited
[<log_id>] PACKER 2023/09/14 19:01:19 /aib/packerInput/packer: plugin process
exited
[<log_id>] PACKER 2023/09/14 19:01:19 /aib/packerInput/packer: plugin process
exited
[<log_id>] PACKER 2023/09/14 19:01:19 /aib/packerInput/packer: plugin process
exited
[<log_id>] PACKER Done exporting Packer logs to Azure Storage.
```

Solution

You can safely ignore the preceding warning.

Image creation was skipped

Warning

text

```
[<log_id>] PACKER 2023/09/14 19:00:18 ui: ==> azure-arm:  -> Snapshot ID :
'/subscriptions/<subscription_id>/resourceGroups/<resourcegroup_name>/providers/Mi
crosoft.Compute/snapshots/<snapshot_name>'
[<log_id>] PACKER 2023/09/14 19:00:18 ui: ==> azure-arm: Skipping image
creation...
[<log_id>] PACKER 2023/09/14 19:00:18 ui: ==> azure-arm:
[<log_id>] PACKER ==> azure-arm: Deleting individual resources ...
[<log_id>] PACKER 2023/09/14 19:00:18 packer-plugin-azure plugin: 202
```

Solution

You can safely ignore the preceding warning.

Resource is not found

Error

Output

```
  "provisioningState": "Succeeded",
  "lastRunStatus": {
   "startTime": "2020-05-01T00:13:52.599326198Z",
   "endTime": "2020-05-01T00:15:13.62366898Z",
   "runState": "Failed",
   "message": "network.InterfacesClient#UpdateTags: Failure responding to request:
StatusCode=404 -- Original Error: autorest/azure: Service returned an error.
Status=404 Code=\"ResourceNotFound\" Message=\"The Resource
'Microsoft.Network/networkInterfaces/aibpls7lz2e.nic.4609d697-be0a-4cb0-86af-
49b6fe877fe1' under resource group
'IT_aibImageRG200_window2019VnetTemplate01_9988723b-af56-413a-9006-84130af0e9df'
was not found.\""
  },
```

Cause

Permissions are missing.

Solution

Recheck that VM Image Builder has all the permissions that it requires.

For more information about configuring permissions, see Configure VM Image Builder permissions by using the Azure CLI or Configure VM Image Builder permissions by using PowerShell.

There's a Sysprep timing problem

Error

text

```
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: Write-Output '>>>
Waiting for GA Service (RdAgent) to start ...'
```

```
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: while ((Get-
Service RdAgent) -and ((Get-Service RdAgent).Status -ne 'Running')) { Start-Sleep
-s 5 }
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: Write-Output '>>>
Waiting for GA Service (WindowsAzureTelemetryService) to start ...'
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: while ((Get-
Service WindowsAzureTelemetryService) -and ((Get-Service
WindowsAzureTelemetryService).Status -ne 'Running')) { Start-Sleep -s 5 }
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: Write-Output '>>>
Waiting for GA Service (WindowsAzureGuestAgent) to start ...'
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: while ((Get-
Service WindowsAzureGuestAgent) -and ((Get-Service WindowsAzureGuestAgent).Status
-ne 'Running')) { Start-Sleep -s 5 }
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: Write-Output '>>>
Sysprepping VM ...'
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: if( Test-Path
$Env:SystemRoot\system32\Sysprep\unattend.xml ) {
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:   Remove-Item
$Env:SystemRoot\system32\Sysprep\unattend.xml -Force
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: }
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: &
$Env:SystemRoot\System32\Sysprep\Sysprep.exe /oobe /generalize /quiet /quit
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: while($true) {
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:   $imageState =
(Get-ItemProperty
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup\State).ImageState
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:   Write-Output
$imageState
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:   if ($imageState
-eq 'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { break }
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:   Start-Sleep -s
5
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: }
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: Write-Output '>>>
Sysprep complete ...'
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: >>> Waiting for
GA Service (RdAgent) to start ...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: >>> Waiting for
GA Service (WindowsAzureTelemetryService) to start ...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: >>> Waiting for
GA Service (WindowsAzureGuestAgent) to start ...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm: >>> Sysprepping
VM ...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_COMPLETE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm: Get-Service :
Cannot find any service with service name 'WindowsAzureGuestAgent'.
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm: At
C:\DeprovisioningScript.ps1:6 char:9
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm: + while ((Get-
Service WindowsAzureGuestAgent) -and ((Get-Service Window ...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm: +
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm:     +
CategoryInfo          : ObjectNotFound: (WindowsAzureGuestAgent:String) [Get-
```

```
Service], ServiceCommandException
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm:     +
FullyQualifiedErrorId :
NoServiceFoundForGivenName,Microsoft.PowerShell.Commands.GetServiceCommand
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm:
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT     azure-arm:
IMAGE_STATE_UNDEPLOYABLE
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER ERR 2020/05/05 22:26:17 Cancelling
builder after context cancellation context canceled
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT Cancelling build after receiving
terminated
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER ERR 2020/05/05 22:26:17 packer:
2020/05/05 22:26:17 Cancelling provisioning due to context cancellation: context
canceled
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm:
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER ERR 2020/05/05 22:26:17 packer:
2020/05/05 22:26:17 Cancelling hook after context cancellation context canceled
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER OUT ==> azure-arm: The resource
group was not created by Packer, deleting individual resources ...
[922bdf36-b53c-4e78-9cd8-6b70b9674685] PACKER ERR ==> azure-arm: The resource
group was not created by Packer, deleting individual resources ...
```

Cause

The D1_V2 VM size might be the reason for the timing problem. If customizations are limited and are run in less than three seconds, VM Image Builder runs `Sysprep` commands to deprovision. When VM Image Builder deprovisions, the `Sysprep` command checks for

`WindowsAzureGuestAgent`, which might not be fully installed and might cause the timing problem.

Solution

To avoid the timing problem, you can increase the VM size or add a 60-second PowerShell sleep customization.

Azure Container Instances provider is unregistered

Error

```text
Azure Container Instances provider not registered for your subscription.
```

Cause

Your template subscription doesn't have the Azure Container Instances provider registered.

Solution

Register the Azure Container Instances provider for your template subscription, and add one of the following commands:

- Azure CLI: `az provider register -n Microsoft.ContainerInstance`
- Azure PowerShell: `Register-AzResourceProvider -ProviderNamespace Microsoft.ContainerInstance`

Subscription exceeded the Azure Container Instances quota

Error

```text
Azure Container Instances quota exceeded
```

Cause

Your subscription doesn't have enough Azure Container Instances quota for VM Image Builder to successfully build an image.

Solution

You can take the following actions to make Azure Container Instances quota available for VM Image Builder:

- Look up other usage of Azure Container Instances in your subscription. Remove any unneeded instances to make quota available for VM Image Builder.
- VM Image Builder deploys Azure Container Instances only temporarily during a build. These instances are deleted after the build finishes.

  If too many concurrent image builds are running in your subscription, you can consider delaying some of the image builds. This delay reduces concurrent usage of Azure Container Instances in your subscription.

  If your image templates are set up for automatic image builds via triggers, VM Image Builder automatically retries such failed builds.
- If the current Azure Container Instances limits for your subscription are too low to support your image-building scenarios, you can request an increase in your Azure Container Instances quota.

> **Note**
>
> Azure Container Instances resources are required for **Isolated Image Builds**.

Too many Azure Container Instances resources are deployed within a period of time

Error

"Too many Azure Container Instances deployed within a period of time."

Cause

Your subscription doesn't have enough Azure Container Instances quota for VM Image Builder to successfully build images concurrently.

Solution

You can try these solutions:

- Retry your failed builds in a less concurrent manner.
- If the current Azure Container Instances limits for your subscription are too low to support your image-building scenarios, you can request an increase in your Azure Container Instances quota.

Isolated Image Builds feature causes a failure

Error

VM Image Builder builds are failing due to Isolated Image Builds.

Cause

VM Image Builder builds can fail for reasons listed elsewhere in this article. However, there's a small chance that a build fails due to Isolated Image Builds, depending on your scenario, subscription quotas, or some unforeseen service error. For more information, see Isolated Image Builds.

Solution

If you determine that a build is failing due to Isolated Image Builds, ensure that:

- No Azure policy is blocking the deployment of resources mentioned in the Prerequisites section of this article (specifically, Azure Container Instances).
- Your subscription has sufficient quota for Azure Container Instances to support all your concurrent image builds. For more information, see the Subscription exceeded the Azure Container Instances quota section of this article.

VM Image Builder is currently in the process of deploying Isolated Image Builds. Specific image templates are not tied to Isolated Image Builds, and the same image template might or might not use Isolated Image Builds during different builds.

To temporarily run your build without Isolated Image Builds, retry your build. Because image templates aren't tied to the Isolated Image Builds feature, retrying a build has a high probability of rerunning without Isolated Image Builds.

If none of these solutions mitigate failing image builds, you can contact Azure support to temporarily opt your subscription out of Isolated Image Builds. For more information, see

Create an Azure support request.

> **Note**
>
> If the Isolated Image Builds feature is eventually enabled in all regions and templates, the preceding mitigations will be only temporary. It's best to address the underlying cause of build failures.

Build is canceled after context cancellation

Error

text

```
PACKER ERR 2020/03/26 22:11:23 Cancelling builder after context cancellation
context canceled
PACKER OUT Cancelling build after receiving terminated
PACKER ERR 2020/03/26 22:11:23 packer-builder-azure-arm plugin: Cancelling hook
after context cancellation context canceled
..
PACKER ERR 2020/03/26 22:11:23 packer-builder-azure-arm plugin: Cancelling
provisioning due to context cancellation: context canceled
PACKER ERR 2020/03/26 22:11:25 packer-builder-azure-arm plugin: [ERROR] Remote
command exited without exit status or exit signal.
PACKER ERR 2020/03/26 22:11:25 packer-builder-azure-arm plugin: [INFO] RPC
endpoint: Communicator ended with: 2300218
PACKER ERR 2020/03/26 22:11:25 [INFO] 148974 bytes written for 'stdout'
PACKER ERR 2020/03/26 22:11:25 [INFO] 0 bytes written for 'stderr'
PACKER ERR 2020/03/26 22:11:25 [INFO] RPC client: Communicator ended with: 2300218
PACKER ERR 2020/03/26 22:11:25 [INFO] RPC endpoint: Communicator ended with:
2300218
```

Cause

VM Image Builder uses port 22 (Linux) or 5986 (Windows) to connect to the build VM. This problem occurs when the service is disconnected from the build VM during an image build. The reasons for the disconnection can vary, but enabling or configuring a firewall in the script can block the previously mentioned ports.

Solution

Review your scripts for firewall changes or enablement, or changes to SSH or WinRM. Ensure that any changes allow for constant connectivity between the service and the build VM on the

previously mentioned ports. For more information, see VM Image Builder networking options.

JWT errors appear in the log early in the build

Error

Early in the build process, the build fails and the log indicates a JSON Web Token (JWT) error:

```text
PACKER OUT Error: Failed to prepare build: "azure-arm"
PACKER ERR
PACKER OUT
PACKER ERR * client_jwt will expire within 5 minutes, please use a JWT that is
valid for at least 5 minutes
PACKER OUT 1 error(s) occurred:
```

Cause

The `buildTimeoutInMinutes` value in the template is set to from 1 to 5 minutes.

Solution

As described in Create a VM Image Builder template, the timeout must be set to 0 minutes to use the default or set to more than 5 minutes to override the default. Change the timeout in your template to 0 minutes to use the default, or change it to a minimum of 6 minutes.

Resource deletion errors appear

Error

Intermediate resources are cleaned up toward the end of the build, and the customization log might show several resource deletion errors:

```text
PACKER OUT ==> azure-arm: Error deleting resource. Will retry.
...
PACKER OUT ==> azure-arm: Error: network.PublicIPAddressesClient#Delete: Failure
sending request: StatusCode=0 -- Original Error:
Code="PublicIPAddressCannotBeDeleted" Message=...
...
PACKER ERR 2022/03/07 18:43:06 packer-plugin-azure plugin: 2022/03/07 18:43:06
```

```
Retryable error: network.SecurityGroupsClient#Delete: Failure sending request:
StatusCode=0 -- Original Error: Code="InUseNetworkSecurityGroupCannotBeDeleted"...
```

Cause

These error log messages are mostly harmless, because resource deletions are retried several times. Ordinarily, they eventually succeed. You can verify this behavior by continuing to follow the deletion logs until you observe a success message. Alternatively, you can inspect the staging resource group to confirm whether the resource was deleted.

Making these observations is especially important in build failures. These error messages might lead you to conclude that they're the reason for the failures, even when the actual errors might be elsewhere.

Error

When images are stuck in template deletion, the customization log might show the following error:

Output

```
error deleting resource id
/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Networ
k/networkInterfaces/<networkInterfacName>: resources.Client#DeleteByID: Failure
sending request: StatusCode=400 --
Original Error: Code="NicInUseWithPrivateEndpoint"
Message="Network interface
/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Networ
k/networkInterfaces/<networkInterfacName> cannot be deleted because it is
currently in use with an private endpoint
(/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.Netwo
rk/privateEndpoints/<pIname>)." Details=[]
```

Cause

The error occurs because the network interface is currently in use with a private endpoint.

Solution

To resolve the problem, delete the following resources one by one in this specific order:

1. Private endpoint connection. You can find this connection in the Azure Private Link service resource by going to the **Private endpoint connections** tab on the page for the Private

Link service resource.
2. Private Link service.
3. Network interface and load balancer.
4. Resource group.
5. Image template.

For more assistance, you can contact Azure support to resolve the stuck deletion error.

Distribution target is not found in the update request

Error

text

```
Validation failed: Distribute target with Runoutput name <runoutputname> not found
in the update request. Deleting a distribution target is not allowed.
```

Cause

This error occurs when an existing distribution target isn't found in the patch request body.

Solution

The distribution array should contain all the distribution targets: new targets (if any), existing targets with no change, and updated targets. If you want to remove an existing distribution target, delete and re-create the image template. Deleting a distribution target is currently not supported through the Patch API.

Required fields are missing

Error

text

```
Validation failed: 'ImageTemplate.properties.distribute[<index>]': Missing field
<fieldname>. Please review http://aka.ms/azvmimagebuildertmplref for details on
fields required in the Image Builder Template.
```

Cause

This error occurs when a required field is missing from a distribution target.

Solution

When you're creating a request, be sure to provide every required field in a distribution target, even if there's no change.

Troubleshoot Azure DevOps

An Azure DevOps task fails only if an error occurs during customization. When this error happens, the task reports the failure and leaves the staging resource group, with the logs, so that you can identify the problem.

To locate the log:

1. Find the template name. Go to **Pipeline** > **Failed build**, and then drill down into the Azure DevOps task in VM Image Builder. Note the `template name` value.

   ```text
   start reading task parameters...
   found build at:  /home/vsts/work/r1/a/_ImageBuilding/webapp
   end reading parameters
   getting storage account details for aibstordot1556933914
   created archive /home/vsts/work/_temp/temp_web_package_21475337782320203.zip
   Source for image:  { type: 'SharedImageVersion',
     imageVersionId:
   '/subscriptions/<subscriptionID>/resourceGroups/<rgName>/providers/Microsoft.
   Compute/galleries/<galleryName>/images/<imageDefName>/versions/<imgVersionNum
   ber>' }
   template name:  t_1556938436xxx
   ```

2. Go to the Azure portal, search for the template name in the resource group, and then search for the resource group by entering **IT_**.

3. Select the storage account name, and then select **blobs** > **containers** > **logs**.

You might occasionally need to investigate successful builds and review their logs. As mentioned earlier, if the image build is successful, the staging resource group that contains the logs is deleted as part of the cleanup. To prevent an automatic cleanup, you can introduce `sleep` after the inline command, and then view the logs as the build is paused:

1. Update the inline command by adding `Write-Host / Echo "Sleep"`. This addition gives you time to search in the log.
2. Add a `sleep` value of at least 10 minutes by using a Start-Sleep or `Sleep` Linux command.

3. Use this method to identify the log location, and then keep downloading or checking the log until it gets to `sleep`.

Operation was canceled

Error

text

```
2020-05-05T18:28:24.9280196Z ##[section]Starting: Azure VM Image Builder Task
2020-05-05T18:28:24.9609966Z
==============================================================================
2020-05-05T18:28:24.9610739Z Task         : Azure VM Image Builder Test
2020-05-05T18:28:24.9611277Z Description  : Build images using Azure Image Builder
resource provider.
2020-05-05T18:28:24.9611608Z Version      : 1.0.18
2020-05-05T18:28:24.9612003Z Author       : Microsoft Corporation
2020-05-05T18:28:24.9612718Z Help         : For documentation, and end to end
example, please visit: http://aka.ms/azvmimagebuilderdevops
2020-05-05T18:28:24.9613390Z
==============================================================================
2020-05-05T18:28:26.0651512Z start reading task parameters...
2020-05-05T18:28:26.0673377Z found build at:
d:\a\r1\a\_AppsAndImageBuilder\webApp
2020-05-05T18:28:26.0708785Z end reading parameters
2020-05-05T18:28:26.0745447Z getting storage account details for
aibstagstor1565047758
2020-05-05T18:28:29.8812270Z created archive
d:\a\_temp\temp_web_package_09737279437949953.zip
2020-05-05T18:28:33.1568013Z Source for image:  { type: 'PlatformImage',
2020-05-05T18:28:33.1584131Z   publisher: 'MicrosoftWindowsServer',
2020-05-05T18:28:33.1585965Z   offer: 'WindowsServer',
2020-05-05T18:28:33.1592768Z   sku: '2016-Datacenter',
2020-05-05T18:28:33.1594191Z   version: '14393.3630.2004101604' }
2020-05-05T18:28:33.1595387Z template name:  t_1588703313152
2020-05-05T18:28:33.1597453Z starting put template...
2020-05-05T18:28:52.9278603Z put template:  Succeeded
2020-05-05T18:28:52.9281282Z starting run template...
2020-05-05T19:33:14.3923479Z ##[error]The operation was canceled.
2020-05-05T19:33:14.3939721Z ##[section]Finishing: Azure VM Image Builder Task
```

Cause

If a user didn't cancel the build, the Azure DevOps user agent canceled it. Most likely, the one-hour timeout occurred because of Azure DevOps capabilities. If you're using a private project and agent, you get 60 minutes of build time. If the build exceeds the timeout, Azure DevOps cancels the running task.

For more information about Azure DevOps capabilities and limitations, see Microsoft-hosted agents.

Solution

You can host your own Azure DevOps agents or try to reduce the time of your build. For example, if you're distributing to Azure Compute Gallery, you can replicate them to one region or replicate them asynchronously.

Windows logon is slow

Error

This error might occur when you create a Windows 10 image by using VM Image Builder, create a VM from the image, and then use Remote Desktop Protocol (RDP). You wait several minutes at the first logon screen, and then a blue screen displays the following message:

```text
Please wait for the Windows Modules Installer
```

Solution

1. In the image build, ensure that:

    - There are no outstanding restarts required by adding a Windows restart customizer as the last customization.
    - All software installation is complete.

2. Add the /mode:vm option to the default `Sysprep` command that VM Image Builder uses. For more information, see Overriding the commands later in this article.

Troubleshoot VMs created unsuccessfully from VM Image Builder

By default, VM Image Builder runs deprovisioning code at the end of each image customization phase to *generalize* the image. Generalizing an image sets it up for reuse in creating multiple VMs. As part of the process, you can pass in VM settings, such as host name and username.

In Windows, VM Image Builder runs a generic `Sysprep` command. However, this command might not be suitable for every successful Windows generalization. With VM Image Builder, you can customize the `Sysprep` command. VM Image Builder is an image automation tool that's responsible for running `Sysprep` commands successfully. But you might need different `Sysprep` commands to make your image reusable.

In Linux, VM Image Builder runs a generic `waagent -deprovision+user` command. For more information, see the Microsoft Azure Linux Agent documentation.

If you're migrating an existing customization and you're using various `Sysprep` or `waagent` commands, you can try the VM Image Builder generic commands. If the VM creation fails, use your previous `Sysprep` or `waagent` commands.

Let's suppose that you used VM Image Builder successfully to create a Windows custom image, but you failed to create a VM successfully from the image. For example, the VM creation fails to finish or it times out. In this event, do either of the following tasks:

- Review the Windows Server `Sysprep` documentation.
- Raise a support request with the Windows Server `Sysprep` customer support team. That team can help troubleshoot your problem and advise you on the correct `Sysprep` command.

Command locations and file names

In Windows:

```
c:\DeprovisioningScript.ps1
```

In Linux:

Bash

```
/tmp/DeprovisioningScript.sh
```

Sysprep command for Windows

Azure PowerShell

```
Write-Output '>>> Waiting for GA Service (RdAgent) to start ...'
while ((Get-Service RdAgent).Status -ne 'Running') { Start-Sleep -s 5 }
```

```
Write-Output '>>> Waiting for GA Service (WindowsAzureTelemetryService) to start
...'
while ((Get-Service WindowsAzureTelemetryService) -and ((Get-Service
WindowsAzureTelemetryService).Status -ne 'Running')) { Start-Sleep -s 5 }
Write-Output '>>> Waiting for GA Service (WindowsAzureGuestAgent) to start ...'
while ((Get-Service WindowsAzureGuestAgent).Status -ne 'Running') { Start-Sleep -s
5 }
Write-Output '>>> Sysprepping VM ...'
if( Test-Path $Env:SystemRoot\system32\Sysprep\unattend.xml ) {
  Remove-Item $Env:SystemRoot\system32\Sysprep\unattend.xml -Force
}
& $Env:SystemRoot\System32\Sysprep\Sysprep.exe /oobe /generalize /quiet /quit
while($true) {
  $imageState = (Get-ItemProperty
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup\State).ImageState
  Write-Output $imageState
  if ($imageState -eq 'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { break }
  Start-Sleep -s 5
}
Write-Output '>>> Sysprep complete ...'
```

Deprovision command for Linux

Bash

```
sudo /usr/sbin/waagent -force -deprovision+user && export HISTSIZE=0 && sync
```

Overriding the commands

To override the commands, use the PowerShell or shell script provisioners to create the command files with the exact file name. Put them in the previously listed directories. VM Image Builder reads these commands and writes output to the `customization.log` file.

Get support

If you used the guidance in this article and are still having problems, you can open a support case.

Use the following code to select the correct product and support topic. Doing so ensures that you're connected with the Azure VM Image Builder support team.

text

```
Product Family: Azure
Product: Virtual Machine Running (Window\Linux)
```

```
Support Topic: Azure Features
Support Subtopic: Azure Image Builder
```

Related content

- Azure VM Image Builder overview

Best practices for Azure VM Image Builder

07/30/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article describes best practices to follow while you're using Azure VM Image Builder.

Use resource locks for image templates

To prevent image templates from being accidentally deleted, use resource locks on them. For more information, see Lock your Azure resources to protect your infrastructure.

Set up image templates for disaster recovery

Make sure your image templates are set up for disaster recovery by following the reliability recommendations for VM Image Builder.

Set up triggers

Set up VM Image Builder triggers to automatically rebuild your images and keep them updated.

Enable VM boot optimization

Enable virtual machine (VM) boot optimization in VM Image Builder to improve the creation time for your VMs.

Bring your own subnets

Specify your own build VM subnets (subnetId) and Azure Container Instances subnets (containerInstanceSubnetId) for tighter control over deployment of network-related resources by VM Image Builder in your subscription. Specifying these subnets also leads to faster and more reliable image builds.

You can read more about this network topology in the Isolated Image Builds section.

Follow the principle of least privilege

Follow the principle of least privilege for your VM Image Builder resources.

Image template

A principal that has access to your image template can run, delete, or tamper with it. This access then allows the principal to change the images that image template created.

Make sure that only required principals can access image templates.

Staging resource group

VM Image Builder uses a staging resource group in your subscription to customize your VM image. You must consider this resource group as sensitive and restrict access to only required principals. Keep these risks in mind:

- Because the process of customizing your image happens in this resource group, a principal that has access to the resource group can compromise the image-building process. For example, such a principal can inject malware into the image.
- VM Image Builder delegates privileges associated with the template identity and the build VM identity to resources in this resource group. A principal that has access to the resource group can get access to these identities.
- VM Image Builder maintains a copy of your customizer artifacts in this resource group. A principal that has access to the resource group can inspect these copies.

Template identity

A principal that has access to your template identity can access all resources that the identity has permissions for. This set of resources includes your customizer artifacts (for example, shell and PowerShell scripts), your distribution targets (for example, an Azure Compute Gallery image version), and your virtual network.

You must provide only the minimum required privileges to this identity.

Build VM identity

A principal that has access to your build VM identity can access all resources that the identity has permissions for. This set of resources includes any artifacts and virtual networks that you might be using from within the build VM via this identity.

You must provide only the minimum required privileges to this identity.

Credentials

Do not put any credentials in the image template or in files used for Shell, PowerShell, and File Customizers & Validators. For example:

- While specifying inline commands for customizers and validators, do not specify any password or other login credentials. Instead, such credentials should be stored in an Azure Key Vault and then accessed from the build VM using the Build VM identity.
- While providing files in customizers and validators, do not specify any credentials in the files. Instead, such credentials should be stored in an Azure Key Vault and then accessed from the build VM using the Build VM identity.
- While specifying script or source URIs for customizers and validators in an image template, do not specify SAS URIs or URIs with credentials (like Personal Access Tokens). Instead, store such files in Azure Storage Account and use Template identity to access them.

Though Azure VM Image Builder does not prohibit specifying such credentials, such use is highly discouraged. In case such a credential is indeed specified, ensure that it does not provide access to any privileged resources and is rotated as soon as possible.

Follow Azure Compute Gallery best practices

If you're distributing to Azure Compute Gallery, be sure to also follow best practices for Azure Compute Gallery resources.

How to Connect to the Azure Image Builder Build VM

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Use this article to learn how to connect to the Azure Image Builder Build VM in order to debug problems and errors you might encounter when you're using Azure VM Image Builder.

Retain the Build VM

By default, Azure Image Builder (AIB) deletes the Build VM after the customization or validation phase, regardless of success or failure. However, in case of failures, it's essential to retain the VM for debugging purposes. Here are two options:

Recommended Option: Specify `errorHandling` Property in the Template:

If you anticipate the need for debugging, create your template with the `errorHandling` property set. This ensures that AIB retains the Build VM even if errors occur during the build process. Follow the instructions in the Properties: errorHandling - Azure Image Builder documentation.

Alternative Option (Not Recommended): Lock the Resource Group:

If you can't modify the template directly, follow these steps:

- Get the name of the staging resource group (specified in the `stagingResourceGroup` property of the template).
- Apply a `CanNotDelete` lock to the resource group using Azure Resource Manager locks.
- Once debugging is complete, remove the lock and delete the resources (except storage accounts) from the resource group.

> **Note**

AIB will attempt to delete resources in the resource group after the build, but the lock will prevent deletion. Eventually, AIB will give up, and the build will complete. Note that even if there were no errors during customization or validation, the AIB build will be reported as "Failed" due to deletion errors.

Get the credentials to connect to the VM

After retaining the Build VM, follow these steps to get the credentials needed to connect to the VM:

- Navigate to the Virtual Machine's Health > Reset Password blade.
- Click on "Reset password" mode.
- Make a note of the `Username`.
- Set the `Password` to a desired value and make a note of it. You'll use these credentials in the next step.

Connect to the VM

Once you have the credentials, you can connect to the VM using one of the options below:

Option 1 (Recommended): Connect to the VM using Azure Bastion

Azure Bastion provides a secure way to connect to VMs without exposing public IPs. Follow the instructions for connecting to a Windows VM using SSH or RDP in the Azure Bastion documentation.

> **Note**
>
> No changes to existing Network Security Groups (NSGs) are needed when connecting to the Azure Image Builder Build VM using Azure Bastion.

Option 2 (Recommended, if available): Connect to the VM using Azure Serial Console

Azure Serial Console offers text-based access to VMs running Linux or Windows. Learn more about how to use Azure Serial Console to connect to a VM by reading the Azure

Serial Console documentation.

Option 3 (Not Recommended): Use a Public IP Address Resource

The recommended practice is to use Azure Bastion or Azure Serial Console for connecting to VMs. However, if Bastion or Serial Console cannot be used, you can connect to the Build VM by using public IPs.

Use the following documentation to connect to a VM using public IPs:

- Connect to a Linux VM using Public IP
- Connect to a Windows VM using Public IP

> **Note**
>
> You may need to associate a public IP address to the build VM first and you can do that by following the instructions in the following documentation: **Associate a public IP address to a virtual machine**.
>
> You also should ensure network connectivity from the client device (running WinRM/SSH) to the Build VM over the public network. Check and update your Network Security Groups (NSGs) as needed.

Option 4 (Not Recommended): Deploying a Jump Box

This option should be your last resort. Only use this option when you cannot use Azure Bastion, Azure Serial Console or a public IP address to connect to the Build VM.

Follow the steps below to deploy a jump box:

1. Create or use a separate subnet in the same VNet as the build VM.
2. Deploy a "Jump Box" VM with a public IP in this subnet.
3. Connect from your client device to the Jump Box using any of the methods mentioned earlier.
4. From the Jump Box, connect to the build VM using SSH or WinRM/RDP.

> **Note**
>
> There must be network line-of-sight from the Client Device to the Jump Box over the public network, and from the Jump Box to the Build VM over the private network. Additionally, it's essential to verify and update Network Security Groups

(NSGs) at each hop in between and ensure that the correct ports are open on all three devices.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Reliability in Azure Image Builder (AIB)

This article contains cross-region disaster recovery and business continuity.

Azure Image Builder (AIB) is a regional service with a cluster that serves single regions. The AIB regional setup keeps data and resources within the regional boundary. AIB as a service doesn't do fail over for cluster and SQL database in region down scenarios.

> **Note**
>
> Azure Image Builder doesn't support availability zones.

Cross-region disaster recovery and business continuity

Disaster recovery (DR) refers to practices that organizations use to recover from high-impact events, such as natural disasters or failed deployments that result in downtime and data loss. Regardless of the cause, the best remedy for a disaster is a well-defined and tested DR plan and an application design that actively supports DR. Before you start creating your disaster recovery plan, see Recommendations for designing a disaster recovery strategy.

For DR, Microsoft uses the shared responsibility model. In this model, Microsoft ensures that the baseline infrastructure and platform services are available. However, many Azure services don't automatically replicate data or fall back from a failed region to cross-replicate to another enabled region. For those services, you're responsible for setting up a disaster recovery plan that works for your workload. Most services that run on Azure platform as a service (PaaS) offerings provide features and guidance to support DR. You can use service-specific features to support fast recovery to help develop your DR plan.

To ensure fast and easy recovery for Azure Image Builder (AIB), it's recommended that you run an image template in region pairs or multiple regions when designing your AIB solution. You should also replicate resources from the start when you're setting up your image templates.

Multi-region geography disaster recovery

When a regional disaster occurs, Microsoft is responsible for outage detection, notifications, and support for AIB. However, you're responsible for setting up disaster recovery for the control (service side) and data planes.

Outage detection, notification, and management

Microsoft sends a notification if there's an outage in the Azure Image Builder (AIB) Service. One common outage symptom is image templates getting 500 errors when attempting to run. You can review Azure Image Builder outage notifications and status updates through support request management.

Set up disaster recovery and outage detection

You're responsible for setting up disaster recovery for your Azure Image Builder (AIB) environment, as there isn't a region failover at the AIB service side. You need to configure both the control plane (service side) and data plane.

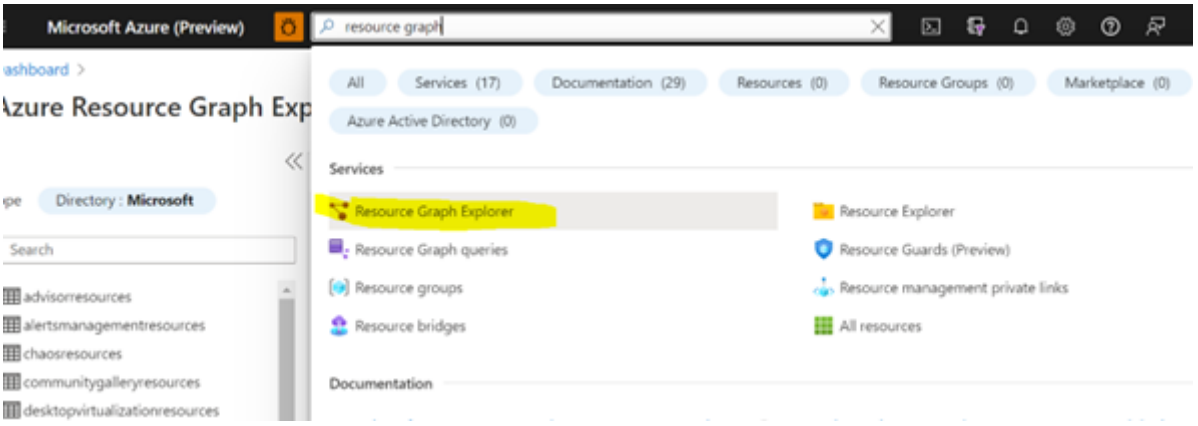
It's recommended that you create an AIB resource in another nearby region, into which you can replicate your resources. For more information, see the supported regions and what resources are included in an AIB creation.

Single-region geography disaster recovery

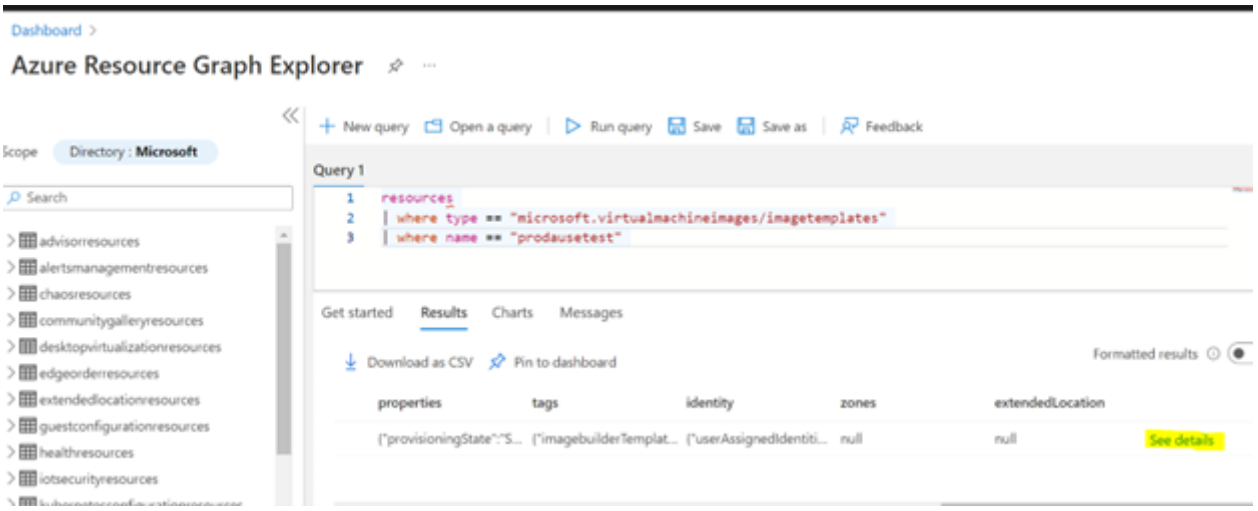
In the case of a disaster for single-region, you still need to get an image template resource from that region even when that region isn't available. You can either maintain a copy of an image template locally or can use Azure Resource Graph from the Azure portal to get an image template resource.

To get an image template resource using Resource Graph from the Azure portal:

1. Go to the search bar in Azure portal and search for *resource graph explorer*.



2. Use the search bar on the far left to search resource by type and name to see how the details give you properties of the image template. The *See details* option on the bottom right shows the image template's properties attribute and tags separately. Template name, location, ID, and tenant ID can be used to get the correct image template resource.



Capacity and proactive disaster recovery resiliency

Microsoft and its customers operate under the shared responsibility model. In customer-enabled DR (customer-responsible services), you're responsible for addressing DR for any service you deploy and control. To ensure that recovery is proactive, you should always pre-deploy secondaries. Without pre-deployed secondaries, there's no guarantee of capacity at time of impact.

When planning where to replicate a template, consider:

- AIB region availability:
  - Choose AIB supported regions close to your users.
  - AIB continually expands into new regions.
- Azure paired regions:
  - For your geographic area, choose two regions paired together.
  - Recovery efforts for paired regions where prioritization is needed.

Additional guidance

In regards to your data processing information, refer to the Azure Image Builder data residency details.

Next steps

- Reliability in Azure
- Enable Azure VM disaster recovery between availability zones
- Azure Image Builder overview

Last updated on 01/22/2026

VM optimization for gallery images with Azure VM Image Builder

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Virtual Machine Scale Sets

In this article, you learn how to use Azure VM Image Builder to optimize your ACG (Azure Compute Gallery) Images or Managed Images or VHDs to improve the create time for your VMs.

Azure VM Optimization

Azure VM optimization improves virtual machine creation time by updating the gallery image to optimize the image for a faster boot time.

Image types supported

Optimization for the following images is supported:

Expand table

Features	Details
OS Type	Linux, Windows
Partition	MBR/GPT
Hyper-V	Gen1/Gen2
OS State	Generalized
Architecture	X64, ARM64

The following types of images aren't supported:

- Images with size greater than 2 TB
- Specialized images

Optimization in Azure VM Image Builder

Optimization can be enabled while creating a VM image using the CLI.

Customers can create an Azure VM Image Builder template using CLI. It contains details regarding source, type of customization, and distribution.

In your template, you will need to enable the additional fields for VM optimization. For more information on how to enable the VM optimization fields for your image builder template, see the Optimize property.

> **Note**
>
> To enable VM optimization benefits, you must be using Azure Image Builder API Version `2022-07-01` or later.

FAQs

Can VM optimization be used without Azure VM Image Builder customization?

Yes, customers can opt for only VM optimization without using Azure VM Image Builder customization feature. Customers can simply enable the optimization flag and keep customization field as empty.

Can an existing ACG image version be optimized?

No, this optimization feature won't update an existing SIG image version. However, optimization can be enabled during new version creation for an existing image

How much time does it take for generating an optimized image?

The below latencies have been observed at various percentiles:

Expand table

OS	Size	P50	P95	Average
Linux	30 GB VHD	20 mins	21 mins	20 mins
Windows	127 GB VHD	34 mins	35 mins	33 mins

This is the end to end duration observed. Note, image generation duration varies based on different factors such as, OS Type, VHD size, OS State, etc.

Is OS image copied out of customer subscription for optimization?

Yes, the OS VHD is copied from customer subscription to Azure subscription for optimization in the same geographic location. Once optimization is finished or timed out, Azure internally deletes all copied OS VHDs.

What are the performance improvements observed for VM boot optimization?

Enabling VM boot optimization feature may not always result in noticeable performance improvement as it depends on several factors like source image already optimized, OS type, customization etc. However, to ensure the best VM boot performance, it's recommended to enable this feature.

Next steps

Learn more about Azure Compute Gallery.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to use Packer to create Linux virtual machine images in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Each virtual machine (VM) in Azure is created from an image that defines the Linux distribution and OS version. Images can include pre-installed applications and configurations. The Azure Marketplace provides many first and third-party images for most common distributions and application environments, or you can create your own custom images tailored to your needs. This article details how to use the open source tool Packer to define and build custom images in Azure.

> **Note**
>
> Azure now has a service, Azure Image Builder, for defining and creating your own custom images. Azure Image Builder is built on Packer, so you can even use your existing Packer shell provisioner scripts with it. To get started with Azure Image Builder, see Create a Linux VM with Azure Image Builder.

Create Azure resource group

During the build process, Packer creates temporary Azure resources as it builds the source VM. To capture that source VM for use as an image, you must define a resource group. The output from the Packer build process is stored in this resource group.

Create a resource group with az group create. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

Azure CLI

```
az group create -n myResourceGroup -l eastus
```

Create Azure credentials

Packer authenticates with Azure using a service principal. An Azure service principal is a security identity that you can use with apps, services, and automation tools like Packer.

You control and define the permissions as to what operations the service principal can perform in Azure.

Create a service principal with az ad sp create-for-rbac and output the credentials that Packer needs:

Azure CLI

```
az ad sp create-for-rbac --role Contributor --scopes
/subscriptions/<subscription_id> --query "{ client_id: appId, client_secret:
password, tenant_id: tenant }"
```

An example of the output from the preceding commands is as follows:

Output

```
{
    "client_id": "f5b6a5cf-fbdf-4a9f-b3b8-3c2cd00225a4",
    "client_secret": "0e760437-bf34-4aad-9f8d-870be799c55d",
    "tenant_id": "72f988bf-86f1-41af-91ab-2d7cd011db47"
}
```

To authenticate to Azure, you also need to obtain your Azure subscription ID with az account show:

Azure CLI

```
az account show --query "{ subscription_id: id }"
```

You use the output from these two commands in the next step.

Define Packer template

To build images, you create a template as a JSON file. In the template, you define builders and provisioners that carry out the actual build process. Packer has a provisioner for Azure that allows you to define Azure resources, such as the service principal credentials created in the preceding step.

Create a file named *ubuntu.json* and paste the following content. Enter your own values for the following parameters:

Expand table

Parameter	Where to obtain
client_id	First line of output from `az ad sp` create command - *appId*
client_secret	Second line of output from `az ad sp` create command - *password*
tenant_id	Third line of output from `az ad sp` create command - *tenant*
subscription_id	Output from `az account show` command
managed_image_resource_group_name	Name of resource group you created in the first step
managed_image_name	Name for the managed disk image that is created

JSON

```
{
  "builders": [{
    "type": "azure-arm",

    "client_id": "f5b6a5cf-fbdf-4a9f-b3b8-3c2cd00225a4",
    "client_secret": "0e760437-bf34-4aad-9f8d-870be799c55d",
    "tenant_id": "72f988bf-86f1-41af-91ab-2d7cd011db47",
    "subscription_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",

    "managed_image_resource_group_name": "myResourceGroup",
    "managed_image_name": "myPackerImage",

    "os_type": "Linux",
    "image_publisher": "canonical",
    "image_offer": "0001-com-ubuntu-server-jammy",
    "image_sku": "22_04-lts",

    "azure_tags": {
        "dept": "Engineering",
        "task": "Image deployment"
    },

    "location": "East US",
    "vm_size": "Standard_DS2_v2"
  }],
  "provisioners": [{
    "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo -E sh '{{
.Path }}'",
    "inline": [
      "apt-get update",
      "apt-get upgrade -y",
      "apt-get -y install nginx",

      "/usr/sbin/waagent -force -deprovision+user && export HISTSIZE=0 &&
```

```
sync"
    ],
    "inline_shebang": "/bin/sh -x",
    "type": "shell"
  }]
}
```

> **Note**
>
> Replace the `image_publisher`, `image_offer`, `image_sku` values and `inline` commands accordingly.

You can also create a filed named *ubuntu.pkr.hcl* and paste the following content with your own values as used for the above parameters table.

HCL

```
source "azure-arm" "autogenerated_1" {
  azure_tags = {
    dept = "Engineering"
    task = "Image deployment"
  }
  client_id                         = "f5b6a5cf-fbdf-4a9f-b3b8-3c2cd00225a4"
  client_secret                     = "0e760437-bf34-4aad-9f8d-870be799c55d"
  image_offer                       = "0001-com-ubuntu-server-jammy"
  image_publisher                   = "canonical"
  image_sku                         = "22_04-lts"
  location                          = "East US"
  managed_image_name                = "myPackerImage"
  managed_image_resource_group_name = "myResourceGroup"
  os_type                           = "Linux"
  subscription_id                   = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
  tenant_id                         = "72f988bf-86f1-41af-91ab-2d7cd011db47"
  vm_size                           = "Standard_DS2_v2"
}

build {
  sources = ["source.azure-arm.autogenerated_1"]

  provisioner "shell" {
    execute_command = "chmod +x {{ .Path }}; {{ .Vars }} sudo -E sh '{{
.Path }}'"
    inline          = ["apt-get update", "apt-get upgrade -y", "apt-get -y
install nginx", "/usr/sbin/waagent -force -deprovision+user && export
HISTSIZE=0 && sync"]
    inline_shebang  = "/bin/sh -x"
  }

}
```

This template builds an Ubuntu 22.04 LTS image, installs NGINX, then deprovisions the VM.

> **Note**
>
> If you expand on this template to provision user credentials, adjust the provisioner command that deprovisions the Azure agent to read `-deprovision` rather than `deprovision+user`. The `+user` flag removes all user accounts from the source VM.

Build Packer image

If you don't already have Packer installed on your local machine, follow the Packer installation instructions.

Build the image by specifying your Packer template file as follows:

Bash

```
sudo ./packer build ubuntu.json
```

You can also build the image by specifying the *ubuntu.pkr.hcl* file as follows:

Bash

```
sudo packer build ubuntu.pkr.hcl
```

An example of the output from the preceding commands is as follows:

Output

```
azure-arm output will be in this color.

==> azure-arm: Running builder ...
    azure-arm: Creating Azure Resource Manager (ARM) client ...
==> azure-arm: Creating resource group ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-swtxmqm7ly’
==> azure-arm:  -> Location          : ‘East US’
==> azure-arm:  -> Tags              :
==> azure-arm:  ->> dept : Engineering
==> azure-arm:  ->> task : Image deployment
==> azure-arm: Validating deployment template ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-swtxmqm7ly’
==> azure-arm:  -> DeploymentName    : ‘pkrdpswtxmqm7ly’
==> azure-arm: Deploying deployment template ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-swtxmqm7ly’
```

```
==> azure-arm:  -> DeploymentName    : ‘pkrdpswtxmqm7ly’
==> azure-arm: Getting the VM’s IP address ...
==> azure-arm:  -> ResourceGroupName   : ‘packer-Resource-Group-swtxmqm7ly’
==> azure-arm:  -> PublicIPAddressName : ‘packerPublicIP’
==> azure-arm:  -> NicName             : ‘packerNic’
==> azure-arm:  -> Network Connection  : ‘PublicEndpoint’
==> azure-arm:  -> IP Address          : ‘40.76.218.147’
==> azure-arm: Waiting for SSH to become available...
==> azure-arm: Connected to SSH!
==> azure-arm: Provisioning with shell script:
/var/folders/h1/ymh5bdx15wgdn5hvgj1wc0zh0000gn/T/packer-shell868574263
    azure-arm: WARNING! The waagent service will be stopped.
    azure-arm: WARNING! Cached DHCP leases will be deleted.
    azure-arm: WARNING! root password will be disabled. You will not be able
to login as root.
    azure-arm: WARNING! /etc/resolvconf/resolv.conf.d/tail and
/etc/resolvconf/resolv.conf.d/original will be deleted.
    azure-arm: WARNING! packer account and entire home directory will be
deleted.
==> azure-arm: Querying the machine’s properties ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-swtxmqm7ly’
==> azure-arm:  -> ComputeName       : ‘pkrvmswtxmqm7ly’
==> azure-arm:  -> Managed OS Disk   :
‘/subscriptions/guid/resourceGroups/packer-Resource-Group-
swtxmqm7ly/providers/Microsoft.Compute/disks/osdisk’
==> azure-arm: Powering off machine ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-swtxmqm7ly’
==> azure-arm:  -> ComputeName       : ‘pkrvmswtxmqm7ly’
==> azure-arm: Capturing image ...
==> azure-arm:  -> Compute ResourceGroupName : ‘packer-Resource-Group-
swtxmqm7ly’
==> azure-arm:  -> Compute Name              : ‘pkrvmswtxmqm7ly’
==> azure-arm:  -> Compute Location          : ‘East US’
==> azure-arm:  -> Image ResourceGroupName   : ‘myResourceGroup’
==> azure-arm:  -> Image Name                : ‘myPackerImage’
==> azure-arm:  -> Image Location            : ‘eastus’
==> azure-arm: Deleting resource group ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-swtxmqm7ly’
==> azure-arm: Deleting the temporary OS disk ...
==> azure-arm:  -> OS Disk : skipping, managed disk was used...
Build ‘azure-arm’ finished.

==> Builds finished. The artifacts of successful builds are:
--> azure-arm: Azure.ResourceManagement.VMImage:

ManagedImageResourceGroupName: myResourceGroup
ManagedImageName: myPackerImage
ManagedImageLocation: eastus
```

It takes a few minutes for Packer to build the VM, run the provisioners, and clean up the deployment.

Create VM from Azure Image

You can now create a VM from your Image with az vm create. Specify the Image you created with the `--image` parameter. The following example creates a VM named *myVM* from *myPackerImage* and generates SSH keys if they don't already exist:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image myPackerImage \
    --admin-username azureuser \
    --generate-ssh-keys
```

If you wish to create VMs in a different resource group or region than your Packer image, specify the image ID rather than image name. You can obtain the image ID with az image show.

It takes a few minutes to create the VM. Once the VM has been created, take note of the `publicIpAddress` displayed by the Azure CLI. This address is used to access the NGINX site via a web browser.

To allow web traffic to reach your VM, open port 80 from the Internet with az vm open-port:

Azure CLI

```
az vm open-port \
    --resource-group myResourceGroup \
    --name myVM \
    --port 80
```

Test VM and NGINX

Now you can open a web browser and enter `http://publicIpAddress` in the address bar. Provide your own public IP address from the VM create process. The default NGINX page is displayed as in the following example:



Next steps

You can also use existing Packer provisioner scripts with Azure Image Builder.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

PowerShell: How to use Packer to create virtual machine images in Azure

Article • 10/28/2024

Applies to: ✔ Windows VMs

Each virtual machine (VM) in Azure is created from an image that defines the Windows distribution and OS version. Images can include pre-installed applications and configurations. The Azure Marketplace provides many first and third-party images for most common OS' and application environments, or you can create your own custom images tailored to your needs. This article details how to use the open-source tool Packer to define and build custom images in Azure.

This article was last tested on 8/5/2020 using Packer version 1.8.1.

> **Note**
>
> Azure now has a service, Azure Image Builder, for defining and creating your own custom images. Azure Image Builder is built on Packer, so you can even use your existing Packer shell provisioner scripts with it. To get started with Azure Image Builder, see Create a Windows VM with Azure Image Builder.

Create Azure resource group

During the build process, Packer creates temporary Azure resources as it builds the source VM. To capture that source VM for use as an image, you must define a resource group. The output from the Packer build process is stored in this resource group.

Create a resource group with New-AzResourceGroup. The following example creates a resource group named *myPackerGroup* in the *eastus* location:

Azure PowerShell

```
$rgName = "myPackerGroup"
$location = "East US"
New-AzResourceGroup -Name $rgName -Location $location
```

Create Azure credentials

Packer authenticates with Azure using a service principal. An Azure service principal is a security identity that you can use with apps, services, and automation tools like Packer. You control and define the permissions as to what operations the service principal can perform in Azure.

Create a service principal with New-AzADServicePrincipal. The value for `-DisplayName` needs to be unique; replace with your own value as needed.

Azure PowerShell

```
$sp = New-AzADServicePrincipal -DisplayName "PackerPrincipal" -role
Contributor -scope /subscriptions/yyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyy
$plainPassword = (New-AzADSpCredential -ObjectId $sp.Id).SecretText
```

Then output the password and application ID.

Azure PowerShell

```
$plainPassword
$sp.AppId
```

To authenticate to Azure, you also need to obtain your Azure tenant and subscription IDs with Get-AzSubscription:

Azure PowerShell

```
$subName = "mySubscriptionName"
$sub = Get-AzSubscription -SubscriptionName $subName
```

Define Packer template

To build images, you create a template as a JSON file. In the template, you define builders and provisioners that carry out the actual build process. Packer has a builder for Azure that allows you to define Azure resources, such as the service principal credentials created in the preceding step.

Create a file named *windows.json* and paste the following content. Enter your own values for the following:

Expand table

Parameter	Where to obtain
client_id	View service principal ID with `$sp.AppId`
client_secret	View the auto-generated password with `$plainPassword`
tenant_id	Output from `$sub.TenantId` command
subscription_id	Output from `$sub.SubscriptionId` command
managed_image_resource_group_name	Name of resource group you created in the first step
managed_image_name	Name for the managed disk image that is created

JSON

```
{
  "builders": [{
    "type": "azure-arm",

    "client_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
    "client_secret": "ppppppp-pppp-pppp-pppp-pppppppppppp",
    "tenant_id": "zzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz",
    "subscription_id": "yyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyy",

    "managed_image_resource_group_name": "myPackerGroup",
    "managed_image_name": "myPackerImage",

    "os_type": "Windows",
    "image_publisher": "MicrosoftWindowsServer",
    "image_offer": "WindowsServer",
    "image_sku": "2016-Datacenter",

    "communicator": "winrm",
    "winrm_use_ssl": true,
    "winrm_insecure": true,
    "winrm_timeout": "5m",
    "winrm_username": "packer",

    "azure_tags": {
        "dept": "Engineering",
        "task": "Image deployment"
    },

    "build_resource_group_name": "myPackerGroup",
    "vm_size": "Standard_D2_v2"
  }],
  "provisioners": [{
    "type": "powershell",
    "inline": [
      "Add-WindowsFeature Web-Server",
      "while ((Get-Service RdAgent).Status -ne 'Running') { Start-Sleep -s 5
}",
```

```
      "while ((Get-Service WindowsAzureGuestAgent).Status -ne 'Running') {
Start-Sleep -s 5 }",
      "& $env:SystemRoot\\System32\\Sysprep\\Sysprep.exe /oobe /generalize
/quiet /quit",
      "while($true) { $imageState = Get-ItemProperty
HKLM:\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Setup\\State | Select
ImageState; if($imageState.ImageState -ne
'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { Write-Output
$imageState.ImageState; Start-Sleep -s 10  } else { break } }"
    ]
  }]
}
```

You can also create a file named *windows.pkr.hcl* and paste the following content with your own values as used for the above parameters table.

HCL

```
source "azure-arm" "autogenerated_1" {
  azure_tags = {
    dept = "Engineering"
    task = "Image deployment"
  }
  build_resource_group_name         = "myPackerGroup"
  client_id                         = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
  client_secret                     = "ppppppp-pppp-pppp-pppp-pppppppppppp"
  communicator                      = "winrm"
  image_offer                       = "WindowsServer"
  image_publisher                   = "MicrosoftWindowsServer"
  image_sku                         = "2016-Datacenter"
  managed_image_name                = "myPackerImage"
  managed_image_resource_group_name = "myPackerGroup"
  os_type                           = "Windows"
  subscription_id                   = "yyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyy"
  tenant_id                         = "zzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz"
  vm_size                           = "Standard_D2_v2"
  winrm_insecure                    = true
  winrm_timeout                     = "5m"
  winrm_use_ssl                     = true
  winrm_username                    = "packer"
}

build {
  sources = ["source.azure-arm.autogenerated_1"]

  provisioner "powershell" {
    inline = ["Add-WindowsFeature Web-Server", "while ((Get-Service
RdAgent).Status -ne 'Running') { Start-Sleep -s 5 }", "while ((Get-Service
WindowsAzureGuestAgent).Status -ne 'Running') { Start-Sleep -s 5 }", "&
$env:SystemRoot\\System32\\Sysprep\\Sysprep.exe /oobe /generalize /quiet
/quit", "while($true) { $imageState = Get-ItemProperty
HKLM:\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Setup\\State | Select
ImageState; if($imageState.ImageState -ne
```

```
'IMAGE_STATE_GENERALIZE_RESEAL_TO_OOBE') { Write-Output
$imageState.ImageState; Start-Sleep -s 10  } else { break } }"]
  }

}
```

This template builds a Windows Server 2016 VM, installs IIS, then generalizes the VM with Sysprep. The IIS install shows how you can use the PowerShell provisioner to run additional commands. The final Packer image then includes the required software install and configuration.

The Windows Guest Agent participates in the Sysprep process. The agent must be fully installed before the VM can be sysprep'ed. To ensure that this is true, all agent services must be running before you execute sysprep.exe. The preceding JSON snippet shows one way to do this in the PowerShell provisioner. This snippet is required only if the VM is configured to install the agent, which is the default.

Build Packer image

If you don't already have Packer installed on your local machine, follow the Packer installation instructions.

Build the image by opening a cmd prompt and specifying your Packer template file as follows:

Batchfile

```
packer build windows.json
```

You can also build the image by specifying the *windows.pkr.hcl* file as follows:

Batchfile

```
packer build windows.pkr.hcl
```

An example of the output from the preceding commands is as follows:

HCL

```
azure-arm output will be in this color.

==> azure-arm: Running builder ...
    azure-arm: Creating Azure Resource Manager (ARM) client ...
==> azure-arm: Creating resource group ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
```

```
==> azure-arm:  -> Location          : ‘East US’
==> azure-arm:  -> Tags              :
==> azure-arm:  ->> task : Image deployment
==> azure-arm:  ->> dept : Engineering
==> azure-arm: Validating deployment template ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm:  -> DeploymentName    : ‘pkrdppq0mthtbtt’
==> azure-arm: Deploying deployment template ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm:  -> DeploymentName    : ‘pkrdppq0mthtbtt’
==> azure-arm: Getting the certificate’s URL ...
==> azure-arm:  -> Key Vault Name        : ‘pkrkvpq0mthtbtt’
==> azure-arm:  -> Key Vault Secret Name : ‘packerKeyVaultSecret’
==> azure-arm:  -> Certificate URL       :
‘https://pkrkvpq0mthtbtt.vault.azure.net/secrets/packerKeyVaultSecret/8c7bd8
23e4fa44e1abb747636128adbb'
==> azure-arm: Setting the certificate’s URL ...
==> azure-arm: Validating deployment template ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm:  -> DeploymentName    : ‘pkrdppq0mthtbtt’
==> azure-arm: Deploying deployment template ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm:  -> DeploymentName    : ‘pkrdppq0mthtbtt’
==> azure-arm: Getting the VM’s IP address ...
==> azure-arm:  -> ResourceGroupName   : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm:  -> PublicIPAddressName : ‘packerPublicIP’
==> azure-arm:  -> NicName             : ‘packerNic’
==> azure-arm:  -> Network Connection  : ‘PublicEndpoint’
==> azure-arm:  -> IP Address          : ‘40.76.55.35’
==> azure-arm: Waiting for WinRM to become available...
==> azure-arm: Connected to WinRM!
==> azure-arm: Provisioning with Powershell...
==> azure-arm: Provisioning with shell script:
/var/folders/h1/ymh5bdx15wgdn5hvgj1wc0zh0000gn/T/packer-powershell-
provisioner902510110
    azure-arm: #< CLIXML
    azure-arm:
    azure-arm: Success Restart Needed Exit Code      Feature Result
    azure-arm: ------- -------------- ---------      --------------
    azure-arm: True    No             Success        {Common HTTP Features,
Default Document, D...
    azure-arm: <Objs Version=“1.1.0.1”
xmlns=“http://schemas.microsoft.com/powershell/2004/04"><Obj S=“progress”
RefId=“0"><TN RefId=“0”><T>System.Management.Automation.PSCustomObject</T>
<T>System.Object</T></TN><MS><I64 N=“SourceId”>1</I64><PR N=“Record”>
<AV>Preparing modules for first use.</AV><AI>0</AI><Nil /><PI>-1</PI>
<PC>-1</PC><T>Completed</T><SR>-1</SR><SD> </SD></PR></MS></Obj></Objs>
==> azure-arm: Querying the machine’s properties ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm:  -> ComputeName       : ‘pkrvmpq0mthtbtt’
==> azure-arm:  -> Managed OS Disk   :
‘/subscriptions/guid/resourceGroups/packer-Resource-Group-
pq0mthtbtt/providers/Microsoft.Compute/disks/osdisk’
==> azure-arm: Powering off machine ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
```

```
==> azure-arm:  -> ComputeName       : ‘pkrvmpq0mthtbtt’
==> azure-arm: Capturing image ...
==> azure-arm:  -> Compute ResourceGroupName : ‘packer-Resource-Group-
pq0mthtbtt’
==> azure-arm:  -> Compute Name              : ‘pkrvmpq0mthtbtt’
==> azure-arm:  -> Compute Location          : ‘East US’
==> azure-arm:  -> Image ResourceGroupName   : ‘myResourceGroup’
==> azure-arm:  -> Image Name                : ‘myPackerImage’
==> azure-arm:  -> Image Location            : ‘eastus’
==> azure-arm: Deleting resource group ...
==> azure-arm:  -> ResourceGroupName : ‘packer-Resource-Group-pq0mthtbtt’
==> azure-arm: Deleting the temporary OS disk ...
==> azure-arm:  -> OS Disk : skipping, managed disk was used...
Build ‘azure-arm’ finished.

==> Builds finished. The artifacts of successful builds are:
--> azure-arm: Azure.ResourceManagement.VMImage:

ManagedImageResourceGroupName: myResourceGroup
ManagedImageName: myPackerImage
ManagedImageLocation: eastus
```

It takes a few minutes for Packer to build the VM, run the provisioners, and clean up the deployment.

Create a VM from the Packer image

You can now create a VM from your Image with New-AzVM. The supporting network resources are created if they do not already exist. When prompted, enter an administrative username and password to be created on the VM. The following example creates a VM named *myVM* from *myPackerImage*:

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName $rgName `
    -Name "myVM" `
    -Location $location `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress" `
    -OpenPorts 80 `
    -Image "myPackerImage"
```

If you wish to create VMs in a different resource group or region than your Packer image, specify the image ID rather than image name. You can obtain the image ID with Get-AzImage.

It takes a few minutes to create the VM from your Packer image.

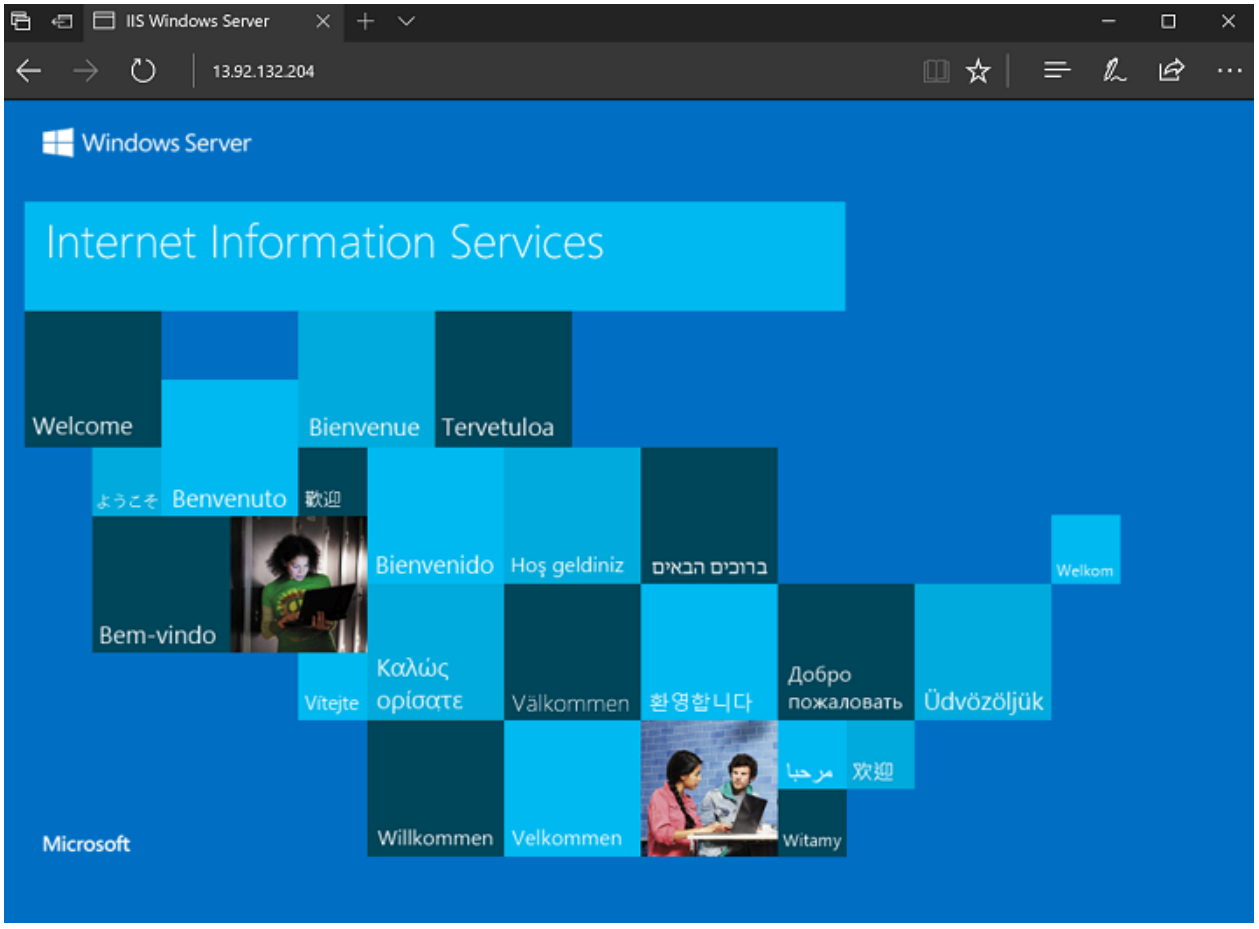
Test VM and webserver

Obtain the public IP address of your VM with Get-AzPublicIPAddress. The following example obtains the IP address for *myPublicIP* created earlier:

Azure PowerShell

```
Get-AzPublicIPAddress `
    -ResourceGroupName $rgName `
    -Name "myPublicIPAddress" | select "IpAddress"
```

To see your VM, that includes the IIS install from the Packer provisioner, in action, enter the public IP address in to a web browser.



Next steps

You can also use existing Packer provisioner scripts with Azure Image Builder.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Dedicated Hosts

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale sets

Azure Dedicated Host is a service that provides physical servers able to host one or more virtual machines assigned to one Azure subscription. Dedicated hosts are the same physical servers used in our data centers, provided instead as a directly accessible hardware resource. You can provision dedicated hosts within a region, availability zone, and fault domain. You can then place VMs directly into your provisioned hosts in whatever configuration best meets your needs.

Video Introduction

https://www.youtube-nocookie.com/embed/Lk9mA1WzfAQ

Benefits

Reserving the entire host provides several benefits beyond those of a standard shared virtual machine host:

- Cost Optimization: With the Azure hybrid benefit, you can bring your own licenses for Windows and SQL to Azure. For more information, see Azure Hybrid Benefit.
- Reliability: You have near complete control over maintenance events initiated by the Azure platform. While most maintenance events have little to no impact on your virtual machines, there are some sensitive workloads where each second of pause can have an impact. With dedicated hosts, you can opt in to a maintenance window to reduce the impact to your service.
- Performance Efficiency: Because you have control over a physical host, you can choose which applications share physical resources such as memory and storage. This can speed up certain workloads that benefit from low latency and high throughput on the host machine.
- Security: Hardware isolation at the physical server level allows for sensitive memory data to remain isolated within a physical host. No other customer's VMs will be placed on your hosts. Dedicated hosts are deployed in the same data centers and share the same network and underlying storage infrastructure as other, non-isolated hosts.

Groups, hosts, and VMs



A **host group** is a resource that represents a collection of dedicated hosts. You create a host group in a region and an availability zone, and add hosts to it.

A **host** is a resource, mapped to a physical server in an Azure data center. The physical server is allocated when the host is created. A host is created within a host group. A host has a SKU describing which VM sizes can be created. Each host can host multiple VMs, of different sizes, as long as they are from the same size series.

High availability considerations

For high availability, you should deploy multiple VMs, spread across multiple hosts (minimum of 2). With Azure Dedicated Hosts, you have several options to provision your infrastructure to shape your fault isolation boundaries.

Use availability zones for fault isolation

Availability zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. A host group is created in a single availability zone. Once created, all hosts will be placed within that zone. To achieve high availability across zones, you need to create multiple host groups (one per zone) and spread your hosts between them accordingly.

If you assign a host group to an availability zone, all VMs created on that host must be created in the same zone.

Use Fault Domains for fault isolation

A host can be created in a specific fault domain. Just like VM in a scale set or availability set, hosts in different fault domains will be placed on different physical racks in the data center. When you create a host group, you're required to specify the fault domain count. When creating hosts within the host group, you assign fault domain for each host. The VMs don't require any fault domain assignment.

Fault domains aren't the same as colocation. Having the same fault domain for two hosts doesn't mean they are in proximity with each other.

Fault domains are scoped to the host group. You shouldn't make any assumption on anti-affinity between two host groups (unless they are in different availability zones).

VMs deployed to hosts with different fault domains, will have their underlying managed disks services on multiple storage stamps, to increase the fault isolation protection.

Using Availability Zones and Fault Domains

You can use both capabilities together to achieve even more fault isolation. To use both, specify the availability zone and fault domain count in for each host group, assign a fault domain to each host in the group, then assign an availability zone to each VM.

The Resource Manager sample template uses zones and fault domains to spread hosts for maximum resiliency in a region.

Manual vs. automatic placement

When creating a VM in Azure, you can select which dedicated host to use. You can also use the option to automatically place your VMs on existing hosts, within a host group.

When creating a new host group, make sure the setting for automatic VM placement is selected. When creating your VM, select the host group and let Azure pick the best host for your VM.

Host groups that are enabled for automatic placement don't require all the VMs to be automatically placed. You'll still be able to explicitly pick a host, even when automatic placement is selected for the host group.

Automatic placement limitations

Known issues and limitations when using automatic VM placement:

- You won't be able to redeploy your VM.
- You won't be able to use DCv2, Lsv2, NVasv4, NVsv3, Msv2, or M-series VMs with dedicated hosts.

Host service healing

In case of any failure relating to the underlying node, network connectivity or software issues can push the host and VMs on the host to a non-healthy state causing disruption and downtime to your workloads. The default action is for Azure to automatically service heal the impacted host to a healthy node and move all VMs to the healthy host. Once the VMs are service healed and restarted the impacted host will be deallocated. During the service healing process the host and VMs would become unavailable incurring a slight downtime.

The newly created host would have all the same constraints as the old host:

- Resource group
- Region
- Fault Domain
- Host Group
- ADH SKU
- Auto replace on failure setting

Users with compliance requirements might need a strong affinity between the host and underlying node and would not like to be automatically service healed, in such scenarios users can choose to opt out of auto service healing at host level by disabling the 'Automatically replace host on failure setting'.

Implications

If you decide to disable auto service healing and if the underlying node encounters a failure your host state will change to 'Host Pending Deallocate' and will eventually be deallocated.

To avoid deallocation, you would need to manually redeploy the host by creating a new dedicated host and moving all the VMs from the old host to the new host.

The auto replace host setting is a create time setting and cannot be changed once the host is created. VMs that are manually stopped/deallocated from the impacted host are not moved as part of the automatic service healing.

Virtual Machine Scale Set support

Virtual Machine Scale Sets let you treat a group of virtual machines as a single resource, and apply availability, management, scaling and orchestration policies as a group. Your existing dedicated hosts can also be used for Virtual Machine Scale Sets.

When creating a Virtual Machine Scale Set, you can specify an existing host group to have all of the VM instances created on dedicated hosts.

The following requirements apply when creating a Virtual Machine Scale Set in a dedicated host group:

- Automatic VM placement needs to be enabled.
- The availability setting of your host group should match your scale set.
  - A regional host group (created without specifying an availability zone) should be used for regional scale sets.
  - The host group and the scale set must be using the same availability zone.
  - The fault domain count for the host group level should match the fault domain count for your scale set. The Azure portal lets you specify *max spreading* for your scale set, which sets the fault domain count of 1.
- Dedicated hosts should be created first, with sufficient capacity, and the same settings for scale set zones and fault domains.
- The supported VM sizes for your dedicated hosts should match the one used for your scale set.

Not all scale-set orchestration and optimizations settings are supported by dedicated hosts. Apply the following settings to your scale set:

- Overprovisioning isn't recommended, and it's disabled by default. You can enable overprovisioning, but the scale set allocation will fail if the host group doesn't have capacity for all of the VMs, including the overprovisioned instances.
- Use the ScaleSetVM orchestration mode
- Don't use proximity placement groups for co-location

Ultra disk support for virtual machines on dedicated hosts

Ultra disks offer higher IOPS (maximum of 160,00 IOPS per disk) and higher disk throughput (maximum of 4000 MBps per disk) depending on the disk size. All the limitations of ultra disks would still apply. To use ultra disks on dedicated hosts following requirements needs to be satisfied:

- Host group needs to have Ultra SSD 'Enabled', this property cannot be changed once the host group is created
- Dedicated host's virtual machine series should be compatible with ultra disks.

Dedicated hosts also support ultra disks with specialty VM sizes: LSv2, M, Mv2, Msv2, Mdsv2, NVv3, NVv4. However in these cases below listed additional limitations would

apply:

- Host group should have Automatic Placement 'Disabled'
- Host group's fault domain count should be set to '1' only.

Maintenance control

The infrastructure supporting your virtual machines may occasionally be updated to improve reliability, performance, security, and to launch new features. The Azure platform tries to minimize the impact of platform maintenance whenever possible, however customers with *maintenance sensitive* workloads can't tolerate even few seconds that the VM needs to be shut down for maintenance.

Maintenance Control provides customers with an option to skip regular platform updates scheduled on their dedicated hosts, then apply it at the time of their choice within a 35-day rolling window. Within the maintenance window, you can apply maintenance directly at the host level, in any order. Once the maintenance window is over, Microsoft will move forward and apply the pending maintenance to the hosts in an order that may not follow the user defined fault domains.

For more information, see Managing platform updates with Maintenance Control.

Capacity considerations

Once a dedicated host is provisioned, Azure assigns it to physical server. Doing so guarantees the availability of the capacity when you need to provision your VM. Azure uses the entire capacity in the region (or zone) to pick a physical server for your host. It also means that customers can expect to be able to grow their dedicated host footprint without the concern of running out of space in the cluster.

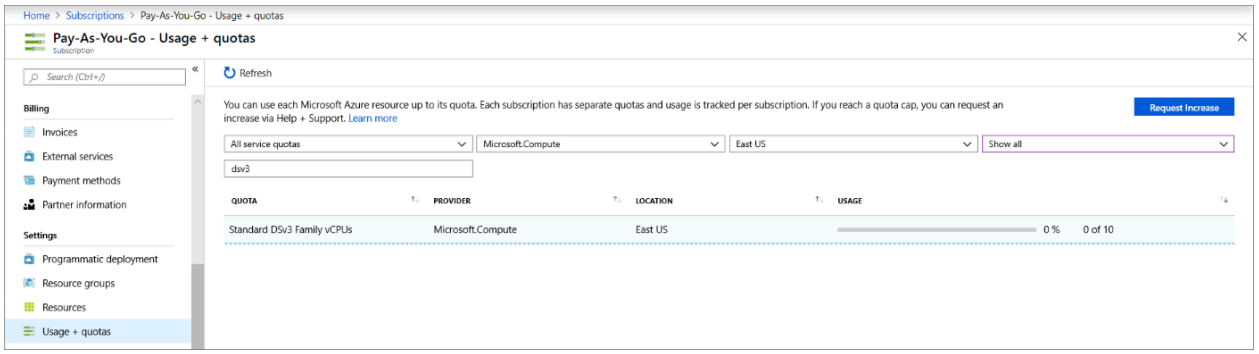
Quotas

There are two types of quota that are consumed when you deploy a dedicated host.

1. Dedicated host vCPU quota. The default quota limit is 3000 vCPUs, per region.
2. VM size family quota. For example, a **Pay-as-you-go** subscription may only have a quota of 10 vCPUs available for the Dsv3 size series, in the East US region. To deploy a Dsv3 dedicated host, you would need to request a quota increase to at least 64 vCPUs before you can deploy the dedicated host.

To request a quota increase, create a support request in the Azure portal.

Provisioning a dedicated host will consume both dedicated host vCPU and the VM family vCPU quota, but it won't consume the regional vCPU. VMs placed on a dedicated host won't count against VM family vCPU quota. Should a VM be moved off a dedicated host into a multi-tenant environment, the VM will consume VM family vCPU quota.



For more information, see Virtual machine vCPU quotas.

Free trial and MSDN subscriptions don't have quota for Azure Dedicated Hosts.

Pricing

Users are charged per dedicated host, regardless how many VMs are deployed. In your monthly statement, you'll see a new billable resource type of hosts. The VMs on a dedicated host will still be shown in your statement, but will carry a price of 0.

The host price is set based on VM family, type (hardware size), and region. A host price is relative to the largest VM size supported on the host.

Software licensing, storage and network usage are billed separately from the host and VMs. There's no change to those billable items.

For more information, see Azure Dedicated Host pricing.

You can also save on costs with a Reserved Instance of Azure Dedicated Hosts.

Sizes and hardware generations

A SKU represents the VM size series and type on a given host. You can mix multiple VMs of different sizes within a single host as long as they are of the same size series.

The *type* is the hardware generation. Different hardware types for the same VM series will be from different CPU vendors and have different CPU generations and number of cores.

The sizes and hardware types vary by region. Refer to the host pricing page to learn more.

> **Note**
>
> Once a Dedicated host is provisioned, you can't change the size or type. If you need a different size of type, you will need to create a new host.

Host life cycle

Azure monitors and manages the health status of your hosts. The following states will be returned when you query your host:

Expand table

Health State	Description
Host Available	There are no known issues with your host.
Host Under Investigation	We're having some issues with the host that we're looking into. This transitional state is required for Azure to try to identify the scope and root cause for the issue identified. Virtual machines running on the host may be impacted.
Host Pending Deallocate	Azure can't restore the host back to a healthy state and ask you to redeploy your virtual machines out of this host. If `autoReplaceOnFailure` is enabled, your virtual machines are *service healed* to healthy hardware. Otherwise, your virtual machine may be running on a host that is about to fail.
Host Deallocated	All virtual machines have been removed from the host. You're no longer being charged for this host since the hardware was taken out of rotation.

Frequently Asked Questions

Q. What happens to my dedicated host in case of a live migration?

A. As of today, Azure dedicated hosts do not support live migration and in case of a hardware failure, we service heal the host to a different node.

Q. Can I run VMs from multiple VM families on the same dedicated host?

A. No, you would be able to run only VMs from the same family as the underlying dedicated host. For e.g., A Dsv3-Type4 host only supports VMs of Dsv3 VM family.

Q. Would I be able to run different VM sizes on a single dedicated host?

A. Yes, you can run multiple sizes of VMs on the same dedicated host as long as the all the VMs belong to the same family as the underlying dedicated host and there is enough capacity on the host to support the VMs sizes. For e.g., on a Dsv3-Type4 host you could run D2sv3, D8sv3, D16sv3 VMs at the same time.

Next steps

- To deploy a dedicated host, see Deploy VMs and scale sets to dedicated hosts.
- There's a sample template that uses both zones and fault domains for maximum resiliency in a region.
- You can also save on costs with a Reserved Instance of Azure Dedicated Hosts.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy VMs and scale sets to dedicated hosts

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale sets

This article guides you through how to create an Azure dedicated host to host your virtual machines (VMs) and scale set instances.

Limitations

- The sizes and hardware types available for dedicated hosts vary by region. Refer to the host pricing page to learn more.
- Not all Azure VM SKUs, regions and availability zones support ultra disks, for more information about this topic, see Azure ultra disks.
- Additional limitations would apply when using ultra disks on the following VM sizes: LSv2, M, Mv2, Msv2, Mdsv2, NVv3, NVv4 on a dedicated host.
- The fault domain count of the virtual machine scale set can't exceed the fault domain count of the host group.
- Users can not select hardware capabilities like accelerated networking when creating a dedicated host.
- Users would not be able to create VMs/VMSS with accelerated networking enabled on a dedicated host.

Create a host group

A **host group** is a resource that represents a collection of dedicated hosts. You create a host group in a region and an availability zone, and add hosts to it. You can use one or both of the following options with your dedicated hosts to ensure high availability:

- Span across multiple availability zones. In this case, you're required to have a host group in each of the zones you wish to use.
- Span across multiple fault domains, which are mapped to physical racks.

In either case, you need to provide the fault domain count for your host group. If you don't want to span fault domains in your group, use a fault domain count of 1.

You can also decide to use both availability zones and fault domains.

Enabling ultra disks is a host group level setting and can't be changed after a host group is created.

Portal

In this example, we'll create a host group using one availability zone and two fault domains.

1. Open the Azure portal.
2. Select **Create a resource** in the upper left corner.
3. Search for **Host group** and then select **Host Groups** from the results.
4. In the **Host Groups** page, select **Create**.
5. Select the subscription you would like to use, and then select **Create new** to create a new resource group.
6. Type *myDedicatedHostsRG* as the **Name** and then select **OK**.
7. For **Host group name**, type *myHostGroup*.
8. For **Location**, select **East US**.
9. For **Availability Zone**, select **1**.
10. Select **Enable Ultra SSD** to use ultra disks with supported Virtual Machines.
11. For **Fault domain count**, select **2**.
12. Select **Automatic placement** to automatically assign VMs and scale set instances to an available host in this group.
13. Select **Review + create** and then wait for validation.
14. Once you see the **Validation passed** message, select **Create** to create the host group.

It should only take a few moments to create the host group.

Create a dedicated host

Now create a dedicated host in the host group. In addition to a name for the host, you're required to provide the SKU for the host. Host SKU captures the supported VM series and the hardware generation for your dedicated host.

For more information about the host SKUs and pricing, see Azure Dedicated Host pricing.

If you set a fault domain count for your host group, you'll need to specify the fault domain for your host.

Portal

1. Select **Create a resource** in the upper left corner.
2. Search for **Dedicated host** and then select **Dedicated hosts** from the results.
3. In the **Dedicated Hosts** page, select **Create**.
4. Select the subscription you would like to use.
5. Select *myDedicatedHostsRG* as the **Resource group**.
6. In **Instance details**, type *myHost* for the **Name** and select *East US* for the location.
7. In **Hardware profile**, select *Standard Es3 family - Type 1* for the **Size family**, select *myHostGroup* for the **Host group** and then select *1* for the **Fault domain**. Leave the defaults for the rest of the fields.
8. Leave the **Automatically replace host on failure** setting *Enabled* to automatically service heal the host in case of any host level failure.
9. When you're done, select **Review + create** and wait for validation.
10. Once you see the **Validation passed** message, select **Create** to create the host.

Create a VM

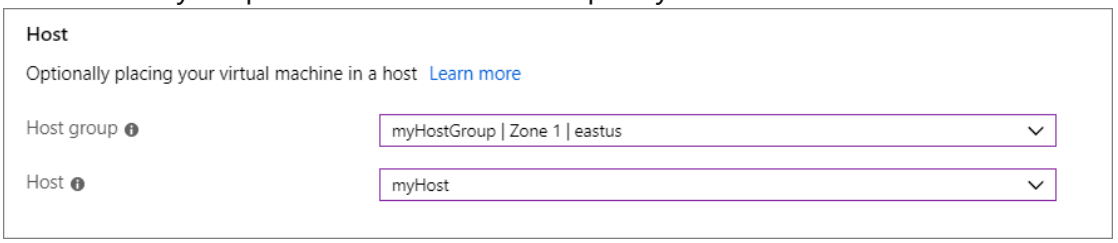
Now create a VM on the host.

If you would like to create a VM with ultra disks support, make sure the host group in which the VM will be placed is ultra SSD enabled. Once you've confirmed, create the VM in the same host group. See Deploy an ultra disk for the steps to attach an ultra disk to a VM.

Portal

1. Choose **Create a resource** in the upper left corner of the Azure portal.
2. In the search box above the list of Azure Marketplace resources, search for and select the image you want use, then choose **Create**.
3. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then select *myDedicatedHostsRG* as the **Resource group**.
4. Under **Instance details**, type *myVM* for the **Virtual machine name** and choose *East US* for your **Location**.
5. In **Availability options** select **Availability zone**, select *1* from the drop-down.
6. For the size, select **Change size**. In the list of available sizes, choose one from the Esv3 series, like **Standard E2s v3**. You may need to clear the filter in order to see all of the available sizes.
7. Complete the rest of the fields on the **Basics** tab as needed.

8. If you want to specify which host to use for your VM, then at the top of the page, select the **Advanced** tab and in the **Host** section, select *myHostGroup* for **Host group** and *myHost* for the **Host**. Otherwise, your VM will automatically be placed on a host with capacity.



9. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.
10. When you see the message that validation has passed, select **Create**.

It will take a few minutes for your VM to be deployed.

Create a scale set

You can also create a scale set on your host.

> **ⓘ Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Portal

When you deploy a scale set, you specify the host group.

1. Search for *Scale set* and select **Virtual machine scale sets** from the list.
2. Select **Add** to create a new scale set.
3. Complete the fields on the **Basics** tab as you usually would, but make sure you select a VM size that is from the series you chose for your dedicated host, like **Standard E2s v3**.
4. On the **Advanced** tab, for **Spreading algorithm** select **Max spreading**.
5. In **Host group**, select the host group from the drop-down. If you recently created the group, it might take a minute to get added to the list.

Reassign an existing VM

You can reassign an existing multitenant VM or dedicated host VM to a different dedicated host, but the VM must first be Stop\Deallocated. Before you move a VM to a dedicated host, make sure that the VM configuration is supported:

- The VM size must be in the same size family as the dedicated host. For example, if your dedicated host is DSv3, then the VM size could be Standard_D4s_v3, but it couldn't be a Standard_A4_v2.
- The VM needs to be located in same region as the dedicated host.
- The VM can't be part of a proximity placement group. Remove the VM from the proximity placement group before moving it to a dedicated host. For more information about this topic, see Move a VM out of a proximity placement group.
- The VM can't be in an availability set.
- If the VM is in an availability zone, it must be the same availability zone as the host group. The availability zone settings for the VM and the host group must match.

Portal

Move the VM to a dedicated host using the portal.

1. Open the page for the VM.
2. Select **Stop** to stop\deallocate the VM.
3. Select **Configuration** from the left menu.
4. Select a host group and a host from the drop-down menus.
5. When you're done, select **Save** at the top of the page.
6. After the VM has been added to the host, select **Overview** from the left menu.
7. At the top of the page, select **Start** to restart the VM.

Move a VM from dedicated host to multitenant infrastructure

You can move a VM that is running on a dedicated host to multitenant infrastructure, but the VM must first be Stop\Deallocated.

- Make sure that your subscription has sufficient vCPU quota for the VM in the region where

- Your multitenant VM will be scheduled in the same region and zone as the dedicated host

Portal

Move a VM from dedicated host to multitenant infrastructure using the portal.

1. Open the page for the VM.
2. Select **Stop** to stop\deallocate the VM.
3. Select **Configuration** from the left menu.
4. Select **None** under host group drop-down menu.
5. When you're done, select **Save** at the top of the page.
6. After the VM has been reconfigured as a multitenant VM, select **Overview** from the left menu.
7. At the top of the page, select **Start** to restart the VM.

Check the status of the host

If you need to know how much capacity is still available on a how, you can check the status.

Portal

1. Search for and select the host.
2. In the **Overview** page for the host, scroll down to see the list of sizes still available for the host. It should look similar to:

Available VM capacity

VM Size	Number remaining
Standard_D2ds_v4	32
Standard_D4ds_v4	17
Standard_D8ds_v4	8
Standard_D16ds_v4	4
Standard_D32ds_v4	2
Standard_D48ds_v4	1
Standard_D64ds_v4	1

Restart a host

Restarting a host does not completely power off the host. When the host restarts, the underlying VMs will also restart. The host will remain on the same underlying physical hardware and both the host ID and asset ID will remain the same after the restart. The host SKU will also remain the same after the restart.

Portal

1. Search for and select the host.
2. In the top menu bar, select the **Restart** button.
3. In the **Essentials** section of the Host Resource Pane, Host Status will switch to **Host undergoing restart** during the restart.
4. Once the restart has completed, the Host Status will return to **Host available**.

Resize a host

Moving a host and all associated VMs to newer generation hardware can be done through the host resize feature. Resize simplifies the migration process and avoids having to manually create new hosts and move all VMs individually.

Resize limitations:

- Host can only be resized to an ADH within the same VM family. A Dsv3-Type3 host can be resized to Dsv3-Type4 but **not** to an **E**sv3-Type4.
- You can only resize to newer generation of hardware. A Dsv3-Type3 host can be resized to Dsv3-Type4 but **not** Dsv3-Type2.
- Resizing changes the 'Host Asset ID'. The 'Host ID' remains the same.
- The host and all associated VMs become unavailable during the resize operation.

> ⚠ **Warning**
>
> The resize operation causes the loss of any non-persistent data such as temp disk data. Save all your work to persistent data storage before triggering resize.

> ⓘ **Note**
>
> If the source host is already running on the latest hardware, 'Size' page would display an empty list. If you're looking for enhanced performance, consider

switching to a different VM family.

Portal

1. Search for and select the host.
2. In the left menu under **Settings**, select **Size**.
3. Once on the size page from the list of SKUs, select the desired SKU to resize to.
4. Selecting a target size from the list would enable **Resize** button on the bottom on the page.
5. Click **Resize**, host's 'Provisioning State' changes from 'Provisioning Succeeded' to 'Updating'
6. Once the resizing is complete, the host's 'Provisioning State' reverts to 'Provisioning Succeeded'

Redeploy a host [Preview]

If a VM or the underlying host remains unresponsive after following all the potential troubleshooting steps users can trigger service healing of the host and not wait for the platform to initiate the repair. Redeploying a host will move the host and all associated VMs to a different node of the same SKU. None of the host parameters would change except for the 'Host asset ID', which corresponds to the underlying Node Id.

> ⚠ **Warning**
>
> Redeploy operation involves service healing hence would result in loss of any non-persistent data such as data stored on ephemeral disks. Save your work before redeploying.

Portal

1. Search for and select the host.
2. In the top menu bar, select the **Redeploy** button.
3. In the **Essentials** section of the Host Resource Pane, host's provisioning state will switch to **Updating** during the redeploy operation.
4. Once the redeploy operation is completed, host's provisioning state will revert to **Provisioning succeeded**.

5. In the **Essentials** section of the Host Resource Pane, **Host asset ID** would be updated to a new ID

Deleting a host

You're being charged for your dedicated host even when no virtual machines are deployed on the host. You should delete any hosts you're currently not using to save costs.

You can only delete a host when there are no any longer virtual machines using it.

Portal

1. Search for and select the host.
2. In the left menu, select **Instances**.
3. Select and delete each virtual machine.
4. When all of the VMs have been deleted, go back to the **Overview** page for the host and select **Delete** from the top menu.
5. Once the host has been deleted, open the page for the host group and select **Delete host group**.

Next steps

- For more information about this topic, see the Dedicated hosts overview.
- There's sample template, available at Azure Quickstart Templates, which uses both zones and fault domains for maximum resiliency in a region.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

General Purpose Azure Dedicated Host SKUs

Article • 08/22/2024

Azure Dedicated Host SKUs are the combination of a VM family and a certain hardware specification. You can only deploy VMs of the VM series that the Dedicated Host SKU specifies. For example, on the Dsv3-Type3, you can only provision Dsv3-series VMs.

This document goes through the hardware specifications and VM packings for all general purpose Dedicated Host SKUs.

Limitations

The sizes and hardware types available for dedicated hosts vary by region. Refer to the host pricing page to learn more.

Dadsv5

Dadsv5-Type1

The Dadsv5-Type1 is a Dedicated Host SKU utilizing AMD's EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The Dadsv5-Type1 runs Dadsv5-series VMs. Refer to the VM size documentation to better understand specific VM performance information.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dadsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	D2ads v5	32
			D4ads v5	27
			D8ads v5	14
			D16ads v5	7
			D32ads v5	3

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			D48ads v5	2
			D64ads v5	1
			D96ads v5	1

Dasv5

Dasv5-Type1

The Dasv5-Type1 is a Dedicated Host SKU utilizing AMD's EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The Dasv5-Type1 runs Dasv5-series VMs. Refer to the VM size documentation to better understand specific VM performance information.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dasv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	D2as v5	32
			D4as v5	28
			D8as v5	14
			D16as v5	7
			D32as v5	3
			D48as v5	2
			D64as v5	1
			D96as v5	1

Dasv4

Dasv4-Type1

The Dasv4-Type1 is a Dedicated Host SKU utilizing AMD's 2.35 GHz EPYC™ 7452 processor. It offers 64 physical cores, 96 vCPUs, and 672 GiB of RAM. The Dasv4-Type1 runs Dasv4-series VMs. Refer to the VM size documentation to better understand specific VM performance information.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dasv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	96	672 GiB	D2as v4	32
			D4as v4	24
			D8as v4	12
			D16as v4	6
			D32as v4	3
			D48as v4	2
			D64as v4	1
			D96as v4	1

You can also mix multiple VM sizes on the Dasv4-Type1. The following are sample combinations of VM packings on the Dasv4-Type1:

- 1 D48asv4 + 3 D16asv4
- 1 D32asv4 + 2 D16asv4 + 8 D4asv4
- 20 D4asv4 + 8 D2asv4

Dasv4-Type2

The Dasv4-Type2 is a Dedicated Host SKU utilizing AMD's EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The Dasv4-Type2 runs Dasv4-series VMs. Refer to the VM size documentation to better understand specific VM performance information.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dasv4-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	D2as v4	32
			D4as v4	25
			D8as v4	12
			D16as v4	6
			D32as v4	3
			D48as v4	2
			D64as v4	1
			D96as v4	1

DCadsv5

DCadsv5-Type1

The DCadsv5-Type1 is a Dedicated Host SKU utilizing the AMD 3rd Generation EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The DCadsv5-Type1 runs DCadsv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an DCadsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	DC2ads v5	32
			DC4ads v5	27
			DC8ads v5	14
			DC16ads v5	7
			DC32ads v5	3
			DC48ads v5	2
			DC64ads v5	1
			DC96ads v5	1

DCasv5

DCasv5-Type1

The DCasv5-Type1 is a Dedicated Host SKU utilizing the AMD 3rd Generation EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The DCasv5-Type1 runs DCasv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an DCasv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	DC2as v5	32
			DC4as v5	28
			DC8as v5	14
			DC16as v5	7
			DC32as v5	3
			DC48as v5	2
			DC64as v5	1
			DC96as v5	1

DCsv3

DCsv3-Type1

The DCsv3-Type1 is a Dedicated Host SKU utilizing the 3rd Generation Intel® Xeon Scalable Processor 8370C. It offers 48 physical cores, 48 vCPUs, and 384 GiB of RAM. The DCsv3-Type1 runs DCsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a DCsv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
48	48	384 GiB	DC1s v3	32
			DC2s v3	24
			DC4s v3	12
			DC8s v3	6
			DC16s v3	3
			DC24s v3	2

DCdsv3

DCdsv3-Type1

The DCdsv3-Type1 is a Dedicated Host SKU utilizing the 3rd Generation Intel® Xeon Scalable Processor 8370C. It offers 48 physical cores, 48 vCPUs, and 384 GiB of RAM. The DCdsv3-Type1 runs DCdsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a DCdsv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
48	48	384 GiB	DC16ds v3	2
			DC24ds v3	1
			DC32ds v3	1
			DC48ds v3	1

DCsv2

DCsv2-Type1

The DCsv2-Type1 is a Dedicated Host SKU utilizing the Intel® Coffee Lake (Xeon® E-2288G with SGX technology) processor. It offers 8 physical cores, 8 vCPUs, and 64 GiB of RAM. The DCsv2-Type1 runs DCsv2-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a DCsv2-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
8	8	64 GiB	DC8 v2	1

Ddsv5

Ddsv5-Type1

The Ddsv5-Type1 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Ddsv5-Type1 runs Ddsv5-series VMs. Refer to the VM size documentation to better understand specific VM performance information.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Ddsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	D2ds v5	32
			D4ds v5	22
			D8ds v5	11
			D16ds v5	5
			D32ds v5	2
			D48ds v5	1
			D64ds v5	1
			D96ds v5	1

Ddsv4

Ddsv4-Type1

The Ddsv4-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Ddsv4-Type1 runs Ddsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Ddsv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	D2ds v4	32
			D4ds v4	17
			D8ds v4	8
			D16ds v4	4
			D32ds v4	2
			D48ds v4	1
			D64ds v4	1

You can also mix multiple VM sizes on the Ddsv4-Type1. The following are sample combinations of VM packings on the Ddsv4-Type1:

- 1 D48dsv4 + 4 D4dsv4 + 2 D2dsv4
- 1 D32dsv4 + 2 D16dsv4 + 1 D4dsv4
- 10 D4dsv4 + 14 D2dsv4

Ddsv4-Type2

The Ddsv4-Type2 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Ddsv4-Type2 runs Ddsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Ddsv4-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	D2ds v4	32

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			D4ds v4	19
			D8ds v4	9
			D16ds v4	4
			D32ds v4	2
			D48ds v4	1
			D64ds v4	1

Dsv5

Dsv5-Type1

The Dsv5-Type1 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Dsv5-Type1 runs Dsv5-series VMs. Refer to the VM size documentation to better understand specific VM performance information.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	D2s v5	32
			D4s v5	25
			D8s v5	12
			D16s v5	6
			D32s v5	3
			D48s v5	2
			D64s v5	1
			D96s v5	1

Dsv4

Dsv4-Type1

The Dsv4-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Dsv4-Type1 runs Dsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	D2s v4	32
			D4s v4	20
			D8s v4	10
			D16s v4	5
			D32s v4	2
			D48s v4	1
			D64s v4	1

You can also mix multiple VM sizes on the Dsv4-Type1. The following are sample combinations of VM packings on the Dsv4-Type1:

- 1 D64sv4 + 1 D16sv4
- 1 D32sv4 + 2 D16dsv4 + 2 D8sv4
- 10 D4sv4 + 20 D2sv4

Dsv4-Type2

The Dsv4-Type2 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Dsv4-Type2 runs Dsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv4-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	D2s v4	32
			D4s v4	25
			D8s v4	12
			D16s v4	6
			D32s v4	3
			D48s v4	2
			D64s v4	1

Dsv3

Dsv3-Type1

> ⓘ Note
>
> **The Dsv3-Type1 will be retired on June 30, 2023**. Refer to the dedicated host retirement guide to learn more.

The Dsv3-Type1 is a Dedicated Host SKU utilizing the Intel® Broadwell (2.3 GHz Xeon® E5-2673 v4) processor. It offers 40 physical cores, 64 vCPUs, and 256 GiB of RAM. The Dsv3-Type1 runs Dsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
40	64	256 GiB	D2s v3	32
			D4s v3	17
			D8s v3	8
			D16s v3	4

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			D32s v3	2
			D48s v3	1
			D64s v3	1

You can also mix multiple VM sizes on the Dsv3-Type1. The following are sample combinations of VM packings on the Dsv3-Type1:

- 1 D32sv3 + 1 D16sv3 + 1 D8sv3
- 1 D48sv3 + 3 D4sv3 + 2 D2sv3
- 10 D4sv3 + 12 D2sv3

Dsv3-Type2

> **Note**
>
> **The Dsv3-Type2 will be retired on June 30, 2023**. Refer to the dedicated host retirement guide to learn more.

The Dsv3-Type2 is a Dedicated Host SKU utilizing the Intel® Skylake (2.1 GHz Xeon® Platinum 8171M) processor. It offers 48 physical cores, 76 vCPUs, and 504 GiB of RAM. The Dsv3-Type2 runs Dsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv3-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
48	76	504 GiB	D2s v3	32
			D4s v3	20
			D8s v3	10
			D16s v3	5
			D32s v3	2
			D48s v3	1
			D64s v3	1

You can also mix multiple VM sizes on the Dsv3-Type2. The following are sample combinations of VM packings on the Dsv3-Type2:

- 1 D64sv3 + 2 D4vs3 + 2 D2sv3
- 1 D48sv3 + 4 D4sv3 + 6 D2sv3
- 12 D4sv3 + 14 D2sv3

Dsv3-Type3

The Dsv3-Type3 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Dsv3-Type3 runs Dsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv3-Type3 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	D2s v3	32
			D4s v3	21
			D8s v3	10
			D16s v3	5
			D32s v3	2
			D48s v3	1
			D64s v3	1

You can also mix multiple VM sizes on the Dsv3-Type3. The following are sample combinations of VM packings on the Dsv3-Type3:

- 1 D64sv3 + 1 D8sv3 + 2 D4sv3
- 1 D48sv3 + 1 D16sv3 + 4 D4sv3
- 15 D4sv3 + 10 D2sv3

Dsv3-Type4

The Dsv3-Type4 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Dsv3-

Type4 runs Dsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Dsv3-Type4 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	D2s v3	32
			D4s v3	24
			D8s v3	12
			D16s v3	6
			D32s v3	3
			D48s v3	2
			D64s v3	1

Next steps

- For more information, see the Dedicated hosts overview.
- There is sample template, available at Azure Quickstart Templates that uses both zones and fault domains for maximum resiliency in a region.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Compute Optimized Azure Dedicated Host SKUs

Article • 08/22/2024

Azure Dedicated Host SKUs are the combination of a VM family and a certain hardware specification. You can only deploy VMs of the VM series that the Dedicated Host SKU specifies. For example, on the Dsv3-Type3, you can only provision Dsv3-series VMs.

This document goes through the hardware specifications and VM packings for all compute optimized Dedicated Host SKUs.

Limitations

The sizes and hardware types available for dedicated hosts vary by region. Refer to the host pricing page to learn more.

Fsv2

Fsv2-Type2

The Fsv2-Type2 is a Dedicated Host SKU utilizing the Intel® Skylake (Xeon® Platinum 8168) processor. It offers 48 physical cores, 72 vCPUs, and 144 GiB of RAM. The Fsv2-Type2 runs Fsv2-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Fsv2-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
48	72	144 GiB	F2s v2	32
			F4s v2	18
			F8s v2	9
			F16s v2	4
			F32s v2	2
			F48s v2	1

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			F64s v2	1
			F72s v2	1

Fsv2-Type3

The Fsv2-Type3 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Fsv2-Type3 runs Fsv2-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Fsv2-Type3 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	F2s v2	32
			F4s v2	21
			F8s v2	10
			F16s v2	5
			F32s v2	2
			F48s v2	1
			F64s v2	1
			F72s v2	1

Fsv2-Type4

The Fsv2-Type4 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Fsv2-Type4 runs Fsv2-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Fsv2-Type4 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	F2s v2	32
			F4s v2	24
			F8s v2	12
			F16s v2	6
			F32s v2	3
			F48s v2	2
			F64s v2	1
			F72s v2	1

FXmds

FXmds-Type1

The FXmds-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Gold 6246R) processor. It offers 32 physical cores, 48 vCPUs, and 1,152 GiB of RAM. The FXmds-Type1 runs FX-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a FXmds-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
32	48	1,152 GiB	FX4mds	12
			FX12mds	4
			FX24mds	2
			FX36mds	1
			FX48mds	1

Next steps

- For more information, see the Dedicated hosts overview.

- There is sample template, available at Azure quickstart templates, that uses both zones and fault domains for maximum resiliency in a region.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Memory Optimized Azure Dedicated Host SKUs

Article • 08/22/2024

Azure Dedicated Host SKUs are the combination of a VM family and a certain hardware specification. You can only deploy VMs of the VM series that the Dedicated Host SKU specifies. For example, on the Dsv3-Type3, you can only provision Dsv3-series VMs.

This document goes through the hardware specifications and VM packings for all memory optimized Dedicated Host SKUs.

Limitations

The sizes and hardware types available for dedicated hosts vary by region. Refer to the host pricing page to learn more.

Eadsv5

Eadsv5-Type1

The Eadsv5-Type1 is a Dedicated Host SKU utilizing AMD's EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The Eadsv5-Type1 runs Eadsv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Eadsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	E2ads v5	32
			E4ads v5	21
			E8ads v5	10
			E16ads v5	5
			E20ads v5	4
			E32ads v5	2

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			E48ads v5	1
			E64ads v5	1
			E96ads v5	1

Easv5

Easv5-Type1

The Easv5-Type1 is a Dedicated Host SKU utilizing AMD's EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The Easv5-Type1 runs Easv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Easv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	E2as v5	32
			E4as v5	21
			E8as v5	10
			E16as v5	5
			E20as v5	4
			E32as v5	2
			E48as v5	1
			E64as v5	1
			E96as v5	1

Easv4

Easv4-Type1

The Easv4-Type1 is a Dedicated Host SKU utilizing AMD's 2.35 GHz EPYC™ 7452 processor. It offers 64 physical cores, 96 vCPUs, and 672 GiB of RAM. The Easv4-Type1 runs Easv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Easv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	96	672 GiB	E2as v4	32
			E4as v4	21
			E8as v4	10
			E16as v4	5
			E20as v4	4
			E32as v4	2
			E48as v4	1
			E64as v4	1
			E96as v4	1

Easv4-Type2

The Easv4-Type2 is a Dedicated Host SKU utilizing AMD's EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The Easv4-Type2 runs Easv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Easv4-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	E2as v4	32
			E4as v4	21
			E8as v4	10

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			E16as v4	5
			E20as v4	4
			E32as v4	2
			E48as v4	1
			E64as v4	1
			E96as v4	1

Ebdsv5

Ebdsv5-Type1

The Ebdsv5-Type1 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Ebdsv5-Type1 runs Ebdsv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Ebdsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2bds v5	8
			E4bds v5	8
			E8bds v5	6
			E16bds v5	3
			E32bds v5	1
			E48bds v5	1
			E64bds v5	1

Ebsv5

Ebsv5-Type1

The Ebsv5-Type1 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Ebsv5-Type1 runs Ebsv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Ebsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2bs v5	8
			E4bs v5	8
			E8bs v5	6
			E16bs v5	3
			E32bs v5	1
			E48bs v5	1
			E64bs v5	1

ECadsv5

ECadsv5-Type1

The ECadsv5-Type1 is a Dedicated Host SKU utilizing the AMD 3rd Generation EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The ECadsv5-Type1 runs ECadsv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an ECadsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	EC2ads v5	32
			EC4ads v5	21

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			EC8ads v5	10
			EC16ads v5	5
			EC20ads v5	4
			EC32ads v5	3
			EC48ads v5	1
			EC64ads v5	1
			EC96ads v5	1

ECasv5

ECasv5-Type1

The ECasv5-Type1 is a Dedicated Host SKU utilizing the AMD 3rd Generation EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 768 GiB of RAM. The ECasv5-Type1 runs ECasv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an ECasv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	768 GiB	EC2as v5	32
			EC4as v5	21
			EC8as v5	10
			EC16as v5	5
			EC20as v5	4
			EC32as v5	3
			EC48as v5	1
			EC64as v5	1
			EC96as v5	1

Edsv5

Edsv5-Type1

The Edsv5-Type1 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Edsv5-Type1 runs Edsv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Edsv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2ds v5	32
			E4ds v5	21
			E8ds v5	10
			E16ds v5	5
			E20ds v5	4
			E32ds v5	2
			E48ds v5	1
			E64ds v5	1

Edsv4

Edsv4-Type1

The Edsv4-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Edsv4-Type1 runs Edsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Edsv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	E2ds v4	31
			E4ds v4	15
			E8ds v4	7
			E16ds v4	3
			E20ds v4	3
			E32ds v4	1
			E48ds v4	1
			E64ds v4	1

Edsv4-Type2

The Edsv4-Type2 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Edsv4-Type2 runs Edsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Edsv4-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2ds v4	32
			E4ds v4	19
			E8ds v4	9
			E16ds v4	4
			E20ds v4	3
			E32ds v4	2
			E48ds v4	1
			E64ds v4	1

Esv5

Esv5-Type1

The Esv5-Type1 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Esv5-Type1 runs Esv5-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv5-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2s v5	32
			E4s v5	21
			E8s v5	10
			E16s v5	5
			E20s v5	4
			E32s v5	2
			E48s v5	1
			E64s v5	1

Esv4

Esv4-Type1

The Esv4-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Esv4-Type1 runs Esv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	E2s v4	31

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			E4s v4	15
			E8s v4	7
			E16s v4	3
			E20s v4	3
			E32s v4	1
			E48s v4	1
			E64s v4	1

Esv4-Type2

The Esv4-Type2 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Esv4-Type2 runs Esv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv4-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2s v4	32
			E4s v4	21
			E8s v4	10
			E16s v4	5
			E20s v4	4
			E32s v4	2
			E48s v4	1
			E64s v4	1

Esv3

Esv3-Type1

> **Note**
>
> **The Esv3-Type1 will be retired on June 30, 2023**. Refer to the dedicated host retirement guide to learn more.

The Esv3-Type1 is a Dedicated Host SKU utilizing the Intel® Broadwell (2.3 GHz Xeon® E5-2673 v4) processor. It offers 40 physical cores, 64 vCPUs, and 448 GiB of RAM. The Esv3-Type1 runs Esv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
40	64	448 GiB	E2s v3	28
			E4s v3	14
			E8s v3	7
			E16s v3	3
			E20s v3	2
			E32s v3	1
			E48s v3	1
			E64s v3	1

Esv3-Type2

> **Note**
>
> **The Esv3-Type2 will be retired on June 30, 2023**. Refer to the dedicated host retirement guide to learn more.

The Esv3-Type2 is a Dedicated Host SKU utilizing the Intel® Skylake (Xeon® 8171M) processor. It offers 48 physical cores, 76 vCPUs, and 504 GiB of RAM. The Esv3-Type2 runs Esv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv3-Type2 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
48	76	504 GiB	E2s v3	31
			E4s v3	15
			E8s v3	7
			E16s v3	3
			E20s v3	3
			E32s v3	1
			E48s v3	1
			E64s v3	1

Esv3-Type3

The Esv3-Type3 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8272CL) processor. It offers 52 physical cores, 80 vCPUs, and 504 GiB of RAM. The Esv3-Type3 runs Esv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv3-Type3 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
52	80	504 GiB	E2s v3	31
			E4s v3	15
			E8s v3	7
			E16s v3	3
			E20s v3	3
			E32s v3	1
			E48s v3	1

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			E64s v3	1

Esv3-Type4

The Esv3-Type4 is a Dedicated Host SKU utilizing the Intel® Ice Lake (Xeon® Platinum 8370C) processor. It offers 64 physical cores, 119 vCPUs, and 768 GiB of RAM. The Esv3-Type4 runs Esv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto an Esv3-Type4 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	768 GiB	E2s v3	32
			E4s v3	21
			E8s v3	10
			E16s v3	5
			E20s v3	4
			E32s v3	2
			E48s v3	1
			E64s v3	1

M

Ms-Type1

The Ms-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8280) processor. It offers 112 physical cores, 128 vCPUs, and 2,048 GiB of RAM. The Ms-Type1 runs M-series VMs, including M, Mls, Ms, and Mts VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Ms-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
112	128	2,048 GiB	M32ls	4
			M32ts	4
			M64	2
			M64s	2
			M64ls	2
			M128	1
			M128s	1

Msm-Type1

The Msm-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8280) processor. It offers 112 physical cores, 128 vCPUs, and 3,892 GiB of RAM. The Msm-Type1 runs M-series VMs, including Ms, Mms, Mts, and Mls VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Msm-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
112	128	3,892 GiB	M8ms	16
			M8-2ms	16
			M8-4ms	16
			M16ms	8
			M16-4ms	8
			M16-8ms	8
			M32ts	4
			M32ls	4
			M32ms	4
			M32-8ms	4

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			M32-16ms	4
			M64ms	2
			M64	2
			M64s	2
			M64m	2
			M64ls	2
			M64-16ms	2
			M64-32ms	2
			M128ms	1
			M128s	1
			M128m	1
			M128	1
			M128-32ms	1
			M128-64ms	1

Mdsv2

Mdmsv2MedMem-Type1

The Mdmsv2MedMem-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8280) processor. It offers 112 physical cores, 192 vCPUs, and 4,096 GiB of RAM. The Mdmsv2MedMem-Type1 runs Msv2-series VMs, including Mdsv2 and Mdmsv2 VMs.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
112	192	4,096 GiB	M32dms v2	4
			M64ds v2	2
			M64dms v2	2

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			M128ds v2	1
			M128dms v2	1

Mdsv2MedMem-Type1

The Mdsv2MedMem-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8280) processor. It offers 112 physical cores, 192 vCPUs, and 2,048 GiB of RAM. The Mdsv2MedMem-Type1 runs Msv2-series VMs, including Mdsv2 and Mdmsv2 VMs.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
112	192	2,048 GiB	M32dms v2	2
			M64ds v2	2
			M64dms v2	1
			M128ds v2	1

Msv2

Mmsv2MedMem-Type1

The Mmsv2MedMem-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8280) processor. It offers 112 physical cores, 192 vCPUs, and 4,096 GiB of RAM. The Mmsv2MedMem-Type1 runs Msv2-series VMs, including Msv2 and Mmsv2 VMs.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
112	192	4,096 GiB	M32ms v2	4
			M64s v2	3
			M64ms v2	2

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			M128ms v2	1
			M128s v2	1

Msv2MedMem-Type1

The Msv2MedMem-Type1 is a Dedicated Host SKU utilizing the Intel® Cascade Lake (Xeon® Platinum 8280) processor. It offers 112 physical cores, 192 vCPUs, and 2,048 GiB of RAM. The Msv2MedMem-Type1 runs Msv2-series VMs, including Msv2 and Mmsv2 VMs.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
112	192	2,048 GiB	M32ms v2	2
			M64s v2	2
			M64ms v2	1
			M128s v2	1

Mv2

Msmv2-Type1

The Msm-Type1 is a Dedicated Host SKU utilizing the Intel® Skylake (Xeon® Platinum 8180M) processor. It offers 224 physical cores, 416 vCPUs, and 11,400 GiB of RAM. The Msmv2-Type1 runs Mv2-series VMs, including Msv2 and Mmsv2 VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Msm-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
224	416	11,400 GiB	M208ms v2	2
			M208s v2	2

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			M416-208ms v2	1
			M416-208s v2	1
			M416ms v2	1
			M416s-8 v2	1
			M416s v2	1

Msv2-Type1

The Msv2-Type1 is a Dedicated Host SKU utilizing the Intel® Skylake (Xeon® Platinum 8180M) processor. It offers 224 physical cores, 416 vCPUs, and 5,700 GiB of RAM. The Msv2-Type1 runs Mv2-series VMs, including Msv2 and Mmsv2 VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Msv2-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
224	416	5,700 GiB	M208ms v2	2
			M208s v2	1
			M416-208s v2	1
			M416s v2	1

Next steps

- For more information, see the Dedicated hosts overview.
- There's sample template, available at Azure Quickstart Templates, which uses both zones and fault domains for maximum resiliency in a region.

Feedback

Was this page helpful? Yes No

Storage Optimized Azure Dedicated Host SKUs

Article • 08/22/2024

Azure Dedicated Host SKUs are the combination of a VM family and a certain hardware specification. You can only deploy VMs of the VM series that the Dedicated Host SKU specifies. For example, on the Dsv3-Type3, you can only provision Dsv3-series VMs.

This document goes through the hardware specifications and VM packings for all storage optimized Dedicated Host SKUs.

Limitations

The sizes and hardware types available for dedicated hosts vary by region. Refer to the host pricing page to learn more.

Lasv3

Lasv3-Type1

The Lasv3-Type1 is a Dedicated Host SKU utilizing the AMD 3rd Generation EPYC™ 7763v processor. It offers 64 physical cores, 112 vCPUs, and 1024 GiB of RAM. The Lasv3-Type1 runs Lasv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Lasv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	112	1024 GiB	L8as v3	10
			L16as v3	5
			L32as v3	2
			L48as v3	1
			L64as v3	1
			L80as v3	1

Lsv3

Lsv3-Type1

The Lsv3-Type1 is a Dedicated Host SKU utilizing the Intel® 3rd Generation Xeon® Platinum 8370C (Ice Lake) processor. It offers 64 physical cores, 119 vCPUs, and 1024 GiB of RAM. The Lsv3-Type1 runs Lsv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Lsv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	119	1024 GiB	L8s v3	10
			L16s v3	5
			L32s v3	2
			L48s v3	1
			L64s v3	1
			L80s v3	1

Lsv2

Lsv2-Type1

The Lsv2-Type1 is a Dedicated Host SKU utilizing the AMD 2.55 GHz EPYC™ 7551 processor. It offers 64 physical cores, 80 vCPUs, and 640 GiB of RAM. The Lsv2-Type1 runs Lsv2-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a Lsv2-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
64	80	640 GiB	L8s v2	10

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
			L16s v2	5
			L32s v2	2
			L48s v2	1
			L64s v2	1
			L80s v2	1

Next steps

- For more information, see the Dedicated hosts overview.
- There's a sample template, available at Azure Quickstart Templates that uses both zones and fault domains for maximum resiliency in a region.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

GPU Optimized Azure Dedicated Host SKUs

Article • 08/22/2024

Azure Dedicated Host SKUs are the combination of a VM family and a certain hardware specification. You can only deploy VMs of the VM series that the Dedicated Host SKU specifies. For example, on the Dsv3-Type3, you can only provision Dsv3-series VMs.

This document goes through the hardware specifications and VM packings for all GPU optimized Dedicated Host SKUs.

Limitations

The sizes and hardware types available for dedicated hosts vary by region. Refer to the host pricing page to learn more.

NVasv4

NVasv4-Type1

The NVasv4-Type1 is a Dedicated Host SKU utilizing the AMD® Rome (EPYC™ 7V12) processor with AMD Radeon Instinct MI25 GPUs. It offers 128 physical cores, 128 vCPUs, and 448 GiB of RAM. The NVasv4-Type1 runs NVsv4-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a NVasv4-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
128	128	448 GiB	NV4as v4	30
			NV8as v4	15
			NV16as v4	7
			NV32as v4	3

NVsv3

NVsv3-Type1

The NVsv3-Type1 is a Dedicated Host SKU utilizing the Intel® Broadwell (E5-2690 v4) processor with NVIDIA Tesla M60 GPUs and NVIDIA GRID technology. It offers 28 physical cores, 48 vCPUs, and 448 GiB of RAM. The NVsv3-Type1 runs NVv3-series VMs.

The following packing configuration outlines the max packing of uniform VMs you can put onto a NVsv3-Type1 host.

Expand table

Physical cores	Available vCPUs	Available RAM	VM Size	# VMs
28	48	448 GiB	NV12s v3	4
			NV24s v3	2
			NV48s v3	1

Next steps

- For more information, see the Dedicated hosts overview.
- There is sample template, available at Azure quickstart templates, that uses both zones and fault domains for maximum resiliency in a region.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Dedicated Host SKU Retirement

03/17/2025

We continue to modernize and optimize Azure Dedicated Host by using the latest innovations in processor and datacenter technologies. Azure Dedicated Host is a combination of a virtual machine (VM) series and a specific Intel or AMD-based physical server. As we innovate and work with our technology partners, we also need to plan how we retire aging technology.

UPDATE: Retirement timeline extension

Considering the feedback from several Azure Dedicated Host customers that are running their critical workloads on SKUs that are scheduled for retirement, we have extended the retirement timeline from March 31, 2023 to June 30, 2023. We don't intend to move the retirement timeline any further and recommend all ADH users that are using any of the listed SKUs to migrate to newer generation based SKUs to avoid workload disruptions.

Migrations required by 30 June 2023 [Updated]

All hardware has a finite lifespan, including the underlying hardware for Azure Dedicated Host. As we continue to modernize Azure datacenters, hardware is decommissioned and eventually retired. The hardware that runs the following Dedicated Host SKUs is reaching end of life:

- Dsv3-Type1
- Dsv3-Type2
- Esv3-Type1
- Esv3-Type2

As a result we'll retire these Dedicated Host SKUs on 30 June 2023.

How does the retirement of Azure Dedicated Host SKUs affect you?

The current retirement impacts the following Azure Dedicated Host SKUs:

- Dsv3-Type1
- Esv3-Type1
- Dsv3-Type2
- Esv3-Type2

Note: If you're running a Dsv3-Type3, Dsv3-Type4, an Esv3-Type3, or an Esv3-Type4 Dedicated Host, you are not impacted.

What actions should you take?

For manually placed VMs, you need to create a Dedicated Host of a newer SKU, stop the VMs on your existing Dedicated Host, reassign them to the new host, start the VMs, and delete the old host. For automatically placed VMs or for Virtual Machine Scale Sets, you need to create a Dedicated Host of a newer SKU, stop the VMs or Virtual Machine Scale Set, delete the old host, and then start the VMs or Virtual Machine Scale Set.

Refer to the Azure Dedicated Host Migration Guide for more detailed instructions. We recommend moving to the latest generation of Dedicated Host for your VM family.

If you have any questions, contact us through customer support.

FAQs

Q: Will migration result in downtime?

A: Yes, you would have to stop/deallocate your VMs or Virtual Machine Scale Sets before moving them to the target host.

Q: When will the other Dedicated Host SKUs retire?

A: We'll announce Dedicated Host SKU retirements 12 months in advance of the official retirement date of a given Dedicated Host SKU.

Q: What are the milestones for the Dsv3-Type1, Dsv3-Type2, Esv3-Type1, and Esv3-Type1 retirement?

A:

Expand table

Date	Action
15 March 2022	Dsv3-Type1, Dsv3-Type2, Esv3-Type1, Esv3-Type2 retirement announcement
30 June 2023	Dsv3-Type1, Dsv3-Type2, Esv3-Type1, Esv3-Type2 retirement

Q: What happens to my Azure Reservation?

A: You need to exchange your reservation through the Azure portal to match the new Dedicated Host SKU.

Q: What would happen to my host if I do not migrate by June 30, 2023?

A: After June 30, 2023 any dedicated host running on the SKUs that are marked for retirement will be set to 'Host Pending Deallocate' state before eventually deallocating the host. For more assistance, please reach out to Azure support.

Q: What will happen to my VMs if a Host is automatically deallocated?

A: If the underlying host is deallocated the VMs that were running on the host would be either in failed or stopped state but not deleted.

You could stop/deallocate the VM or virtual machine scale set and dissociate the VM or virtual machine scale set from the host/host group (depending on explicit vs implicit placement) and then you would be able to either create a new host (of the same VM family) and allocate VMs on the host/host group or run the VMs on multitenant infrastructure.

Refer to the Azure Dedicated Host Migration Guide for more detailed instructions.

Azure Dedicated Host SKU Retirement Migration Guide

As hardware ages, it must be retired and workloads must be migrated to newer, faster, and more efficient Azure Dedicated Host SKUs. The legacy Dedicated Host SKUs should be migrated to newer Dedicated Host SKUs. The main differences between the retiring Dedicated Host SKUs and the newly recommended Dedicated Host SKUs are:

- Newer, more efficient processors
- Increased RAM
- Increased available vCPUs
- Greater regional capacity compared to the retiring Dedicated Host SKUs

Review the FAQs before you get started on migration. The next section will go over which Dedicated Host SKUs to migrate to help aid in migration planning and execution.

Host SKUs being retired

Some Azure Dedicated Host SKUs will be retired soon. Refer to the Azure Dedicated Host SKU Retirement documentation to learn more.

Dsv3-Type1 and Dsv3-Type2

The Dsv3-Type1 and Dsv3-Type2 run Dsv3-series VMs, which offer a combination of vCPU, memory, and temporary storage best suited for most general-purpose workloads. We recommend migrating your existing VMs to one of the following Dedicated Host SKUs:

- Dsv3-Type3
- Dsv3-Type4

Note that both the Dsv3-Type3 and Dsv3-Type4 won't be impacted by the 31 March 2023 retirement date. We recommend moving to either the Dsv3-Type3 or Dsv3-Type4 based on regional availability, pricing, and your organization’s needs.

Esv3-Type1 and Esv3-Type2

The Esv3-Type1 and Esv3-Type2 run Esv3-series VMs, which offer a combination of vCPU, memory, and temporary storage best suited for most memory-intensive workloads. We recommend migrating your existing VMs to one of the following Dedicated Host SKUs:

- Esv3-Type3

- Esv3-Type4

Note that both the Esv3-Type3 and Esv3-Type4 won't be impacted by the 31 March 2023 retirement date. We recommend moving to either the Esv3-Type3 or Esv3-Type4 based on regional availability, pricing, and your organization's needs.

Migrating to supported hosts

To migrate your workloads and avoid Dedicated Host SKU retirement, follow the directions for your migration method of choice.

Automatic migration (Resize)

Moving a host and all associated VMs to newer generation hardware can be done through the host resize feature. Resize simplifies the migration process and avoids having to manually create new hosts and move all VMs individually.

Resize limitations:

- Host can only be resized to an ADH within the same VM family. A Dsv3-Type3 host can be resized to Dsv3-Type4 but **not** to an **E**sv3-Type4.
- You can only resize to newer generation of hardware. A Dsv3-Type3 host can be resized to Dsv3-Type4 but **not** Dsv3-Type2.
- Resizing changes the 'Host Asset ID'. The 'Host ID' remains the same.
- The host and all associated VMs become unavailable during the resize operation.

> ⚠ **Warning**
>
> The resize operation causes the loss of any non-persistent data such as temp disk data. Save all your work to persistent data storage before triggering resize.

> ⓘ **Note**
>
> If the source host is already running on the latest hardware, 'Size' page would display an empty list. If you're looking for enhanced performance, consider switching to a different VM family.

Portal

1. Search for and select the host.

2. In the left menu under **Settings**, select **Size**.
3. Once on the size page from the list of SKUs, select the desired SKU to resize to.
4. Selecting a target size from the list would enable **Resize** button on the bottom on the page.
5. Click **Resize**, host's 'Provisioning State' changes from 'Provisioning Succeeded' to 'Updating'
6. Once the resizing is complete, the host's 'Provisioning State' reverts to 'Provisioning Succeeded'

Manual migration

This includes steps for manually placed VMs, automatically placed VMs, and virtual machine scale sets on your Dedicated Hosts:

Manually Placed VMs

1. Choose a target Dedicated Host SKU to migrate to.
2. Ensure you have quota for the VM family associated with the target Dedicated Host SKU in your given region.
3. Provision a new Dedicated Host of the target Dedicated Host SKU in the same Host Group.
4. Stop and deallocate the VM(s) on your old Dedicated Host.
5. Reassign the VM(s) to the target Dedicated Host.
6. Start the VM(s).
7. Delete the old host.

More detailed instructions can be found in the following sections.

> ⓘ **Note**
>
> **Certain sections are different for automatically placed VMs or virtual machine scale set**. These differences will explicitly be called out in the respective steps.

Ensure quota for the target VM family

Be sure that you have enough vCPU quota for the VM family of the Dedicated Host SKU that you'll be using. If you need quota, follow this guide to request an increase in vCPU quota for

your target VM family in your target region. Select the Dsv3-series or Esv3-series as the VM family, depending on the target Dedicated Host SKU.

Create a new Dedicated Host

Within the same Host Group as the existing Dedicated Host, create a Dedicated Host of the target Dedicated Host SKU.

Stop the VM(s) or virtual machine scale set

PowerShell

Refer to the PowerShell documentation to stop a VM through PowerShell or stop a virtual machine scale set through PowerShell.

Reassign the VM(s) to the target Dedicated Host

> ⓘ Note
>
> **Skip this step for automatically placed VMs and virtual machine scale set.**

Once the target Dedicated Host has been created and the VM has been stopped, reassign the VM to the target Dedicated Host.

Start the VM(s) or virtual machine scale set

> ⓘ Note
>
> **Automatically placed VM(s) and virtual machine scale set require that you delete the old host *before* starting the autoplaced VM(s) or virtual machine scale set.**

PowerShell

Refer to the PowerShell documentation to start a VM through PowerShell or start a virtual machine scale set through PowerShell.

Delete the old Dedicated Host

Once all VMs have been migrated from your old Dedicated Host to the target Dedicated Host, delete the old Dedicated Host.

Help and support

If you have questions, ask community experts in Microsoft Q&A.

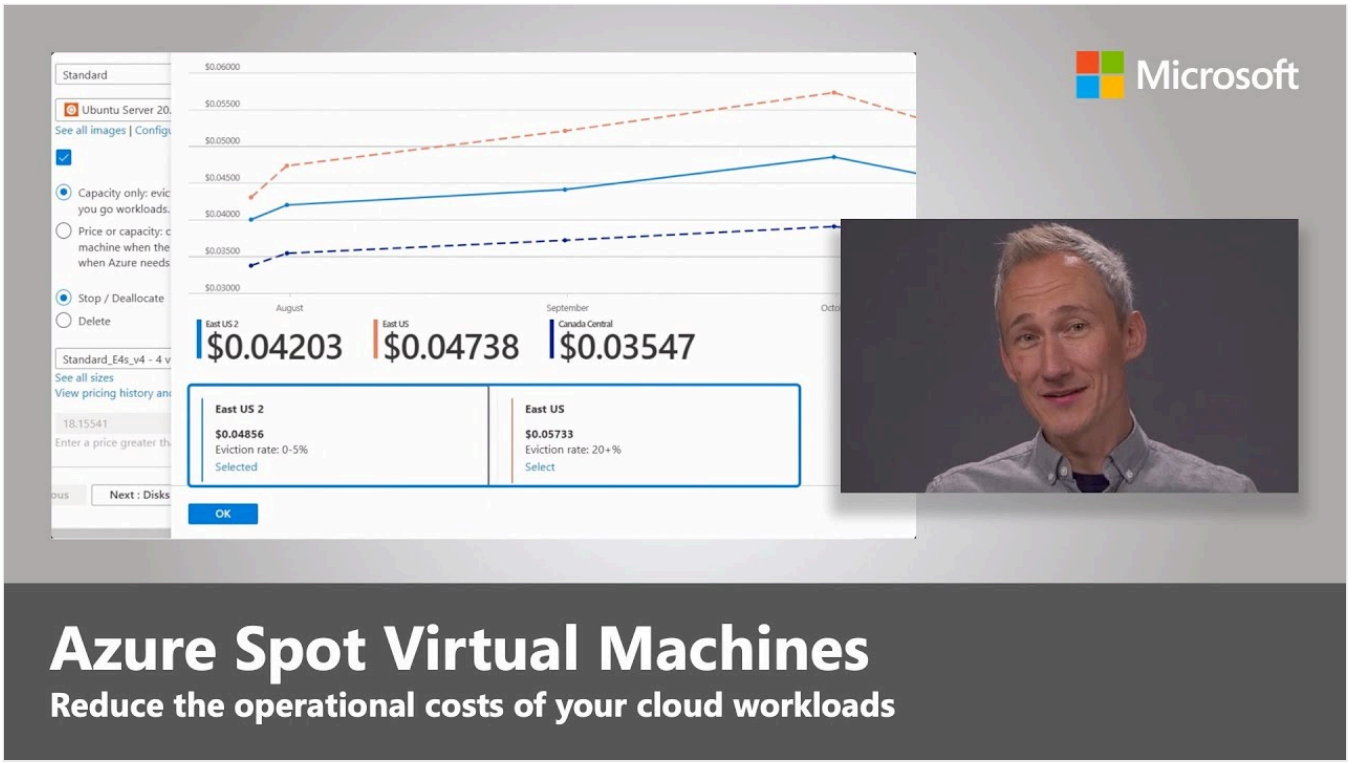
Last updated on 03/17/2025

Azure Spot Virtual Machines

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Using Azure Spot Virtual Machines allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Azure Spot Virtual Machines. Therefore, Azure Spot Virtual Machines are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

The amount of available capacity can vary based on size, region, time of day, and more. When deploying Azure Spot Virtual Machines, Azure will allocate the VMs if there's capacity available, but there's no SLA for these VMs. An Azure Spot Virtual Machine offers no high availability guarantees. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Azure Spot Virtual Machines with 30-seconds notice.



Eviction policy

Spot VMs can be stopped if Azure needs capacity for other pay-as-you-go workloads or when the price of the spot instance exceeds the maximum price that you have set. When creating an Azure Spot Virtual Machine, you can set the eviction policy to *Deallocate* (default) or *Delete*.

The *Deallocate* policy moves your VM to the stopped-deallocated state, allowing you to redeploy it later. However, there's no guarantee that the allocation will succeed. The deallocated VMs will count against your quota and you'll be charged storage costs for the underlying disks.

If you would like your VM to be deleted when it's evicted, you can set the eviction policy to *delete*. The evicted VMs are deleted together with their underlying disks, so you'll not continue to be charged for the storage.

You can opt in to receive in-VM notifications through Azure Scheduled Events. These are delivered on a best effort basis up to 30 seconds prior to the eviction.

Expand table

Option	Outcome
Max price is set to >= the current price.	VM is deployed if capacity and quota are available.
Max price is set to < the current price.	The VM isn't deployed. You'll get an error message that the max price needs to be >= current price.
Restarting a stopped/deallocated VM if the max price is >= the current price	If there's capacity and quota, then the VM is deployed.
Restarting a stopped/deallocated VM if the max price is < the current price	You'll get an error message that the max price needs to be >= current price.
Price for the VM has gone up and is now > the max price.	The VM gets evicted. Azure will attempt scheduled event delivery up to 30 seconds before actual eviction.
After eviction, the price for the VM goes back to being < the max price.	The VM won't be automatically restarted. You can restart the VM yourself, and it will be charged at the current price.
If the max price is set to `-1`	The VM won't be evicted for pricing reasons. The max price will be the current price, up to the price for standard VMs. You'll never be charged above the standard price.
Changing the max price	You need to deallocate the VM to change the max price. Deallocate the VM, set a new max price, then update the VM.

Tip

Check out our Azure Virtual Machine Spot Eviction guide to learn how to create a reliable interruptible workload in Azure.

Limitations

The following VM sizes aren't supported for Azure Spot Virtual Machines:

- B-series

- Promo versions of any size (like Dv2, NV, NC, H promo sizes)

Azure Spot Virtual Machines can be deployed to any region, except Microsoft Azure operated by 21Vianet.

The following offer types are currently supported:

- Enterprise Agreement
- Pay-as-you-go offer code (003P)
- Sponsored (0036P and 0136P) - not available in Fairfax
- For Cloud Service Provider (CSP), see the Partner Center or contact your partner directly.

Pricing

Pricing for Azure Spot Virtual Machines is variable, based on region and SKU. For more information, see VM pricing for Linux and Windows.

You can also query pricing information using the Azure retail prices API to query for information about Spot pricing. The `meterName` and `skuName` will both contain `Spot`.

With variable pricing, you have option to set a max price, in US dollars (USD), using up to five decimal places. For example, the value `0.98765` would be a max price of $0.98765 USD per hour. If you set the max price to be `-1`, the VM won't be evicted based on price. The price for the VM will be the current price for spot or the price for a standard VM, which ever is less, as long as there's capacity and quota available.

Pricing and eviction history

Portal

You can see historical pricing and eviction rates per size in a region in the portal while you are creating the VM. After selecting the checkbox to **Run with Azure Spot discount**, a link will appear under the size selection of the VM titled **View pricing history and compare prices in nearby regions**. By selecting that link you will be able to see a table or graph of spot pricing for the specified VM size. The pricing and eviction rates in the following images are only examples.

> **Tip**

Eviction rates are quoted *per hour*. For example, an eviction rate of 10% means a VM has a 10% chance of being evicted within the next hour, based on historical eviction data of the last 7 days.

Chart:

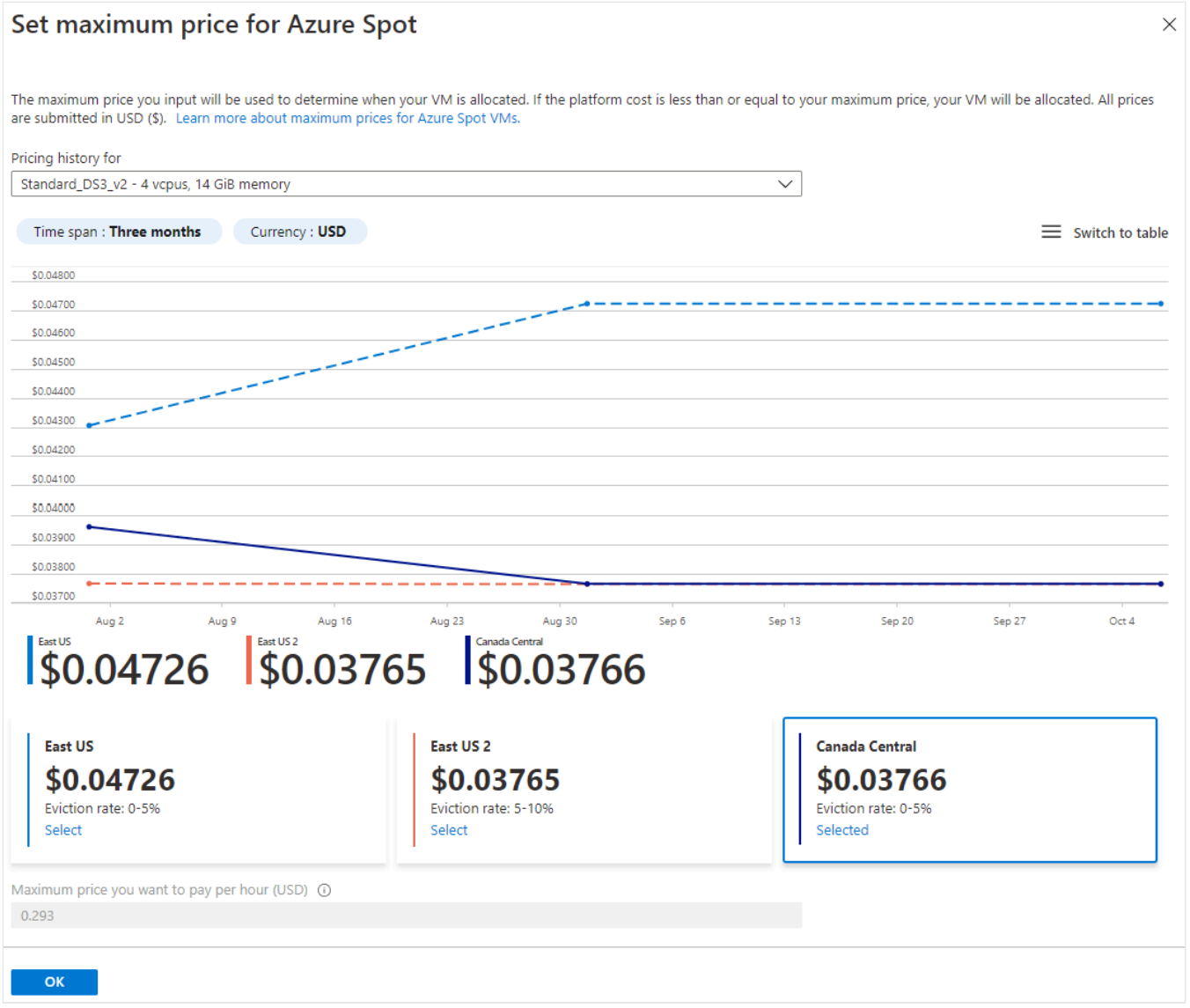
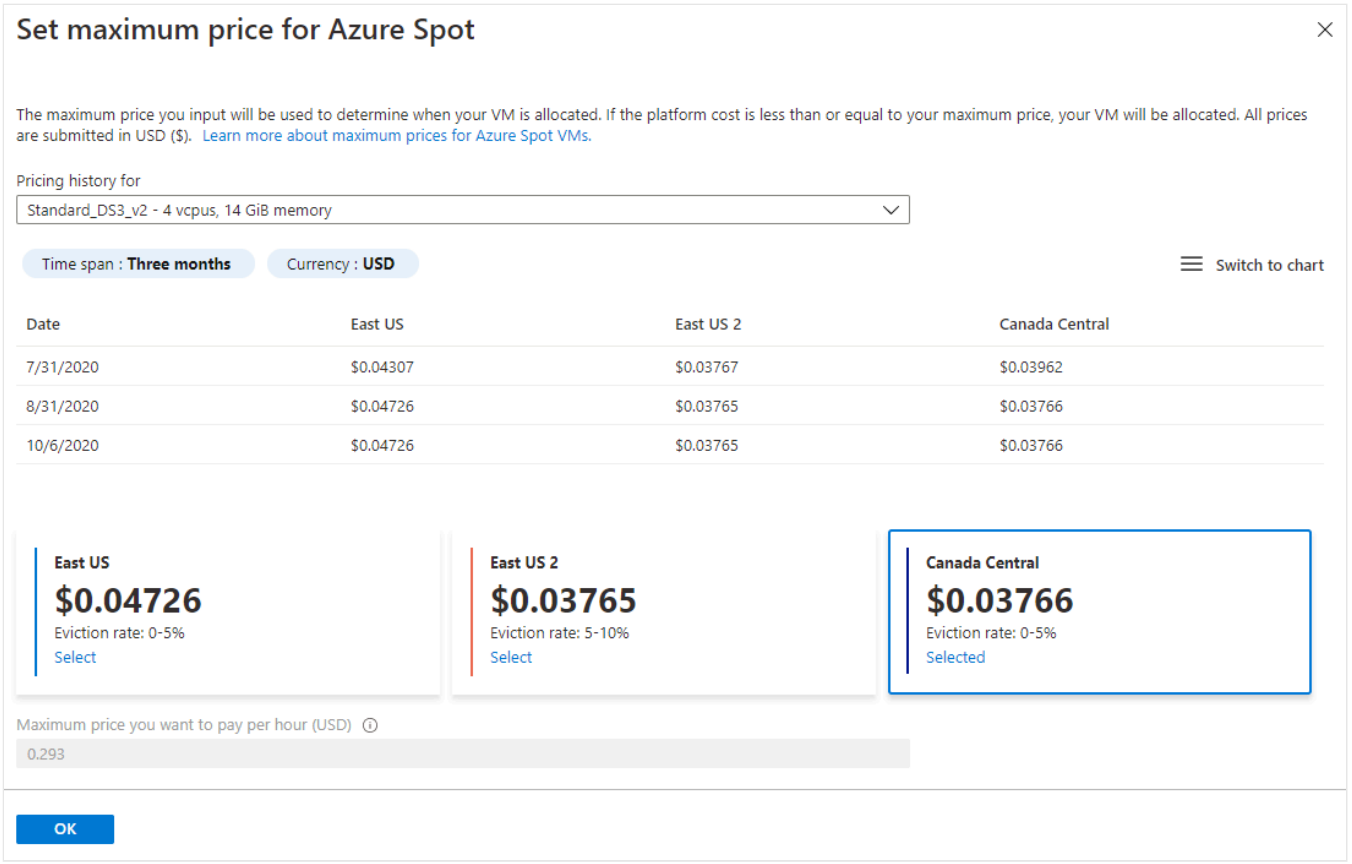


Table:



Azure Resource Graph

You can programmatically access relevant Spot VM SKU data through Azure Resource Graph. Get pricing history in the last 90 days and eviction rates for the last 28 trailing days to identify SKUs that better meet your specific needs.

Key benefits:

- Query Spot eviction rates and the last few months of Spot prices programmatically through ARM or the ARG Explorer in Azure portal
- Create a custom query to extract the specific data relevant to your scenario with the ability to filter across a variety of parameters, such as SKU and region
- Easily compare data across multiple regions and SKUs
- Find a different Spot SKU or region with a lower price and/or eviction rate

Try out the following sample queries for Spot pricing history and eviction rates using the ARG Explorer in Azure portal. Spot pricing history and eviction rates data are available in the `SpotResources` table.

Spot pricing history sample query:

SQL

```
SpotResources
| where type =~ 'microsoft.compute/skuspotpricehistory/ostype/location'
```

```
| where sku.name in~ ('standard_d2s_v4', 'standard_d4s_v4')
| where properties.osType =~ 'linux'
| where location in~ ('eastus', 'southcentralus')
|
 project skuName = tostring(sku.name), osType = tostring(properties.osType), locati
on, latestSpotPriceUSD = todouble(properties.spotPrices[0].priceUSD)
| order by latestSpotPriceUSD asc
```

Spot eviction rates sample query:

SQL

```
SpotResources
| where type =~ 'microsoft.compute/skuspotevictionrate/location'
| where sku.name in~ ('standard_d2s_v4', 'standard_d4s_v4')
| where location in~ ('eastus', 'southcentralus')
|
 project skuName = tostring(sku.name), location, spotEvictionRate = tostring(proper
ties.evictionRate)
| order by skuName asc, location asc
```

Alternatively, try out the ARG REST API to get the pricing history and eviction rate history data.

Frequently asked questions

Q: Once created, is an Azure Spot Virtual Machine the same as regular standard VM?

A: Yes, except there's no SLA for Azure Spot Virtual Machines and they can be evicted at any time.

Q: What to do when you get evicted, but still need capacity?

A: We recommend you use standard VMs instead of Azure Spot Virtual Machines if you need capacity right away.

Q: How is quota managed for Azure Spot Virtual Machines?

A: Azure Spot Virtual Machines will have a separate quota pool. Spot quota will be shared between VMs and scale-set instances. For more information, see Azure subscription and service limits, quotas, and constraints.

Q: Can I request for additional quota for Azure Spot Virtual Machines?

A: Yes, you'll be able to submit the request to increase your quota for Azure Spot Virtual Machines through the standard quota request process.

Q: Where can I post questions?

A: You can post and tag your question with `azure-spot` at Q&A.

Q: How can I change the max price for a spot VM?

A: Before you can change the max price, you need to deallocate the VM. Then you can change the max price in the portal, from the **Configuration** section for the VM.

Q: Can I convert existing VMs to Spot VMs?

A: No, setting the Spot flag is only supported at creation time.

Q: Can I convert Spot VMs to Standard VMs?

A: No, Spot VMs cannot be directly converted to Standard VMs.

Next steps

Use the CLI, portal, ARM template, or PowerShell to deploy Azure Spot Virtual Machines.

You can also deploy a scale set with Azure Spot Virtual Machine instances.

If you encounter an error, see Error codes.

Last updated on 02/06/2026

Deploy Azure Spot Virtual Machines using the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Using Azure Spot Virtual Machines allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure evicts Azure Spot Virtual Machines. Therefore, Azure Spot Virtual Machines are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Azure Spot Virtual Machines is variable, based on region and SKU. For more information, see VM pricing for Linux and Windows.

You have option to set a max price you're willing to pay, per hour, for the VM. The max price for an Azure Spot Virtual Machine can be set in US dollars (USD), using up to 5 decimal places. For example, the value `0.98765` is a max price of $0.98765 USD per hour. If you set the max price to be `-1`, the VM is evicted based on price. The price for the VM is the current price for Azure Spot Virtual Machine or the price for a standard VM, which ever is less, as long as there's capacity and quota available. For more information about setting the max price, see Azure Spot Virtual Machines - Pricing.

The process to create an Azure Spot Virtual Machine using the Azure CLI is the same as detailed in the quickstart article. Just add the '--priority Spot' parameter, set the `--eviction-policy` to either `Deallocate` (the default) or `Delete`, and provide a max price or `-1`.

Install Azure CLI

To create Azure Spot Virtual Machines, you need to be running the Azure CLI version 2.0.74 or later. Run **az --version** to find the version. If you need to install or upgrade, see Install the Azure CLI.

Sign in to Azure using az login.

Azure CLI

```
az login
```

Create an Azure Spot Virtual Machine

This example shows how to deploy a Linux Azure Spot Virtual Machine that is not evicted based on price. The eviction policy is set to deallocate the VM, so that it can be restarted at a later time. If you want to delete the VM and the underlying disk when the VM is evicted, set `--eviction-policy` to `Delete`.

Azure CLI

```
az group create -n mySpotGroup -l eastus
az vm create \
    --resource-group mySpotGroup \
    --name myVM \
    --image Ubuntu2204 \
    --admin-username azureuser \
    --generate-ssh-keys \
    --priority Spot \
    --max-price -1 \
    --eviction-policy Deallocate
```

After the VM is created, you can query to see the max billing price for all of the VMs in the resource group.

Azure CLI

```
az vm list \
   -g mySpotGroup \
   --query '[].{Name:name, MaxPrice:billingProfile.maxPrice}' \
   --output table
```

Simulate an eviction

You can simulate an eviction of an Azure Spot Virtual Machine using REST, PowerShell, or the CLI, to test how well your application responds to a sudden eviction.

In most cases, you want to use the REST API Virtual Machines - Simulate Eviction to help with automated testing of applications. For REST, a `Response Code: 204` means the simulated eviction was successful. You can combine simulated evictions with the Scheduled Event service, to automate how your app responds when the VM is evicted.

To see scheduled events in action, watch Azure Friday - Using Azure Scheduled Events to prepare for VM maintenance.

Quick test

For a quick test to show how a simulated eviction works, let's walk through querying the scheduled event service to see what it looks like when you simulate an eviction using the Azure CLI.

The Scheduled Event service is enabled for your service the first time you make a request for events.

Remote into your VM, and then open a command prompt.

From the command prompt on your VM, type:

```
curl -H Metadata:true http://169.254.169.254/metadata/scheduledevents?api-version=2019-08-01
```

This first response could take up to 2 minutes. From now on, they should display output almost immediately.

From a computer that has the Azure CLI installed (like your local machine), simulate an eviction using az vm simulate-eviction. Replace the resource group name and VM name with your own.

Azure CLI

```
az vm simulate-eviction --resource-group mySpotRG --name mySpot
```

The response output has `Status: Succeeded` if the request was successfully made.

Quickly go back to your remote connection to your Spot Virtual Machine and query the Scheduled Events endpoint again. Repeat the following command until you get an output that contains more information:

```
curl -H Metadata:true http://169.254.169.254/metadata/scheduledevents?api-version=2019-08-01
```

When the Scheduled Event Service gets the eviction notification, you get a response that looks similar to the following output:

Output

```
{"DocumentIncarnation":1,"Events":[{"EventId":"A123BC45-1234-5678-AB90-ABCDEF123456","EventStatus":"Scheduled","EventType":"Preempt","ResourceType"
```

```
:"VirtualMachine","Resources":["myspotvm"],"NotBefore":"Tue, 16 Mar 2021
00:58:46 GMT","Description":"","EventSource":"Platform"}]}
```

You can see that `"EventType":"Preempt"`, and the resource is the VM resource `"Resources":["myspotvm"]`.

You can also see the VM's eviction condition by checking the `"NotBefore"` - the VM will not be evicted before the time given, so that is your window for your application to gracefully close out.

Next steps

You can also create an Azure Spot Virtual Machine using Azure PowerShell, portal, or a template.

Query current pricing information using the Azure retail prices API for information about Azure Spot Virtual Machine. The `meterName` and `skuName` both contain `Spot`.

If you encounter an error, see Error codes.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy Azure Spot Virtual Machines using the Azure portal

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale sets

Using Azure Spot Virtual Machines allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Azure Spot Virtual Machines. Therefore, Azure Spot Virtual Machines are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Azure Spot Virtual Machines is variable, based on region and SKU. For more information, see VM pricing for Linux and Windows. For more information about setting the max price, see Azure Spot Virtual Machines - Pricing.

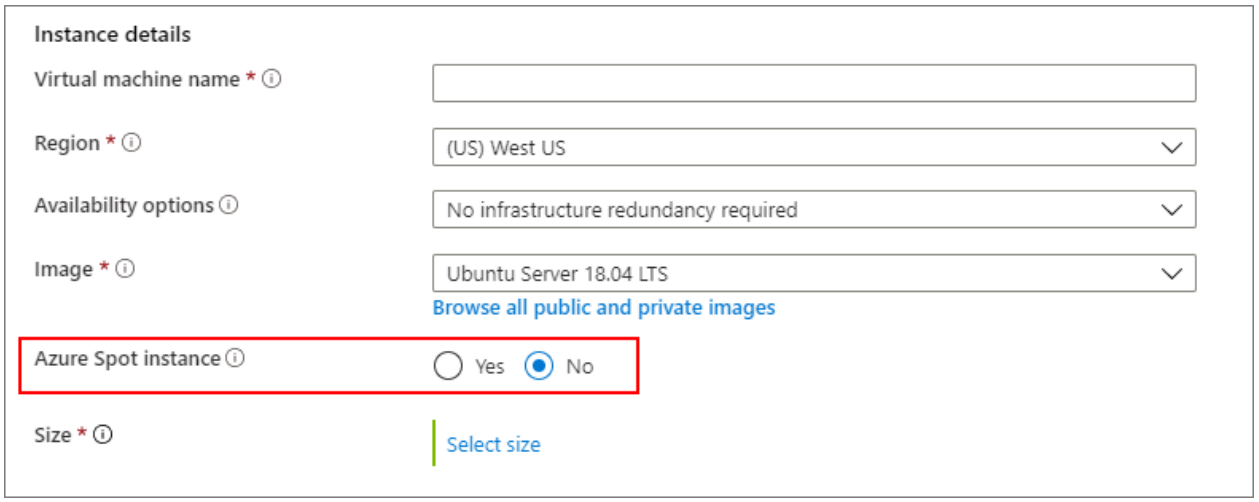
You have option to set a max price you are willing to pay, per hour, for the VM. The max price for an Azure Spot Virtual Machine can be set in US dollars (USD), using up to 5 decimal places. For example, the value `0.05701` would be a max price of $0.05701 USD per hour. If you set the max price to be `-1`, the VM won't be evicted based on price. The price for the VM will be the current price for spot or the price for a standard VM, which ever is less, as long as there is capacity and quota available.

When the VM is evicted, you have the option to either delete the VM and the underlying disk or deallocate the VM so that it can be restarted later.

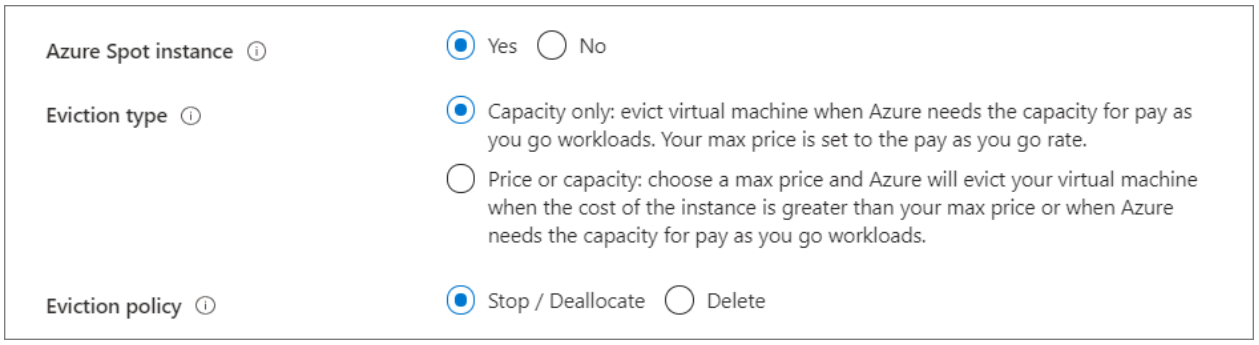
Create the VM

When you are deploying a VM, you can choose to use an Azure spot instance.

On the **Basics** tab, in the **Instance details** section, **No** is the default for using an Azure spot instance.

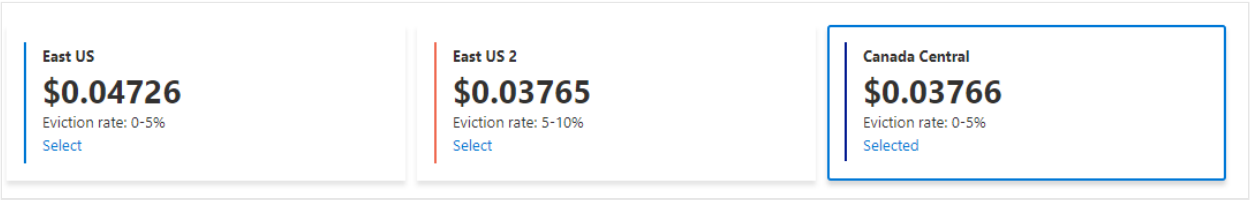


If you select **Yes**, the section expands and you can choose your eviction type and eviction policy.



You can also compare the pricing and eviction rates with other similar regions by selecting **View pricing history and compare prices in nearby regions**.

In this example, the Canada Central region is less expensive and has a lower eviction rate than the East US region.



You can change the region by selecting the choice that works the best for you and then selecting **OK**.

Simulate an eviction

You can simulate an eviction of an Azure Spot Virtual Machine, to testing how well your application will respond to a sudden eviction.

Replace the following with your information:

- `subscriptionId`
- `resourceGroupName`
- `vmName`

rest

```
POST
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmName}/simu
lateEviction?api-version=2020-06-01
```

`Response Code: 204` means the simulated eviction was successful.

Next steps

You can also create Azure Spot Virtual Machines using PowerShell, CLI, or a template.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy Azure Spot Virtual Machines using Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

Using Azure Spot Virtual Machines allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure will evict Azure Spot Virtual Machines. Therefore, Azure Spot Virtual Machines are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Azure Spot Virtual Machines is variable, based on region and SKU. For more information, see VM pricing for Linux and Windows. For more information about setting the max price, see Azure Spot Virtual Machines - Pricing.

You have option to set a max price you are willing to pay, per hour, for the VM. The max price for an Azure Spot Virtual Machine can be set in US dollars (USD), using up to 5 decimal places. For example, the value `0.98765` would be a max price of $0.98765 USD per hour. If you set the max price to be `-1`, the VM won't be evicted based on price. The price for the VM will be the current price for spot or the price for a standard VM, which ever is less, as long as there is capacity and quota available.

Create the VM

Create a spotVM using New-AzVmConfig to create the configuration. Include `-Priority Spot` and set `-MaxPrice` to either:

- `-1` so the VM is not evicted based on price.
- a dollar amount, up to 5 digits. For example, `-MaxPrice .98765` means that the VM will be deallocated once the price for a spotVM goes about $.98765 per hour.

This example creates a spotVM that will not be deallocated based on pricing (only when Azure needs the capacity back). The eviction policy is set to deallocate the VM, so that it can be restarted at a later time. If you want to delete the VM and the underlying disk when the VM is evicted, set `-EvictionPolicy` to `Delete` in `New-AzVMConfig`.

Azure PowerShell

```
$resourceGroup = "mySpotRG"
$location = "eastus"
```

```
$vmName = "mySpotVM"
$cred = Get-Credential -Message "Enter a username and password for the
virtual machine."
New-AzResourceGroup -Name $resourceGroup -Location $location
$subnetConfig = New-AzVirtualNetworkSubnetConfig `
   -Name mySubnet -AddressPrefix 192.168.1.0/24
$vnet = New-AzVirtualNetwork -ResourceGroupName $resourceGroup `
   -Location $location -Name MYvNET -AddressPrefix 192.168.0.0/16 `
   -Subnet $subnetConfig
$pip = New-AzPublicIpAddress -ResourceGroupName $resourceGroup -Location
$location `
  -Name "mypublicdns$(Get-Random)" -AllocationMethod Static -
IdleTimeoutInMinutes 4
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name
myNetworkSecurityGroupRuleRDP  -Protocol Tcp `
  -Direction Inbound -Priority 1000 -SourceAddressPrefix * -SourcePortRange
* -DestinationAddressPrefix * `
  -DestinationPortRange 3389 -Access Deny
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup -
Location $location `
  -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName $resourceGroup
-Location $location `
  -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -
NetworkSecurityGroupId $nsg.Id

# Create a virtual machine configuration and set this to be an Azure Spot
Virtual Machine

$vmConfig = New-AzVMConfig -VMName $vmName -VMSize Standard_D1 -Priority
"Spot" -MaxPrice -1 -EvictionPolicy Deallocate | `
Set-AzVMOperatingSystem -Windows -ComputerName $vmName -Credential $cred | `
Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer
WindowsServer -Skus 2016-Datacenter -Version latest | `
Add-AzVMNetworkInterface -Id $nic.Id

New-AzVM -ResourceGroupName $resourceGroup -Location $location -VM $vmConfig
```

After the VM is created, you can query to see the max price for all of the VMs in the resource group.

Azure PowerShell

```
Get-AzVM -ResourceGroupName $resourceGroup | `
   Select-Object Name,@{Name="maxPrice"; Expression=
{$_.BillingProfile.MaxPrice}}
```

Simulate an eviction

You can simulate an eviction of an Azure Spot Virtual Machine using REST, PowerShell, or the CLI, to test how well your application will respond to a sudden eviction.

In most cases, you will want to use the REST API Virtual Machines - Simulate Eviction to help with automated testing of applications. For REST, a `Response Code: 204` means the simulated eviction was successful. You can combine simulated evictions with the Scheduled Event service, to automate how your app will respond when the VM is evicted.

To see scheduled events in action, watch Azure Friday - Using Azure Scheduled Events to prepare for VM maintenance.

Quick test

For a quick test to show how a simulated eviction will work, let's walk through querying the scheduled event service to see what it looks like when you simulate an eviction using PowerShell.

The Scheduled Event service is enabled for your service the first time you make a request for events.

Remote into your VM, and then open a command prompt.

From the command prompt on your VM, type:

```
curl -H Metadata:true http://169.254.169.254/metadata/scheduledevents?api-version=2019-08-01
```

This first response could take up to 2 minutes. From now on, they should display output almost immediately.

From a computer that has the Az PowerShell module installed (like your local machine), simulate an eviction using Set-AzVM. Replace the resource group name and VM name with your own.

Azure PowerShell

```
Set-AzVM -ResourceGroupName "mySpotRG" -Name "mySpotVM" -SimulateEviction
```

The response output will have `Status: Succeeded` if the request was successfully made.

Quickly go back to your remote connection to your Spot Virtual Machine and query the Scheduled Events endpoint again. Repeat the following command until you get an output that contains more information:

```
curl -H Metadata:true http://169.254.169.254/metadata/scheduledevents?api-version=2019-08-01
```

When the Scheduled Event Service gets the eviction notification, you will get a response that looks similar to this:

Output

```
{"DocumentIncarnation":1,"Events":[{"EventId":"A123BC45-1234-5678-AB90-ABCDEF123456","EventStatus":"Scheduled","EventType":"Preempt","ResourceType":"VirtualMachine","Resources":["myspotvm"],"NotBefore":"Tue, 16 Mar 2021 00:58:46 GMT","Description":"","EventSource":"Platform"}]}
```

You can see that `"EventType":"Preempt"`, and the resource is the VM resource `"Resources":["myspotvm"]`.

You can also see when the VM will be evicted by checking the `"NotBefore"` value. The VM will not be evicted before the time given in `NotBefore`, so that is your window for your application to gracefully close out.

Next steps

You can also create an Azure Spot Virtual Machine using the Azure CLI, portal or a template.

Query current pricing information using the Azure retail prices API for information about Azure Spot Virtual Machine pricing. The `meterName` and `skuName` will both contain `Spot`.

If you encounter an error, see Error codes.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy Azure Spot Virtual Machines using a Resource Manager template

Article • 08/22/2024

Applies to: ✔ Linux VMs

Using Azure Spot Virtual Machines allows you to take advantage of our unused capacity at a significant cost savings. At any point in time when Azure needs the capacity back, the Azure infrastructure evicts Azure Spot VMs. Azure Spot VMs are great for workloads that can handle interruptions like batch processing jobs, dev/test environments, large compute workloads, and more.

Pricing for Azure Spot Virtual Machines is variable, based on region and SKU. For more information, see VM pricing for Linux and Windows.

You have option to set a max price you're willing to pay, per hour, for the VM. The max price for an Azure Spot VM can be set in US dollars (USD), using up to five decimal places. For example, the value `0.98765` would be a max price of $0.98765 USD per hour. If you set the max price to be `-1`, the VMs eviction is not based on price and it's price will be the current price for Azure Spot VMs or the price for a standard VM, whichever is less, as long as there's capacity and quota available. For more information about setting the max price, see Azure Spot VMs - Pricing.

Use a template

For Azure Spot VM template deployments, use `"apiVersion": "2019-03-01"` or later. Add the `priority`, `evictionPolicy` and `billingProfile` properties to in your template:

JSON

```
"priority": "Spot",
"evictionPolicy": "Deallocate",
"billingProfile": {
    "maxPrice": -1
}
```

Here's a sample template with added properties for an Azure Spot VM. Replace the resource names with your own and `<password>` with a password for the local administrator account on the VM.

JSON

```
{
    "$schema": "http://schema.management.azure.com/schemas/2019-03-
01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
    },
    "variables": {
        "vnetId": "/subscriptions/ec9fcd04-e188-48b9-abfc-
abcd515f1836/resourceGroups/spotVM/providers/Microsoft.Network/virtualNetwor
ks/spotVM",
        "subnetName": "default",
        "networkInterfaceName": "spotVMNIC",
        "publicIpAddressName": "spotVM-ip",
        "publicIpAddressType": "Dynamic",
        "publicIpAddressSku": "Basic",
        "virtualMachineName": "spotVM",
        "osDiskType": "Premium_LRS",
        "virtualMachineSize": "Standard_D2s_v3",
        "adminUsername": "azureuser",
        "adminPassword": "<password>",
        "diagnosticsStorageAccountName": "diagstoragespot2019",
        "diagnosticsStorageAccountId":
"Microsoft.Storage/storageAccounts/diagstoragespot2019",
        "diagnosticsStorageAccountType": "Standard_LRS",
        "diagnosticsStorageAccountKind": "Storage",
        "subnetRef": "[concat(variables('vnetId'), '/subnets/',
variables('subnetName'))]"
    },
    "resources": [
        {
            "name": "spotVM",
            "type": "Microsoft.Network/networkInterfaces",
            "apiVersion": "2019-03-01",
            "location": "eastus",
            "dependsOn": [
                "[concat('Microsoft.Network/publicIpAddresses/',
variables('publicIpAddressName'))]"
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "subnet": {
                                "id": "[variables('subnetRef')]"
                            },
                            "privateIPAllocationMethod": "Dynamic",
                            "publicIpAddress": {
                                "id": "[resourceId(resourceGroup().name,
'Microsoft.Network/publicIpAddresses', variables('publicIpAddressName'))]"
                            }
                        }
                    }
                ]
```

```
            }
        },
        {
            "name": "[variables('publicIpAddressName')]",
            "type": "Microsoft.Network/publicIpAddresses",
            "apiVersion": "2019-02-01",
            "location": "eastus",
            "properties": {
                "publicIpAllocationMethod": "
[variables('publicIpAddressType')]"
            },
            "sku": {
                "name": "[variables('publicIpAddressSku')]"
            }
        },
        {
            "name": "[variables('virtualMachineName')]",
            "type": "Microsoft.Compute/virtualMachines",
            "apiVersion": "2019-03-01",
            "location": "eastus",
            "dependsOn": [
                "[concat('Microsoft.Network/networkInterfaces/',
variables('networkInterfaceName'))]",
                "[concat('Microsoft.Storage/storageAccounts/',
variables('diagnosticsStorageAccountName'))]"
            ],
            "properties": {
                "hardwareProfile": {
                    "vmSize": "[variables('virtualMachineSize')]"
                },
                "storageProfile": {
                    "osDisk": {
                        "createOption": "fromImage",
                        "managedDisk": {
                            "storageAccountType": "
[variables('osDiskType')]"
                        }
                    },
                    "imageReference": {
                        "publisher": "Canonical",
                        "offer": "UbuntuServer",
                        "sku": "22.04-LTS",
                        "version": "latest"
                    }
                },
                "networkProfile": {
                    "networkInterfaces": [
                        {
                            "id": "
[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
                        }
                    ]
                },
                "osProfile": {
```

```
                    "computerName": "[variables('virtualMachineName')]",
                    "adminUsername": "[variables('adminUsername')]",
                    "adminPassword": "[variables('adminPassword')]"
                },
                "diagnosticsProfile": {
                    "bootDiagnostics": {
                        "enabled": true,
                        "storageUri": "[concat('https://',
variables('diagnosticsStorageAccountName'), '.blob.core.windows.net/')]"
                    }
                },
                "priority": "Spot",
                "evictionPolicy": "Deallocate",
                "billingProfile": {
                    "maxPrice": -1
                }
            }
        },
        {
            "name": "[variables('diagnosticsStorageAccountName')]",
            "type": "Microsoft.Storage/storageAccounts",
            "apiVersion": "2019-04-01",
            "location": "eastus",
            "properties": {},
            "kind": "[variables('diagnosticsStorageAccountKind')]",
            "sku": {
                "name": "[variables('diagnosticsStorageAccountType')]"
            }
        }
    ],
    "outputs": {
        "adminUsername": {
            "type": "string",
            "value": "[variables('adminUsername')]"
        }
    }
}
```

Simulate an eviction

You can simulate an eviction of an Azure Spot VM, to test your application response to a sudden eviction.

Replace the below parameters with your information:

- `subscriptionId`
- `resourceGroupName`
- `vmName`

```
HTTP
```

```
POST
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmName}/simu
lateEviction?api-version=2020-06-01
```

Next steps

- You can also create an Azure Spot VM using Azure PowerShell or the Azure CLI.
- For more information about Azure Spot VM current pricing, see Azure retail prices API. Both `meterName` and `skuName` contains `Spot`.
- To learn more about an error, see Error codes.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Spot Priority Mix for high availability and cost savings

10/08/2025

Applies to: ✔ Flexible scale sets

Spot Priority Mix allows you to run a combination of standard Virtual Machines (VMs) and Spot VMs within a single Virtual Machine Scale Set. This feature helps you balance cost savings with availability by letting Azure automatically manage the mix of VM types based on your requirements.

Overview

With Spot Priority Mix, you can:

- **Save up to 90%** on compute costs by using Spot VMs for interruptible workloads
- **Ensure availability** with standard VMs that aren't evicted
- **Protect against mass evictions** by maintaining a guaranteed number of standard VMs
- **Simplify management** with automatic orchestration of VM creation and deletion

How it works

Spot Priority Mix uses two key parameters to control your VM distribution:

1. `baseRegularPriorityCount`: The minimum number of standard (non-Spot) VMs that are always maintained
2. `regularPriorityPercentageAboveBase`: The percentage of standard VMs vs. Spot VMs for any capacity beyond the base count

Example: With a `baseRegularPriorityCount` of 10 and a `regularPriorityPercentageAboveBase` of 50:

- At 10 total VMs: All 10 are standard VMs (at or below base count)
- At 30 total VMs: 10 base standard VMs + 10 other standard VMs (50% of the 20 above base) + 10 Spot VMs (50% of the 20 above base)

Prerequisites

Before you use Spot Priority Mix, ensure you have the following:

- A Virtual Machine Scale Set with Flexible orchestration mode
- Understanding of Azure Spot VMs and their eviction behavior
- Appropriate quota for Spot VMs

Limitations

- Spot Priority Mix requires Flexible orchestration mode
- Not supported with `singlePlacementMode` enabled on the scale set
- Changes to the mix configuration apply only to future scaling operations (existing VMs aren't rebalanced)

Configure your mix

The platform automatically orchestrates scale-out and scale-in operations to maintain your desired distribution of Spot and standard VMs. You configure the mix using two key settings `baseRegularPriorityCount` and `regularPriorityPercentageAboveBase`.

Understanding the parameters

Expand table

Parameter	Description	Details
`baseRegularPriorityCount`	Sets the minimum number of standard VMs	When total capacity is at or below this number, all VMs are standard. Provides guaranteed capacity that isn't evicted.
`regularPriorityPercentageAboveBase`	Defines the standard-to-Spot ratio for VMs beyond the base count	Value between 0-100 (for example, 50 means 50% standard, 50% Spot). Only applies when capacity exceeds the base count.

Common configurations

Expand table

Use Case	Base Count	Percentage Above Base	Description
High availability	20	80%	Most VMs are standard, minimal Spot usage

Use Case	Base Count	Percentage Above Base	Description
Balanced	10	50%	Equal mix above base capacity
Cost-optimized	5	20%	Maximum Spot usage with small standard baseline
Dev/Test	0	0%	All Spot VMs (no availability guarantee)

Eviction policy

When Spot VMs are evicted due to capacity constraints or pricing, the eviction policy determines what happens:

Expand table

Policy	Behavior	Cost Impact
`Deallocate` (default)	Evicted VMs move to a stopped-deallocated state and can be restarted later	No compute charges, storage costs continue
`Delete`	Evicted VMs and their underlying disks are permanently deleted	No charges continue

Scale-in behavior

When you scale in, Spot Priority Mix maintains your configured percentage split by intelligently choosing which VMs to remove (Spot or standard) rather than simply removing the oldest or newest VMs.

ARM Template

Configure Spot Priority Mix in an ARM template by adding the `priorityMixPolicy` properties to a scale set with Flexible orchestration and Spot priority:

JSON

```
"priorityMixPolicy": {
    "baseRegularPriorityCount": 10,
    "regularPriorityPercentageAboveBase": 50
},
```

This example configuration:

- Maintains a minimum of 10 standard VMs
- For any VMs beyond 10, maintains a 50/50 split between standard and Spot VMs
- At 30 total VMs: 10 base + 10 standard + 10 Spot

Portal

Configure Spot Priority Mix when you create a Virtual Machine Scale Set in the Azure portal:

1. Sign in to the Azure portal.
2. Search for and select **Virtual Machine Scale Sets**.
3. Select **Create**.
4. On the **Basics** tab:
    - Fill out the required fields.
    - Set **Orchestration mode** to **Flexible**.
    - Select **Run with Azure Spot discount**.
5. On the **Spot** tab:
    - Select **Scale with VMs and Spot VMs** under **Scale with VMs and discounted Spot VMs**.
    - Set `baseRegularPriorityCount` by using **Base VM (uninterruptible) count**, the minimum number of standard VMs.
    - Set `regularPriorityPercentageAboveBase` by using **Instance distribution**, the percentage of standard VMs above the base count.
6. Complete the remaining configuration and create your scale set.

Update your Spot Priority Mix

You can modify the Spot Priority Mix configuration after your scale set is deployed. The updated configuration applies only to future scaling operations. Existing VMs remain unchanged until the scale set scales in or out.

Portal

Update your existing Spot Priority Mix in the Azure portal:

> **Note**

In the Azure portal, you can only update Spot Priority Mix for scale sets that already have this feature enabled.

1. Navigate to your Virtual Machine Scale Set in the Azure portal.
2. In the left menu, select **Configuration**.
3. Under the Spot Priority Mix section, update:
    - `baseRegularPriorityCount` by using **Base VM (uninterruptible) count**
    - `regularPriorityPercentageAboveBase` by using the **Instance distribution** percentage
4. Select **Save** to apply your changes.

Examples

The following examples demonstrate how Spot Priority Mix works in different scenarios. Each example includes a configuration, a table that shows the VM distribution after various operations, and a detailed walk-through.

Key terminology

- **Total capacity**: The total number of VMs in the Virtual Machine Scale Set.
- **Base standard VMs**: The guaranteed minimum number of standard VMs (set by `baseRegularPriorityCount`).
- **Extra standard VMs**: Standard VMs beyond the base count, calculated by using `regularPriorityPercentageAboveBase`.
- **Spot VMs**: Interruptible VMs that provide cost savings.

Scenario 1: 50/50 split with 10 base VMs

Configuration:

- `baseRegularPriorityCount`: 10
- `regularPriorityPercentageAboveBase`: 50%
- Eviction policy: Delete
- Starting capacity: 10 VMs

VM distribution during scaling:

Expand table

Action	Total capacity	Base standard VMs	Extra standard VMs	Spot VMs
Initial creation	10	10	0	0
Scale out to 20	20	10	5	5
Scale out to 30	30	10	10	10
Scale out to 40	40	10	15	15
Scale out to 41	41	10	15	16
Scale out to 42	42	10	16	16
All Spot VMs evicted	26	10	16	0
Scale out to 30	30	10	16	4
Scale out to 42	42	10	16	16
Scale out to 44	44	10	17	17

Walk-through:

1. **Initial state (10 VMs)**: All VMs are standard because the total is at the base count.
2. **Scale to 20 VMs**: Added 10 VMs above base, which is 5 standard (50%) and 5 Spot (50%).
3. **Scale to 30 VMs**: Added 20 VMs above base, which is 10 standard (50%) and 10 Spot (50%).
4. **Scale to 41 VMs**: With odd numbers, Spot VMs get the extra VM (16 Spot versus 15 standard).
5. **Scale to 42 VMs**: Balance restored with 16 of each type above the 10 base VMs.
6. **Eviction event**: All 16 Spot VMs are deleted, which leaves 26 total VMs (10 base and 16 extra standard).
7. **Scale to 30 VMs**: Only 4 Spot VMs added to rebalance toward the 50/50 target.
8. **Scale to 42 VMs**: Added 8 more Spot VMs and maintained 16 standard to restore balance.
9. **Scale to 44 VMs**: One of each type added to maintain the 50/50 split.

Scenario 2: Cost-optimized with 25% standard VMs

Configuration:

- `baseRegularPriorityCount`: 10
- `regularPriorityPercentageAboveBase`: 25%

- Eviction policy: Deallocate (VMs stop but aren't deleted)
- Starting capacity: 20 VMs

VM distribution during scaling:

Expand table

Action	Total capacity	Base standard VMs	Extra standard VMs	Spot VMs (running)	Spot VMs (deallocated)
Initial creation	20	10	2	8	0
Scale out to 50	50	10	10	30	0
Scale out to 110	110	10	25	75	0
10 Spot VMs evicted	110	10	25	65	10
Scale out to 120	120	10	27	73	10

Walk-through:

1. **Initial state (20 VMs)**: 10 base standard, 2 extra standard (25% of 10), and 8 Spot (75% of 10).
    - Formula: For the 10 VMs above base, 25% standard (2.5 rounded to 2) and 75% Spot (8).
2. **Scale to 50 VMs**: Added 30 VMs above base. Total of 40 above base equals 10 standard (25%) and 30 Spot (75%).
3. **Scale to 110 VMs**: 100 VMs above base, which is 25 standard (25%) and 75 Spot (75%).
4. **Eviction event**: 10 Spot VMs are deallocated (stopped but not deleted).
    - Total capacity remains 110, but only 65 Spot VMs are running.
    - The deallocated VMs count toward capacity but aren't running.
5. **Scale to 120 VMs**: Added 10 more VMs, which is 2 standard and 8 Spot, to maintain the 25/75 ratio.
    - The 10 deallocated VMs remain deallocated.

Troubleshooting

If Spot Priority Mix isn't available to you, be sure to configure the `priorityMixPolicy` to specify a *Spot* priority in the `virtualMachineProfile`. Without enabling the `priorityMixPolicy` setting, you won't be able to access this Spot feature.

Frequently asked questions

Why aren't my existing VMs changing after I updated the Spot Priority Mix?

Spot Priority Mix configuration applies only to future scaling operations. When you change the percentage split, existing VMs remain unchanged. The new distribution takes effect as the scale set scales in or out. To apply the new configuration, you can manually scale your set or wait for autoscale events.

What happens when Spot VMs are evicted?

When Azure needs capacity back, Spot VMs are evicted based on your eviction policy.

Your base standard VMs are never evicted, ensuring minimum capacity availability.

Can I use Spot Priority Mix with Uniform orchestration?

No, Spot Priority Mix is only available with Flexible orchestration mode. Uniform orchestration doesn't support this feature.

Which Azure regions support Spot Priority Mix?

Spot Priority Mix is available in all Azure regions that support Spot VMs. This availability includes all global Azure regions. For the most current region availability, see Spot VMs documentation.

Does changing the mix trigger immediate VM creation or deletion?

No. Updating your Spot Priority Mix configuration doesn't immediately create or delete VMs. The new settings apply during the next scale operation (scale in, scale out, or replacement of evicted VMs).

Can I have zero base standard VMs?

Yes, you can set `baseRegularPriorityCount` to 0, which means all VMs follow the percentage distribution. However, this means you could potentially lose all VMs to eviction during high-demand periods. For production workloads, we recommend that you maintain at least some base standard VMs.

Next steps

Learn more about Spot virtual machines

Spot Placement Score

10/07/2025

Spot Placement Score evaluates the likelihood of success for individual Spot deployments by considering parameters like desired Spot Virtual Machine (VM) count, VM size, and your deployment region or zone. This feature enables you to generate a placement score to deploy a desired number of Spot Virtual Machines (VMs) across various combinations of regions, zones, and VM sizes. By inputting lists of up to eight regions and five VM sizes, you can obtain placement scores categorized as either High, Medium, or Low. A score of High indicates that the deployment is highly likely to succeed while a score of Low indicates that the deployment has a low chance of success. These scores are based on analyses of Spot capacity allocation probability of the specified number of Spot VMs within each region and VM size combination. This functionality enhances deployment planning by providing predictive insights into deployment success and optimizing resource allocation for your Spot VMs.

Using Spot Placement Score, you can achieve the following:

- A clear evaluation of how likely your Spot deployment is to succeed based on specified parameters.
- Identify the most suitable combination of regions and VM sizes to maximize Spot VM availability based on placement scores.
- Improve the overall success rate of deploying Spot VMs by applying data-driven placement scores, reducing the risk of capacity issues or failures during deployment.

Cost

There are no costs associated with this feature.

Considerations

- Spot placement scores serve purely as a recommendation based on certain data points like Spot VM availability. A high placement score doesn't guarantee that the Spot request will be fully or partially fulfilled.
- Placement Scores are only valid at the time when it's requested. The same Placement Score isn't valid at a different time of the same day or another day. Any similarities are purely coincidental.

- The Spot Placement Score is only relevant if the Spot request has the same configuration as the Spot Placement Score configuration; desired count, VM size, location, and zone. In all other circumstances, the likelihood of getting available Spot capacity won't align with the placement score generated by the tool.
- Spot Placement Scores don't consider other constraints, such as Virtual Machine Scale Sets `SinglePlacementGroup`.
- A subscription's available Spot VM quota needs to be checked or requested separately.
- Spot Placement Score supports both regionally and zonally scoped placement score.
- Spot Placement Score API internally calls other GET APIs and is part of your GET call quota.
- A score of **High** or **Medium** doesn't guarantee allocation success or no evictions.
- This feature is available for all public Azure regions at this time.

Configure your Spot Placement Score

Configure your Spot Placement Score by defining your Spot specific requirements:

- Number of desired Spot VMs
- Up to five VM sizes
- Up to eight regions
- Availability zones

We recommend that the placement scores for each combination of subscription, desired count, region, zone, and VM size are cached, to avoid calling the API with the same configuration frequently within a short period of time. The suggested cache TTL is a minimum of 15 minutes and a maximum of 30 minutes.

Azure portal

Find the Spot Placement Score in the Spot tab of the Virtual Machine Scale Sets creation process in the Azure portal. The following steps instruct you on how to access this feature during that process.

1. Log in to the Azure portal.
2. In the search bar, search for and select **Virtual Machine Scale Sets**.
3. Select **Create** on the **Virtual Machine Scale Sets** page.

4. In the **Spot** tab, turn on the Spot option under the **Save money with Spot** section.
5. Fill out the *Size*, *Region*, *Availability Zones*, and *Initial instance count* fields in the **Your Placement Score** section.
6. Click on **Save + Apply** to receive your placement score for this configuration.

Examples

The following examples have scenario assumptions and a table with the results score to help you understand how Spot Placement Score works.

Scenario 1

This table is an example of a request returning regionally scoped placement scores for multiple desired VM sizes and regions.

The following scenario assumptions apply to this example:

- **Desired locations:** `westus`, `eastus`
- **Desired sizes:** `Standard_D2_v2`, `Standard_D4_v2`
- **Desired count:** 100
- **Availability zones:** False

Expand table

SKU	Region	Availability zone	Is quota available?	Placement score
Standard_D2_v2	westus	False	True	High
Standard_D4_v2	westus	False	True	Low
Standard_D2_v2	eastus	False	True	Medium
Standard_D4_v2	eastus	False	True	High

Scenario 2

This table is an example of a request returning zonally scoped placement scores for multiple desired VM sizes and regions.

The following scenario assumptions apply to this example:

- **Desired locations:** `westus`, `eastus`
- **Desired sizes:** `Standard_D2_v2`, `Standard_D4_v2`
- **Desired count:** 100
- **Availability zones:** True

Expand table

SKU	Region	Availability zone	Is quota available?	Placement score
Standard_D2_v2	westus	1	True	Medium
Standard_D2_v2	westus	2	True	Medium
Standard_D2_v2	westus	3	True	Medium
Standard_D4_v2	westus	1	True	High
Standard_D4_v2	westus	2	True	High
Standard_D4_v2	westus	3	True	High
Standard_D2_v2	eastus	1	True	Low
Standard_D2_v2	eastus	2	True	Low
Standard_D2_v2	eastus	3	True	Low
Standard_D4_v2	eastus	1	True	Medium
Standard_D4_v2	eastus	2	True	Medium
Standard_D4_v2	eastus	3	True	Medium

Troubleshooting

Expand table

Status code	Type	Condition
200	Successful request	Spot Placement Score operations complete successfully.
400	Bad error request	At least one required input parameter isn't present, or the values of the provided parameters aren't valid. Produces a detailed error message about the failed request.
429	Too many	Unable to generate placement score due to hitting a rate limit.

Status code	Type	Condition
	requests	
500	Internal server error	The placement score generation failed. Produces a detailed error message about the failed request.

Next steps

Learn about Spot Priority Mix

Spot Virtual Machine size recommendation

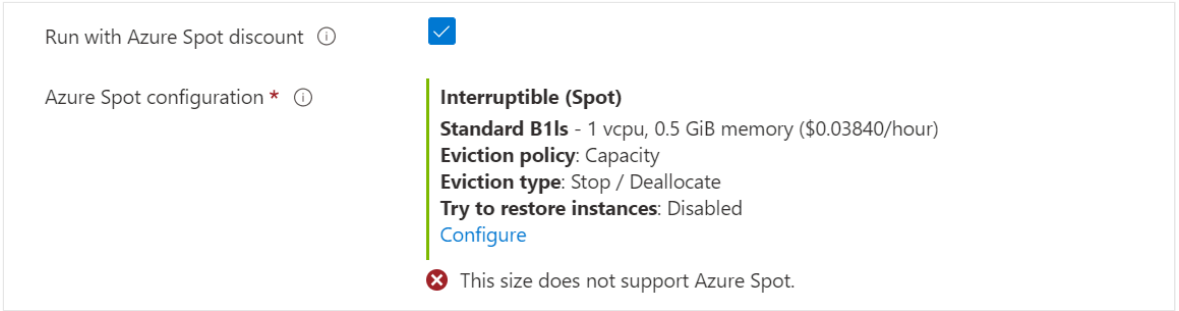
Article • 08/22/2024

The Spot VM size recommendations tool is an easy way to view and select alternative VM sizes that are better suited for your stateless, flexible, and fault tolerant workload needs during the Virtual Machine Scale Set deployment process in the Azure portal. This tool allows Azure to recommend appropriate VM sizes to you after you filter by region, price, and eviction rate. You can further filter the recommended VMs list by size, type, generation, and disk (premium or ephemeral OS disk).

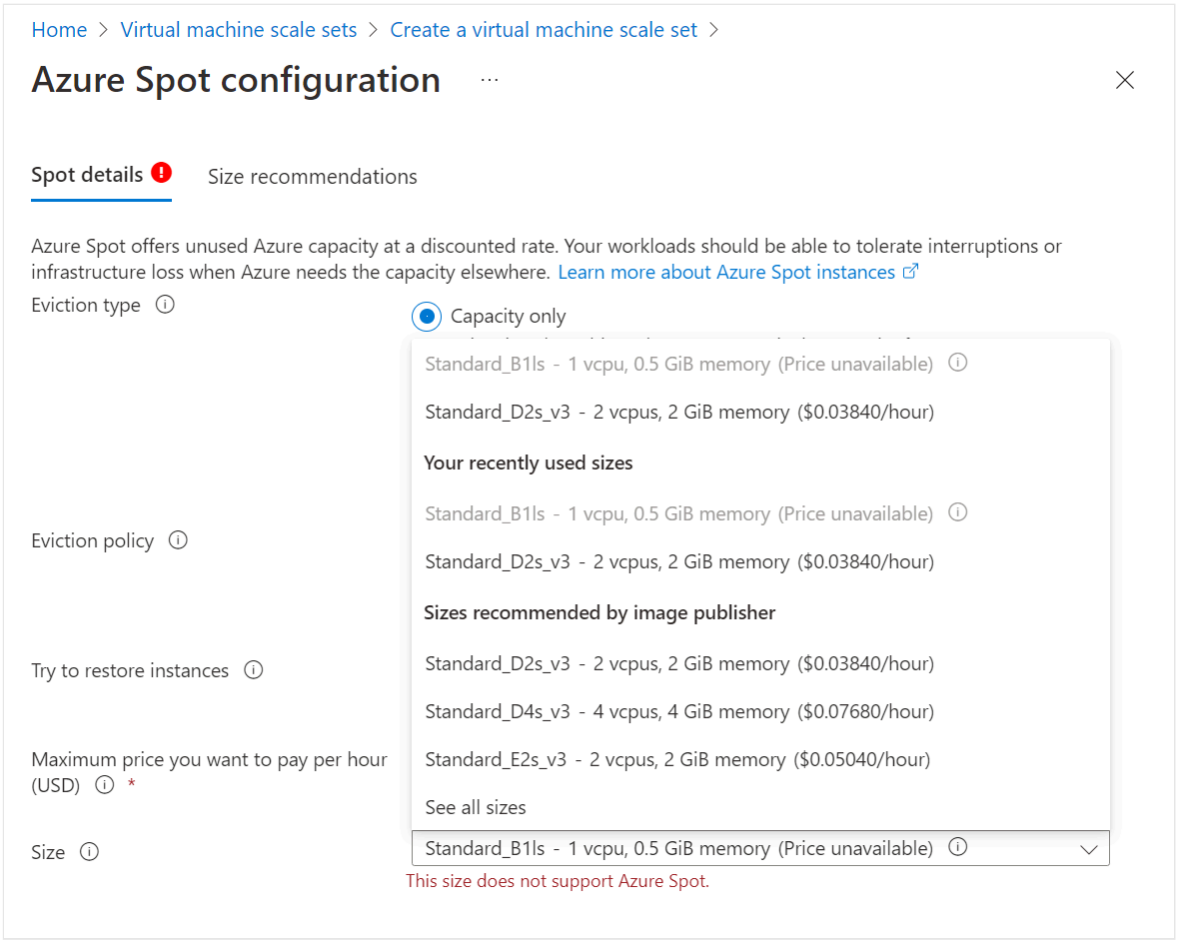
Azure portal

You can access Azure's size recommendations through the Virtual Machine Scale Sets creation process in the Azure portal. The following steps will instruct you on how to access this tool during that process.

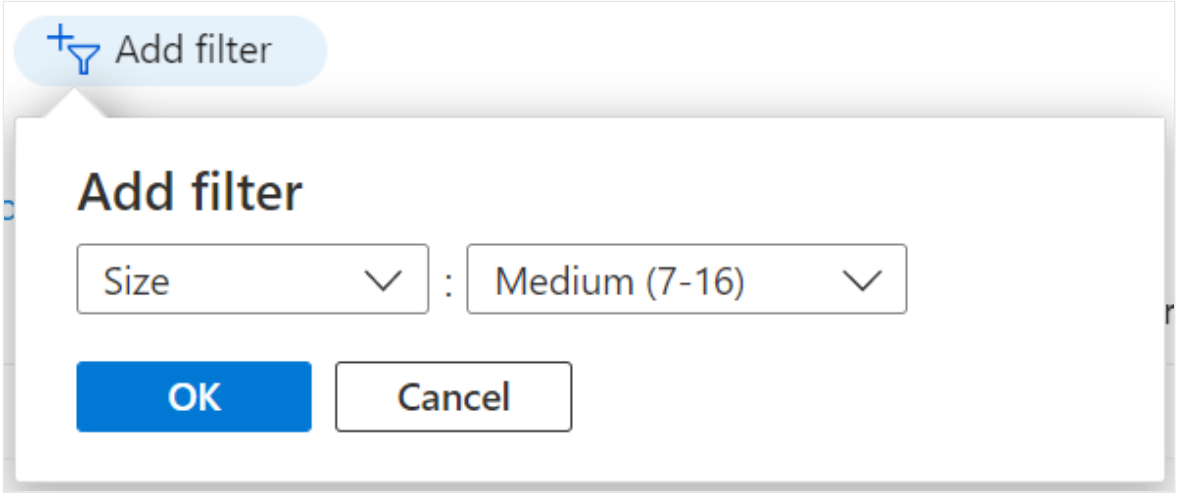
1. Log in to the Azure portal.
2. In the search bar, search for and select **Virtual Machine Scale Sets**.
3. Select **Create** on the **Virtual Machine Scale Sets** page.
4. In the **Basics** tab, fill out the required fields.
5. Under **Instance details**, select **Run with Azure Spot discount**.



6. In the same section, under **Azure Spot configuration**, select **Configure**.
7. On the **Azure Spot configuration** page, in the **Spot details** tab, go to the **Size** selector.
8. Expand the **Size** drop-down and select **See all sizes** option at the bottom of the list.

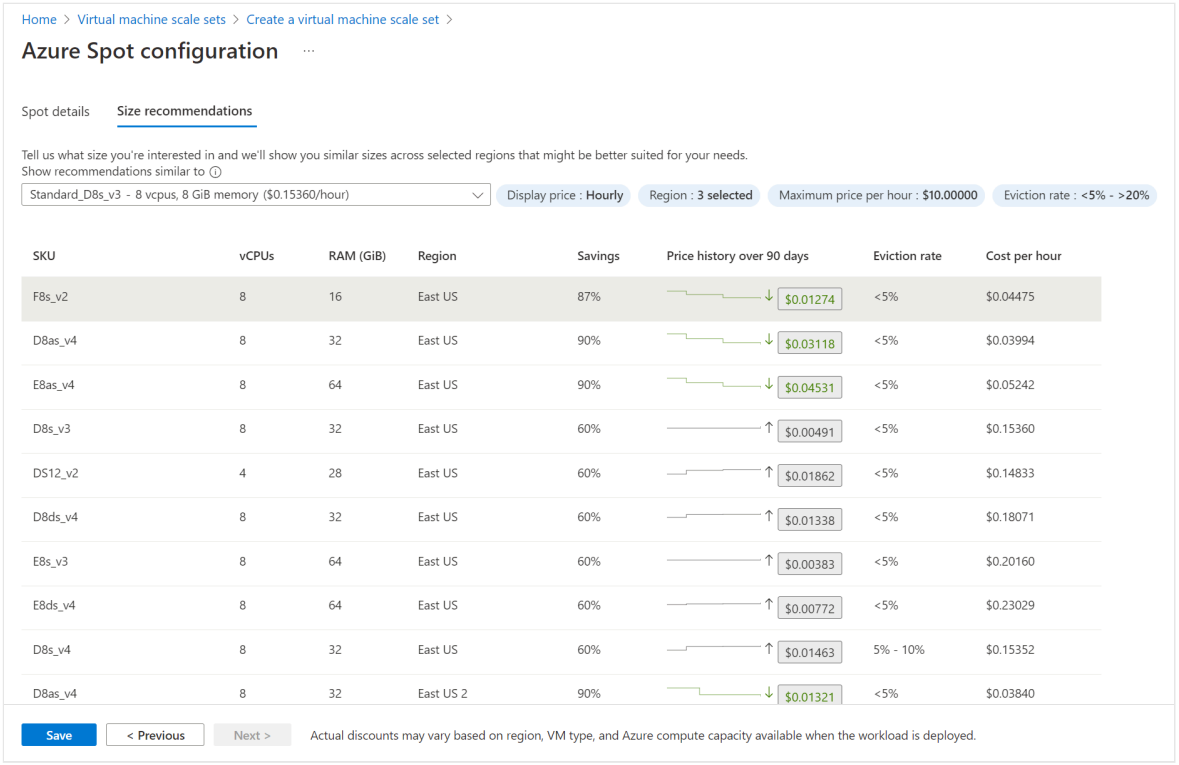


9. On the **Select a VM size** page, click **Add filter**.
10. You can choose which filters to apply. For this example, we will only apply **Size** and set it to *Medium (7-16)* for the number of vCPU.



11. Click **OK**.
12. From the resulting list of VMs, select a preferred VM size.
13. Click **Select** at the bottom to continue.
14. Back on the **Spot details** tab, click **Next** to go to the next tab.

15. The **Size recommendations** tab allows you to view and select alternative VM sizes that are better suited for your stateless, flexible, and fault tolerant workload needs with regard to region, pricing, and eviction rates.



16. Make your selection and click **Save**.

17. Continue through the Virtual Machine Scale Set creation process.

Next steps

Learn more about Spot virtual machines

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Error messages for Azure Spot Virtual Machines and scale sets

05/22/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

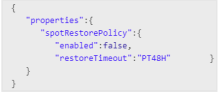
Here are some possible error codes you could receive when using Azure Spot Virtual Machines and scale sets.

Expand table

Key	Message	Description
SkuNotAvailable	The requested tier for resource '<resource>' is currently not available in location '<location>' for subscription '<subscriptionID>'. Try another tier or deploy to a different location.	There is not enough Azure Spot Virtual Machine capacity in this location to create your VM or scale set instance.
Standard_XX-XX_XX. This size does not support Azure Spot.	There is insufficient Azure Spot Virtual Machine capacity in this location to create your VM or scale set instance.	For Spot instances, if there is no available capacity, the portal page will fail immediately. This fail-fast feature ensures that users are promptly informed of the allocation failure, preventing them from waiting for an error after attempting deployment.

Key	Message	Description
EvictionPolicyCanBeSetOnlyOnAzureSpotVirtualMachines	Eviction policy can be set only on Azure Spot Virtual Machines.	This VM is not an Azure Spot Virtual Machine, so you can't set the eviction policy.
AzureSpotVMNotSupportedInAvailabilitySet	Azure Spot Virtual Machine is not supported in Availability Set.	You need to choose to either use an Azure Spot Virtual Machine or use a VM in an availability set, you can't choose both.
AzureSpotFeatureNotEnabledForSubscription	Subscription not enabled with Azure Spot Virtual Machine feature.	Use a subscription that supports Azure Spot Virtual Machines.
VMPriorityCannotBeApplied	The specified priority value '{0}' cannot be applied to the Virtual Machine '{1}' since no priority was specified when the Virtual Machine was created.	Specify the priority when the VM is created.
SpotPriceGreaterThanProvidedMaxPrice	Unable to perform operation '{0}' since the provided max price '{1} USD' is lower than the current spot price '{2} USD' for Azure Spot Virtual Machine size '{3}'.	Select a higher max price. For more information, see pricing information for Linux or Windows.
MaxPriceValueInvalid	Invalid max price value. The only supported values for max price are -1 or a decimal greater	Enter a valid max price. For more information, see pricing for

Key	Message	Description
	than zero. Max price value of -1 indicates the Azure Spot Virtual Machine will not be evicted for price reasons.	Linux or Windows.
MaxPriceChangeNotAllowedForAllocatedVMs	Max price change is not allowed when the VM '{0}' is currently allocated. Deallocate and try again.	Stop\Deallocate the VM so that you can change the max price.
MaxPriceChangeNotAllowed	Max price change is not allowed.	You cannot change the max price for this VM.
AzureSpotIsNotSupportedForThisAPIVersion	Azure Spot Virtual Machine is not supported for this API version.	The API version needs to be 2019-03-01.
AzureSpotIsNotSupportedForThisVMSize	Azure Spot Virtual Machine is not supported for this VM size {0}.	Select another VM size. For more information, see Azure Spot Virtual Machines.
MaxPriceIsSupportedOnlyForAzureSpotVirtualMachines	Max price is supported only for Azure Spot Virtual Machines.	For more information, see Azure Spot Virtual Machines.
MoveResourcesWithAzureSpotVMNotSupported	The Move resources request contains an Azure Spot Virtual Machine. Not supported. Check the error details for virtual machine Ids.	You cannot move Azure Spot Virtual Machines.
MoveResourcesWithAzureSpotVmssNotSupported	The Move resources request contains an Azure Spot virtual machine scale set. Not supported.	You cannot move Azure Spot virtual machine scale set instances.

Key	Message	Description
	Check the error details for virtual machine scale set Ids.	
AzureSpotVMNotSupportedInVmssWithVMOrchestrationMode	Azure Spot Virtual Machine is not supported in Virtual machine scale set with VM Orchestration mode.	Set the orchestration mode to virtual machine scale set in order to use Azure Spot Virtual Machine instances.
SpotRestorationIsNotSupportedForThisAPIVersion	Spot restoration feature is not supported for this API version.	For an existing scaleset, perform a PATCH using API version 2021-07-01 or later. For new scale set deployments, add the following property to the Azure Resource Manager template using API version 2021-07-01 or later: 
SpotRestorationIsSupportedOnlyForAzureSpotScaleSets	Spot restoration feature is supported only for Azure Spot virtual machine scale sets.	Spot restoration feature is only supported for Azure Spot virtual machine scale sets. To use this feature, deploy Azure Spot using virtual machine scale sets.

Next steps For more information, see spot Virtual Machines.

Build workloads on spot virtual machines

Azure Virtual Machines

This article describes best practices for how to build on Azure Spot Virtual Machines. Spot virtual machines (spot VMs) provide access to compute capacity at lower prices than regular VMs. This discount makes them a good option for organizations that want to optimize costs. But the savings come with a trade-off. Spot VMs can be evicted at any time, which means that they lose access to compute resources. Workloads that run on spot VMs must be able to handle these interruptions in compute. The appropriate workload and a flexible orchestration mechanism are the keys to success. The following recommendations describe how to build on spot VMs.

Understand spot VMs

On a technical level, spot VMs are the same as regular VMs. They use the same images, hardware, and disks that translate to the same performance. The key difference between spot VMs and regular VMs is their priority and availability. Spot VMs have no priority to access compute capacity, and they have no availability guarantees after they access that compute capacity.

- **No priority access.** Regular VMs have priority access to compute capacity. They access compute capacity when they request it. However, spot VMs only deploy when there's spare compute capacity. And they only continue to run when a regular VM doesn't need the underlying hardware.
- **No availability guarantee.** Spot VMs don't have any availability guarantees or service-level agreements (SLAs). Spot VMs can lose access to compute capacity immediately, or anytime after deployment or eviction. Spot VMs are cheaper because they can be evicted. When Azure needs the compute capacity back, an eviction notice is sent and evicts the spot VM. Azure provides a minimum of 30-seconds advance notice before the actual eviction occurs. For more information, see Continuously monitor for eviction.

Understand spot VM pricing

Spot VMs can be up to 90% cheaper than regular pay-as-you-go VMs. The discount varies based on demand, VM size, deployment region, and the operating system. To get a cost savings estimate, see Azure Spot Virtual Machines pricing tool and Spot Virtual Machines pricing overview. You can also query the Azure retail prices API to programmatically obtain the spot pricing for any SKU.

Understand interruptible workloads

Spot VMs are ideal for interruptible workloads, which share several common characteristics. Interruptible workloads have minimal to no time constraints, low organizational priority, and short processing times. They run processes that can stop suddenly and resume later without harming essential organizational processes. Examples of interruptible workloads are batch processing applications, data analytics, and workloads that create a continuous integration and continuous deployment agent for a nonproduction environment. These features compare to regular or mission-critical workloads that have SLAs, sticky sessions, and stateful data.

You can use spot VMs in non-interruptible workloads, but they shouldn't be the only source of compute capacity. Use as many regular VMs as you need to meet your uptime requirements.

Understand eviction

Spot VMs have no SLAs after creation, and they can lose access to compute at any time. We call this compute loss an *eviction*. Compute supply and demand drive evictions. When the demand for a specific VM size exceeds a specific level, Azure evicts spot VMs to make compute available to regular VMs. Demand is location specific. For example, an increase in demand in region A doesn't affect spot VMs in region B.

Spot VMs have two configuration options that affect eviction. These configurations are the *eviction type* and *eviction policy* of the spot VM. You set these configurations when you create the spot VM. The eviction type defines the conditions of an eviction. The eviction policy determines what eviction does to your spot VM.

Eviction type

Capacity changes or price changes cause evictions. The way capacity and price changes affect spot VMs depends on the eviction type that you choose when the VM is created. The type of eviction defines the conditions of an eviction. The eviction types are *capacity-only eviction* and *price or capacity eviction*.

- **Capacity-only eviction:** This eviction type triggers an eviction when excess compute capacity is no longer available. By default, the price is capped at the pay-as-you-go rate. Use this eviction type when you don't want to pay more than the pay-as-you-go VM price.
- **Price or capacity eviction:** This eviction type has two triggers. Azure evicts a spot VM when excess compute capacity is no longer available or the cost of the VM exceeds the maximum price that you set. This eviction type allows you to set a maximum price far less than the pay-as-you-go price. Use this eviction type to set your own price cap.

Eviction policy

The eviction policy that you choose for a spot VM affects its orchestration. Orchestration is the process of handling an eviction and is discussed later in this article. The eviction policies are the *Stop/Deallocate policy* and the *Delete policy*.

Stop/Deallocate policy: The Stop/Deallocate policy is ideal when the workload can wait for release capacity within the same location and VM type. The Stop/Deallocate policy stops the VM and ends its lease with the underlying hardware. Stopping and deallocating a spot VM is the same as stopping and deallocating a regular VM. The VM remains accessible in Azure, and you can restart the same VM later. The VM loses compute capacity and nonstatic IP addresses with the Stop/Deallocate policy. However, the VM data disks remain and continue to incur charges. The VM also occupies cores in the subscription. VMs can't be moved from their region or zone even when they're stopped or deallocated. For more information, see Power states and billing.

Delete policy: Use the Delete policy if the workload can change location or VM size. Changing the location or VM size allows the VM to redeploy faster. The Delete policy deletes the VM and any data disk. The VM doesn't occupy cores in subscriptions. For more information, see Eviction policy. When you use the Delete policy, consider using ephemeral OS disks for your spot VMs. Because the VM and its disks are deleted on eviction, a persistent OS disk incurs storage cost for a resource that doesn't survive eviction. Ephemeral OS disks eliminate that cost and reduce VM provisioning time, which helps your orchestration replace evicted VMs faster. Ephemeral OS disks aren't compatible with the Stop/Deallocate policy because deallocation destroys the OS disk state.

Design for flexible orchestration

Orchestration is the process of replacing a spot VM after an eviction. It's the foundation for building a reliably interruptible workload. A good orchestration system has built-in flexibility. Flexibility means designing your orchestration to have options, use multiple VM sizes, deploy to different regions, have eviction awareness, and account for different eviction scenarios to improve workload reliability and speed.

Design for speed

For a workload that runs on spot VMs, compute capacity is crucial. Because of the potential for eviction, ensure that you understand the allocated compute time so that you can make informed design decisions that prioritize workload speed. Generally, you should optimize the compute time that you have. Build a VM image that has all the required software preinstalled. Preinstalled software helps minimize the time between eviction and a fully operational

application. Avoid using compute time on processes that don't contribute to the workload purpose. For example, a workload for data analytics should focus most of its compute time on data processing and as little time as possible on gathering eviction metadata. Eliminate nonessential processes from your application.

Use multiple VM sizes and locations

To increase flexibility, build an orchestration to use multiple types and sizes of VMs. The goal is to give your orchestration options to replace an evicted VM. Azure has different types and sizes of VMs that provide similar capabilities for about the same price. Filter for the minimum vCPUs or cores, the minimum RAM for VMs, and the maximum price. This process helps you find multiple VMs that fit in your budget and have enough power to run your workload.

Each type of VM has an eviction rate that's expressed as a percentage range, such as 0%-5%, 5%-10%, 10%-15%, 15%-20%, or 20+%. Eviction rates can vary across regions. You might find a better eviction rate for the same type of VM in a different region. You can find the eviction rates for each type of VM in the portal under the **Basics** tab. Next to **Size**, select **View pricing history** or **See all sizes**. You can also programmatically get spot VM data by using Azure Resource Graph.

In your orchestration system, consider using the spot placement score feature to evaluate the likelihood of success for individual spot deployments.

For more information, see the following resources:

- Eviction rates
- Resource Graph
- Spot placement score

Use the most flexible eviction policy

The eviction policy of the evicted spot VM affects the replacement process. For example, a Delete policy is more flexible than a Stop/Deallocate policy.

- **Consider the Delete policy first:** Use a Delete policy if your workload can handle it. Deletion allows the orchestration to deploy replacement spot VMs to new zones and regions. This deployment flexibility could help your workload find spare compute capacity faster than a stopped or deallocated VM. Stopped or deallocated VMs have to wait for spare compute capacity in the same zone that they were created in. For the Delete policy, you need an external process to monitor for evictions and orchestrate deployments to various regions, use different VM SKUs, or both.

- **Understand the Stop/Deallocate policy:** The Stop/Deallocate policy has less flexibility than the Delete policy. The spot VMs must stay in the same region and zone. You can't move a stopped or deallocated VM to another location. Because the VMs have a fixed location, you need something in place to reallocate the VM when compute capacity becomes available. There's no way to predict the availability of compute capacity. So you should use an automated schedule pipeline to attempt a redeployment after an eviction. An eviction should trigger the schedule pipeline, and the redeployment attempts should continuously check for compute capacity until it becomes available.

Expand table

Policy	When to use the policy
Delete policy	- Ephemeral compute and data - Don't want to pay for data disks - Minimal budget
Stop/Deallocate policy	- Need a specific VM size - Can't change location - Long application installation process - Indefinite wait time - Not driven by cost savings alone

Continuously monitor for eviction

Monitoring is the key to workload reliability on spot VMs. Spot VMs have no SLA after creation and can be evicted at any time. The best way to improve workload reliability on spot VMs is to anticipate when they're going to be evicted. If you have this information, you can attempt a workload graceful shutdown and trigger automation to orchestrate the replacement.

- **Use Scheduled Events:** Use the Scheduled Events service for each VM. Azure sends signals to VMs when infrastructure maintenance is going to affect them. Evictions qualify as infrastructure maintenance. Azure sends out the `Preempt` signal to all VMs at a minimum of 30 seconds before they're evicted. The Scheduled Events service allows you

to capture this `Preempt` signal by querying an endpoint at the static, nonroutable IP address `169.254.169.254`.

- **Use frequent queries:** Query the Scheduled Events endpoint often enough to orchestrate a graceful shutdown. You can query the Scheduled Events endpoint up to every second, but a one-second frequency might not be necessary for all use cases. These queries must come from an application that runs on the spot VM. The query can't come from an external source. As a result, the queries consume VM compute capacity and steal processing power from the main workload. You need to balance those competing priorities to meet your specific situation.

- **Automate orchestration:** After you collect the `Preempt` signal, your orchestration should act on that signal. Given the time constraints, the `Preempt` signal should attempt a graceful shutdown of your workload and start an automated process that replaces the spot VM. For more information, see the following resources:
  - Scheduled Events
  - Scheduled event types
  - Endpoint querying frequency

Build a deployment system

Your orchestration needs an automated pipeline to deploy new spot VMs after eviction. The pipeline should run outside the interruptible workload to help ensure permanence. The deployment pipeline should work according to the eviction policy that you choose for your spot VMs.

For a Delete policy, we recommend that you build a pipeline that uses different VM sizes and deploys to different regions. For a Stop/Deallocate policy, the deployment pipeline needs two distinct actions. For the initial creation of a VM, the pipeline needs to deploy the correct size VMs to the correct location. For an evicted VM, the pipeline needs to try to restart the VM until it works. A combination of Azure Monitor alerts and Azure functions is one way to automate a deployment system. The pipeline could use bicep templates. They're declarative and idempotent and represent a best practice for infrastructure deployment.

Prepare for immediate eviction

It's possible for Azure to evict a spot VM as soon as you create it and before your workload runs. In some cases, there might be enough capacity to create a spot VM, but it won't last. Spot VMs have no availability guarantees, or SLAs, after creation. Your orchestration needs to account for immediate evictions. The `Preempt` signal provides a minimum of 30-seconds advance notice of the eviction.

Incorporate VM health checks into your orchestration to prepare for immediate evictions. Orchestration for immediate evictions can't depend on the Scheduled Events `Preempt` signal. Only the VM itself can query the `Preempt` signal, and there's not enough time to start an application, query the Scheduled Events endpoint, and gracefully shut down. So the health check needs to reside outside the workload environment. The health checks need to monitor the status of the spot VM and start the deployment pipeline to replace the spot VM when the status changes to *deallocating* or *stopping*.

Plan for multiple simultaneous evictions

If you run a cluster of spot VMs, architect the workload to withstand multiple simultaneous evictions. Multiple spot VMs in the workload can be evicted at the same time. A simultaneous eviction of multiple VMs could affect the throughput of the application. To prevent this situation, your deployment pipeline should be able to gather signals from multiple VMs and deploy multiple replacement VMs simultaneously.

Design for a graceful shutdown

The VM shutdown process should be less than 30 seconds and allow your VM to shut down before an eviction. The amount of time the shutdown should take depends on how frequently your workload queries the Scheduled Events endpoint. The more often you query the endpoint, the longer the shutdown process can take. The shutdown process should release resources, drain connections, and flush event logs. You should regularly create and save checkpoints to retain the context and build a more efficient recovery strategy. The checkpoint is just information about what processes or transactions the next VM needs to start on. They should indicate if the VM should resume where the previous VM left off or if the new VM should revert the changes and start the entire process again. Store the checkpoints outside the spot VM environment, like in a storage account.

Test the orchestration

Simulate eviction events to test orchestration in dev/test environments. For more information, see Simulate eviction.

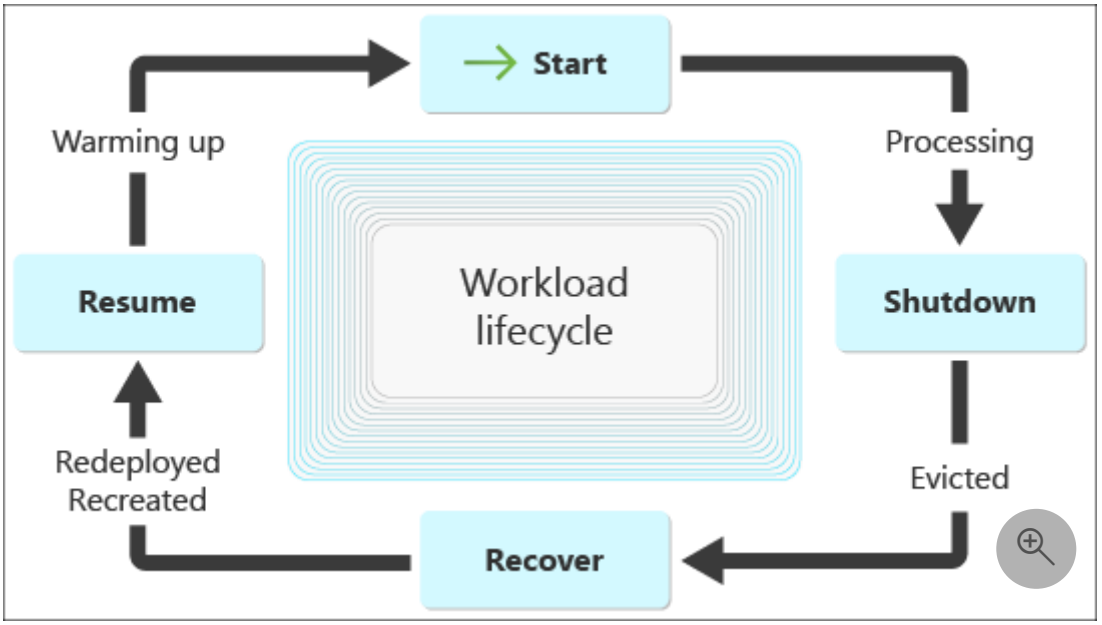
Design an idempotent workload

We recommend that you design an idempotent workload. The outcome of processing an event more than one time should be the same as processing it once. Evictions can result in forced shutdowns, despite efforts to ensure graceful shutdowns. Forced shutdowns can terminate

processes before completion. Idempotent workloads can receive the same message more than one time without changing the outcome. For more information, see Idempotency.

Use an application warmup period

Most interruptible workloads run applications. Applications need time to install and to start up. They also need time to connect to external storage and gather information from checkpoints. Have an application warmup period before you allow it to start processing. During the warmup period, the application should start, establish connections, and prepare to contribute. Only allow an application to start processing data after you validate the health of the application.



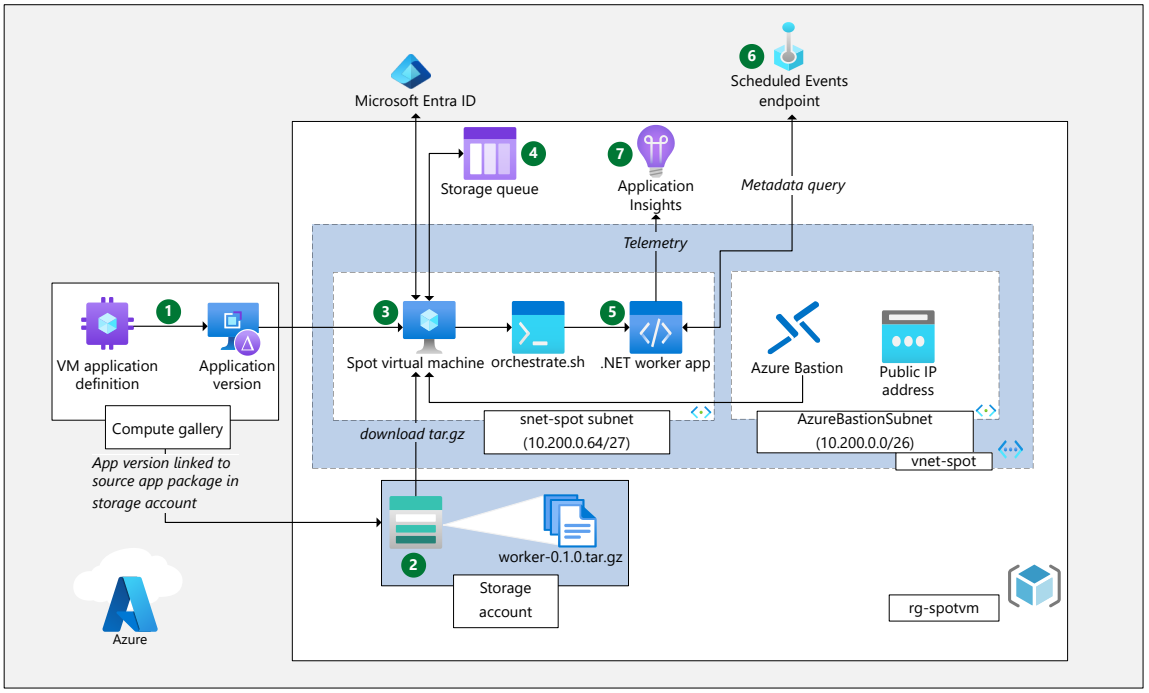
Configure user-assigned managed identities

Assign user-assigned managed identities to streamline the authentication and authorization process. User-assigned managed identities let you avoid putting credentials in code and aren't tied to a single resource like system-assigned managed identities. The user-assigned managed identities contain permissions and access tokens from Microsoft Entra ID that can be reused and assigned to spot VMs during orchestration. Token consistency across spot VMs helps streamline orchestration and simplifies the access to workload resources that the spot VMs have.

If you use system-assigned managed identities, a new spot VM might get a different access token from Microsoft Entra ID. If you need to use system-assigned managed identities, make the workloads resilient to `403 Forbidden Error` responses. Your orchestration needs to get tokens from Microsoft Entra ID with the correct permissions. For more information, see Managed identities.

Example scenario

The example scenario deploys a queue processing application that qualifies as an interruptible workload. The scripts in the scenario serve as examples. The scenario guides you through a one-time, manual push to deploy resources. This implementation doesn't have a deployment pipeline. However, a deployment pipeline is essential to automate the orchestrating process. The following diagram shows the architecture of the example scenario.



Microsoft Azure

[Download a Visio file](#) of this architecture.

The following workflow corresponds to the previous diagram:

1. **VM application definition:** The VM application definition is created in the Azure compute gallery. It defines the application name, location, operating system, and metadata. The application version is a numbered version of the VM application definition. The application version represents the VM application. It needs to be in the same region as the spot VM. The application version links to the source application package in the storage account.

2. **Storage account:** The storage account stores the source application package. In this architecture, it's a compressed tar file named `worker-0.1.0.tar.gz`. It contains two files. One file is the `orchestrate.sh` bash script that installs the .NET worker application.

3. **Spot VM:** The spot VM deploys. It must be in the same region as the application version. It downloads `worker-0.1.0.tar.gz` to the VM after deployment. The bicep template

deploys an Ubuntu image on a standard family VM. These configurations meet the needs of this application and aren't general recommendations for your applications.

4. **Storage queue:** The other service that runs in the .NET worker contains message queue logic. Microsoft Entra ID grants the spot VM access to the storage queue in Azure Queue Storage with a user-assigned identity by using role-based access control.

5. **.NET worker application:** The `orchestrate.sh` script installs a .NET worker application that runs two background services. The first service queries the Scheduled Events endpoint, looks for the `Preempt` signal, and sends this signal to the second service. The second service processes messages from the storage queue and listens for the `Preempt` signal from the first service. When the second service receives the signal, it interrupts storage queue processing and begins to shut down.

6. **Query Scheduled Events endpoint:** An API request is sent to a static nonroutable IP address `169.254.169.254`. The API request queries the Scheduled Events endpoint for infrastructure maintenance signals.

7. **Application Insights:** The architecture uses Application Insights only for learning purposes. It's not an essential component of interruptible workload orchestration, but allows you to validate the telemetry from the .NET worker application. The .NET worker application sends telemetry to Application Insights via the Azure Monitor OpenTelemetry Distro. For more information, see Enable live metrics from the .NET application.

Next step

Spot Virtual Machines

Related resources

- Architectural approaches for cost management and allocation in a multitenant solution
- Run a Windows VM on Azure
- Run a Linux VM on Azure

Microsoft Azure Boost

Article • 03/18/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Sizes

Azure Boost is a system designed by Microsoft that offloads server virtualization processes traditionally performed by the hypervisor and host OS onto purpose-built software and hardware. This offloading frees up CPU resources for the guest virtual machines, resulting in improved performance. Azure Boost also provides a secure foundation for your cloud workloads. Microsoft's in-house developed hardware and software systems provide a secure environment for your virtual machines.

Benefits

Azure Boost contains several features that can improve the performance and security of your virtual machines. These features are available on select Azure Boost compatible virtual machine sizes.

- **Networking:** Azure Boost includes a suite of software and hardware networking systems that provide a significant boost to both network performance (Up to 200-Gbps network bandwidth) and network security. Azure Boost compatible virtual machine hosts contain the new Microsoft Azure Network Adapter (MANA). Learn more about Azure Boost networking.

- **Storage:** Storage processing operations are offloaded to the Azure Boost FPGA. This offload to FPGA provides leading efficiency and performance while improving security, reducing jitter, and improving latency for workloads. Local storage now runs at up to 36GBps throughput and 6.6 million IOPS, and with remote storage up to 14GBps throughput and 750K IOPS. Learn more about Azure Boost Storage.

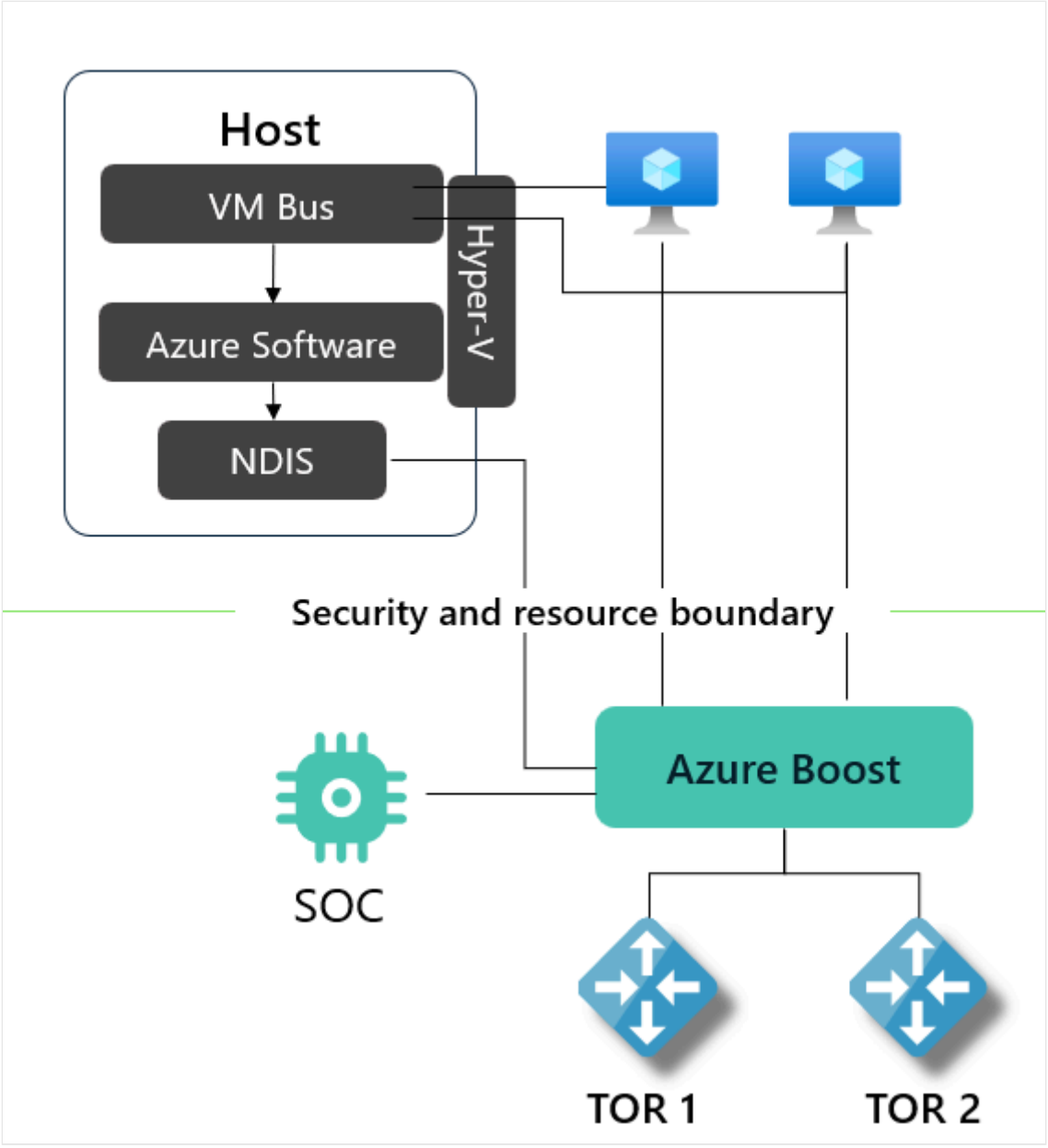
- **Security:** Azure Boost uses Cerberus as an independent HW Root of Trust to achieve NIST 800-193 certification. Customer workloads can't run on Azure Boost powered architecture unless the firmware and software running on the system is trusted. Learn more about Azure Boost Security.

- **Performance:** With Azure Boost offloading storage and networking, CPU resources are freed up for increased virtualization performance. Resources that would normally be used for these essential background tasks are now available to the guest VM. Learn more about Azure Boost Performance.

Networking

The next generation of Azure Boost will introduce the Microsoft Azure Network Adapter (MANA). This network interface card (NIC) includes the latest hardware acceleration features and provides competitive performance with a consistent driver interface. This custom hardware and software implementation ensures optimal networking performance, tailored specifically for Azure's demands. MANA's features are designed to enhance your networking experience with:

- **Over 200-Gbps of network bandwidth:** Custom hardware and software drivers facilitating faster and more efficient data transfers. Starting up to 200Gbps network bandwidth with increases in the future.
- **High network availability and stability:** With an active/active network connection to the Top of Rack (ToR) switch, Azure Boost ensures your network is always up and running at the highest possible performance.
- **Native support for DPDK:** Learn more about Azure Boost's support for Data Plane Development Kit (DPDK) on Linux VMs.
- **Consistent driver interface:** Assuring a one-time transition that won't be disrupted during future hardware changes.
- **Integration with future Azure features:** Consistent updates and performance enhancements ensures you're always a step ahead.

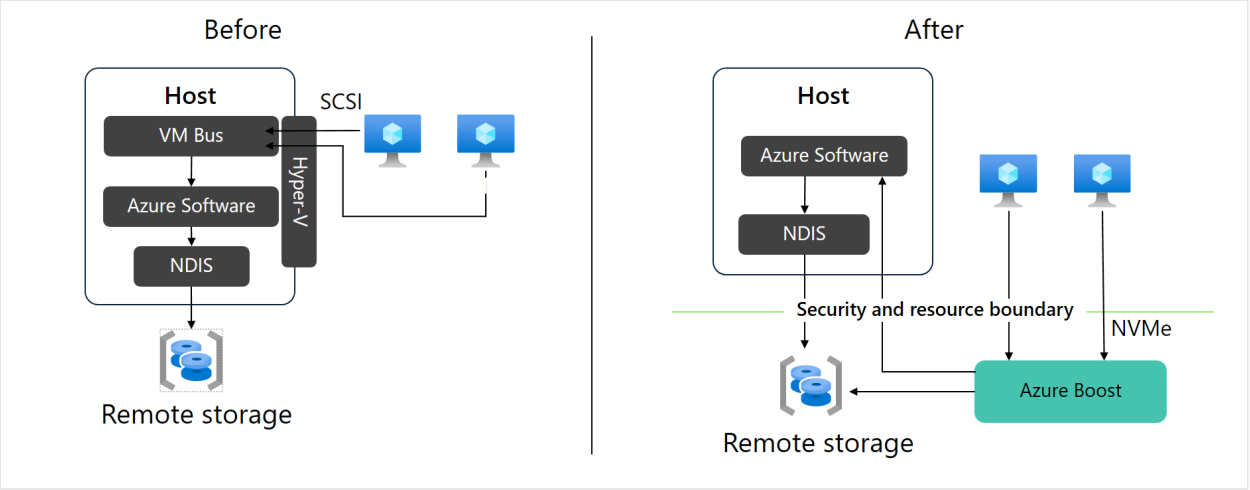


Storage

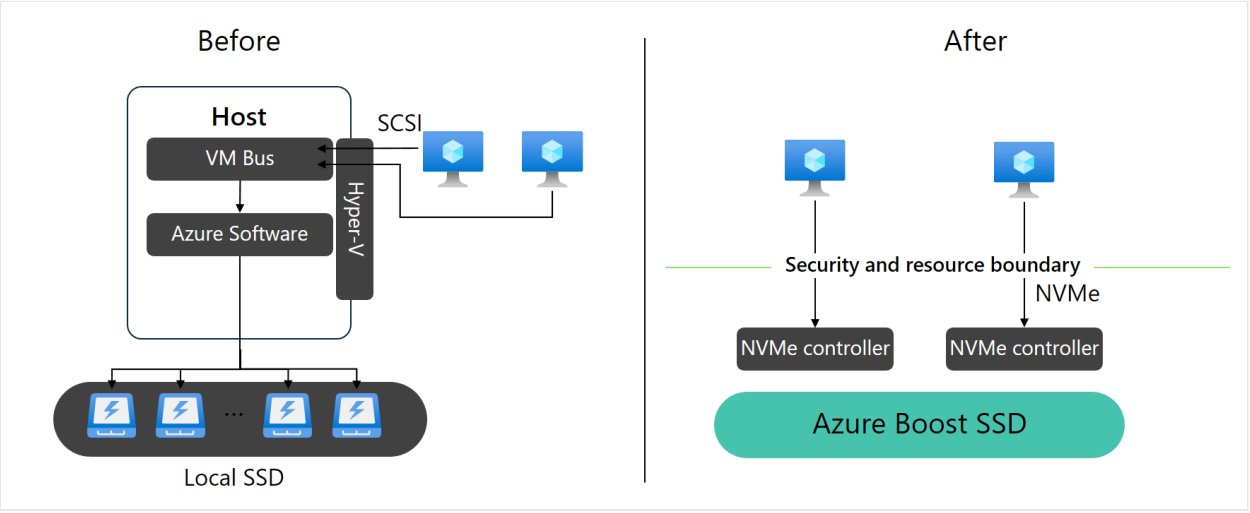
Azure Boost architecture offloads storage covering local, remote and cached disks that provide leading efficiency and performance while improving security, reducing jitter & improving latency for workloads. Azure Boost already provides acceleration for workloads in the fleet using remote storage including specialized workloads such as the Ebsv5 VM types. Also, these improvements provide potential cost saving for customers by consolidating existing workload into fewer or smaller sized VMs.

Azure Boost delivers industry leading throughput performance at up to 14GBps throughput and 750K IOPS for remote disk access performance. This performance is enabled by accelerated storage processing and exposing NVMe disk interfaces to VMs. Storage processing tasks are offloaded from the host processor to dedicated programmable Azure Boost hardware in our dynamically programmable FPGA. This

architecture allows us to update the FPGA hardware in the fleet enabling continuous delivery for our customers.



With Azure Boost SSD architecture, we deliver local, and cached disk performance improvements at up to 36GBps throughput and 6.6M IOPS depending on the VM size of your choice. Azure Boost SSDs are designed to provide high performance optimized encryption at rest, and minimal jitter to NVMe local disks for Azure VMs with local disks.



Security

Azure Boost's security contains several components that work together to provide a secure environment for your virtual machines. Microsoft's in-house developed hardware and software systems provide a secure foundation for your cloud workloads.

- **Security chip:** Boost employs the Cerberus chip as an independent hardware root of trust to achieve NIST 800-193 certification. Customer workloads can't run on Azure Boost powered architecture unless the firmware and software running on the system garners trust.

- **Attestation:** HW RoT identity, Secure Boot, and Attestation through Azure's Attestation Service ensures that Boost and its powered hosts always operate in a healthy and trusted state. Any machine that can't be securely attested is prevented from hosting workloads and it's restored to a trusted state offline.
- **Code integrity:** Boost systems embrace multiple layers of defense-in-depth, including ubiquitous code integrity verification that enforces only Microsoft approved and signed code runs on the Boost system on chip. Microsoft has sought to learn from and contribute back to the wider security community, up streaming advancements to the Integrity Measurement Architecture.
- **Security Enhanced OS:** Azure Boost uses Security Enhanced Linux (SELinux) to enforce principle of least privilege for all software running on its system on chip. All control plane and data plane software running on top of the Boost OS is restricted to running only with the minimum set of privileges required to operate – the operating system restricts any attempt by Boost software to act in an unexpected manner. Boost OS properties make it difficult to compromise code, data, or the availability of Boost and Azure hosting Infrastructure.
- **Rust memory safety:** Rust serves as the primary language for all new code written on the Boost system, to provide memory safety without impacting performance. Control and data plane operations are isolated with memory safety improvements that enhance Azure's ability to keep tenants safe.
- **FIPS certification:** Boost employs a FIPS 140 certified system kernel, providing reliable and robust security validation of cryptographic modules.

Performance

The hardware running virtual machines are a shared resource. The hypervisor (host system) must perform several tasks to ensure that each virtual machine is both isolated from other virtual machines and that each virtual machine receives the resources it needs to run. These tasks include networking between the physical and virtual networks, security, and storage management. Azure Boost reduces the overhead of these tasks by offloading them to dedicated hardware. This offloading frees up CPU resources for the guest virtual machines, resulting in improved performance.

- **VMs using large sizes:** Large sizes that encompass most of a host's resources benefit from Azure Boost. While a large VM size running on a Boost-enabled host might not directly see extra resources, workloads and applications that stress the host processes replaced by Azure Boost see a performance increase.

- **Dedicated hosts:** Performance improvements also have significant impact to Azure Dedicated Hosts (ADH) users. Azure Boost-enabled hosts can potentially run extra, small VMs or increase the size of existing VMs. This allows you to do more work on a single host, reducing your overall costs.

Current availability

Azure Boost is currently available on several VM size families:

Expand table

Size Series	Series Type	Deployment Status
Mbsv3	Memory Optimized	Preview
Mbdsv3	Memory Optimized	Preview
Easv6	Memory Optimized	Production
Eadsv6	Memory Optimized	Production
Epdsv6	Memory Optimized	Production
Epsv6	Memory Optimized	Production
ECesv5/ECedsv5	Memory Optimized	Preview
Dsv6	General Purpose	Production
Dldsv6	General Purpose	Production
Ddsv6	General Purpose	Production
DCesv5	General Purpose	Preview
DCedsv5	General Purpose	Preview
Dasv6	General Purpose	Production
Dalsv6	General Purpose	Production
Daldsv6	General Purpose	Production
Dadsv6	General Purpose	Production
Dpsv6	General Purpose	Production
Dplsv6	General Purpose	Production
Ddsv6	General Purpose	Production

Size Series	Series Type	Deployment Status
Dlsv6	General Purpose	Production
Dpdsv6	General Purpose	Production
Dpldsv6	General Purpose	Production
Nvadsv5	GPU/AI workload optimized	Production
Msv3	Memory Optimized	Production
Mdsv3	Memory Optimized	Production
Msv3	High Memory Optimized	Production
Mdsv3	High Memory Optimized	Production
Msv2	Memory Optimized	Production
Lsv3	Storage Optimized	Production
HX	High Performance Compute	Production
HBv4	High Performance Compute	Production
Fasv6	Compute Optimized	Production
Falsv6	Compute Optimized	Production
Famsv6	Compute Optimized	Production
Ev5	Memory Optimized	Production
Esv6	Memory Optimized	Production
Esv5	Memory Optimized	Production
Epsv5	Memory Optimized	Production
Epdsv5	Memory Optimized	Production
Edv5	Memory Optimized	Production
Edsv6	Memory Optimized	Production
Edsv5	Memory Optimized	Production
Ebsv5	Memory Optimized	Production
Ebdsv5	Memory Optimized	Production
Dv5	General Purpose	Production

Size Series	Series Type	Deployment Status
Dsv5	General Purpose	Production
Dpsv5	General Purpose	Production
Dplsv5	General Purpose	Production
Dpldsv5	General Purpose	Production
Dpdsv5	General Purpose	Production
Dlsv5	General Purpose	Production
Dldsv5	General Purpose	Production
Ddv5	General Purpose	Production
Ddsv5	General Purpose	Production
DCdsv3	General Purpose	Production
Bsv2	General Purpose	Production
Bpsv2	General Purpose	Production

Next Steps

- Learn more about Azure Virtual Network.
- Look into Azure Dedicated Hosts.
- Learn more about Azure Storage.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Hybrid Benefit for Linux virtual machines

08/21/2025

You can use Azure Hybrid Benefit for Linux to easily switch the software subscription model for your Linux virtual machine (VM). Change your subscription model without redeploying your VM and with no risk of downtime. You gain flexibility and savings.

You can switch seamlessly between two subscription models on Azure by using Azure Hybrid Benefit:

- **Bring your own subscription (BYOS)**: In the BYOS model, you bring your own Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES) subscription directly to Azure. You pay only for the infrastructure costs of your VM on Azure. The software fee is covered by your RHEL or SLES subscription you bought from the vendor.
- **Pay-as-you-go (PAYG)**: Use the PAYG subscription model in Azure to pay for RHEL and SLES subscriptions as you use them.

This article defines the BYOS and PAYG subscription models, compares the benefits of each model, and shows you how to use the Azure Hybrid Benefit to switch between the two subscription models for your Linux VMs on Azure.

This process applies to:

- Azure Virtual Machine Scale Sets
- Azure Spot Virtual Machines
- Custom images

Azure Hybrid Benefit gives you the option to make seamless bidirectional conversions between the two subscription models on eligible VM instances.

You might see combined savings estimated to up to 76% with Azure Hybrid Benefit for Linux and three-year Azure Reserved VM Instances. Savings estimates are based on one standard D2s v5 Azure VM with an RHEL or SLES subscription in the East US region running at a PAYG rate versus a reduced rate for a three-year reserved instance plan. The savings estimates are based on Azure pricing as of September 2024. Prices are subject to change. Actual savings might vary based on location, instance type, or usage.

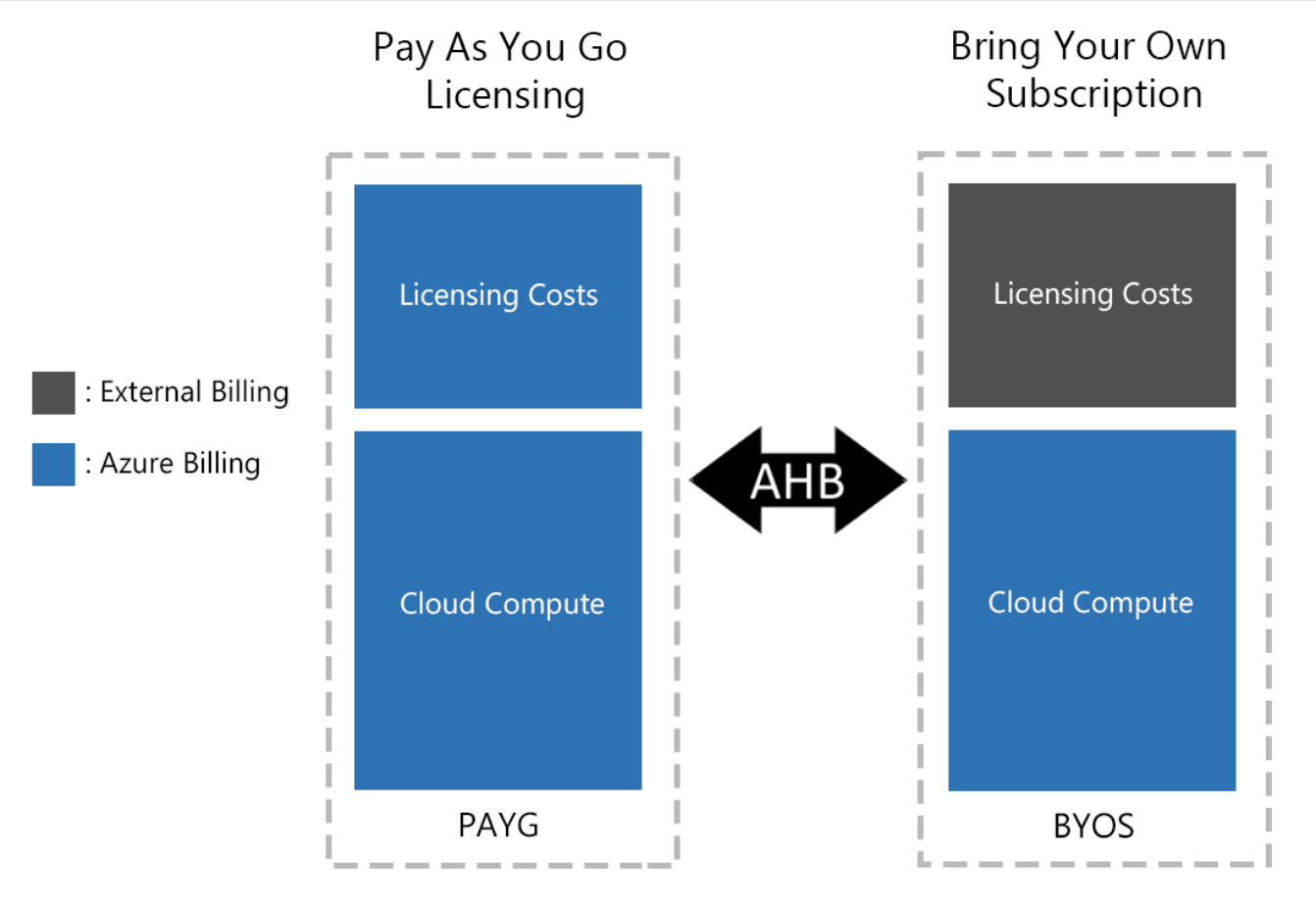
> **Tip**

Try the Azure Pricing calculator to visualize the cost-saving benefits of this feature at a Virtual Machine level.

PAYG vs. BYOS

Azure offers two main licensing pricing options: PAYG and BYOS. With PAYG, you pay only for the resources you use. You can scale up or scale down as needed.

With BYOS, you can use your existing license subscriptions. You don't pay the license fees in Azure.



VMs utilizing a PAYG subscription model, whether deployed from PAYG images or VMs converted to PAYG from BYOS, incur *both* an infrastructure fee and a software fee.

If you have an existing software subscription, use Azure Hybrid Benefit to convert from a PAYG model to a BYOS model.

You can use Azure Hybrid Benefit to switch between the two subscription options at any time.

Linux VMs you can use with Azure Hybrid Benefit

PAYG

The following PAYG RHEL and SLES Marketplace offers are eligible to use with Azure Hybrid Benefit:

RHEL PAYG

Limitations

Only RHEL images *published by Red Hat, Inc.* are eligible for the PAYG to BYOS conversion using the Azure Hybrid Benefit. Images from other vendors aren't supported.

Red Hat-published RHEL PAYG offers eligible for Azure Hybrid Benefit

The following Red Hat-published RHEL PAYG offers are eligible to use with the Azure Hybrid Benefit. Links to the offers in Azure Marketplace are included.

Within these offers, associated images are described as "Pay-As-You-Go".

- Red Hat Enterprise Linux (RHEL)
- Red Hat Enterprise Linux (RHEL)
- Red Hat Enterprise Linux (RHEL) for SAP Apps
- RHEL for SAP with HA and Update Services
- Red Hat Enterprise Linux (RHEL) with High Availability (HA) Add-On
- Red Hat Enterprise Linux (RHEL) Arm64
- Red Hat Enterprise Linux Confidential VM
- Red Hat Enterprise Linux (RHEL) RAW

BYOS

Azure Hybrid Benefit is also available for RHEL and SLES BYOS Azure Marketplace images and images brought from on-prem or other cloud providers.

Currently, one RHEL BYOS offer is available. This offer is a private listing. To gain access to this private listing, you must join the Red Hat Cloud Access program.

You can identify SUSE BYOS marketplace offers by their name, which includes "BYOS". An example is the ** SUSE Linux Enterprise Server 15 SP7 - BYOS** offering.

Azure dedicated host instances and SQL hybrid benefits aren't eligible to use with Azure Hybrid Benefit if you already use Azure Hybrid Benefit with Linux VMs.

> **Note**
>
> For Red Hat VMs, the Azure account must be part of the Red Hat Cloud Access program. Register with Red Hat Cloud Access before you try to enable Azure Hybrid Benefit on your VMs.

Enable Azure Hybrid Benefit

You can enable Azure Hybrid Benefit on new VMs, on existing VMs, and on multiple VMs.

New VM

You can invoke Azure Hybrid Benefit when you create a VM. The benefits of using this approach include:

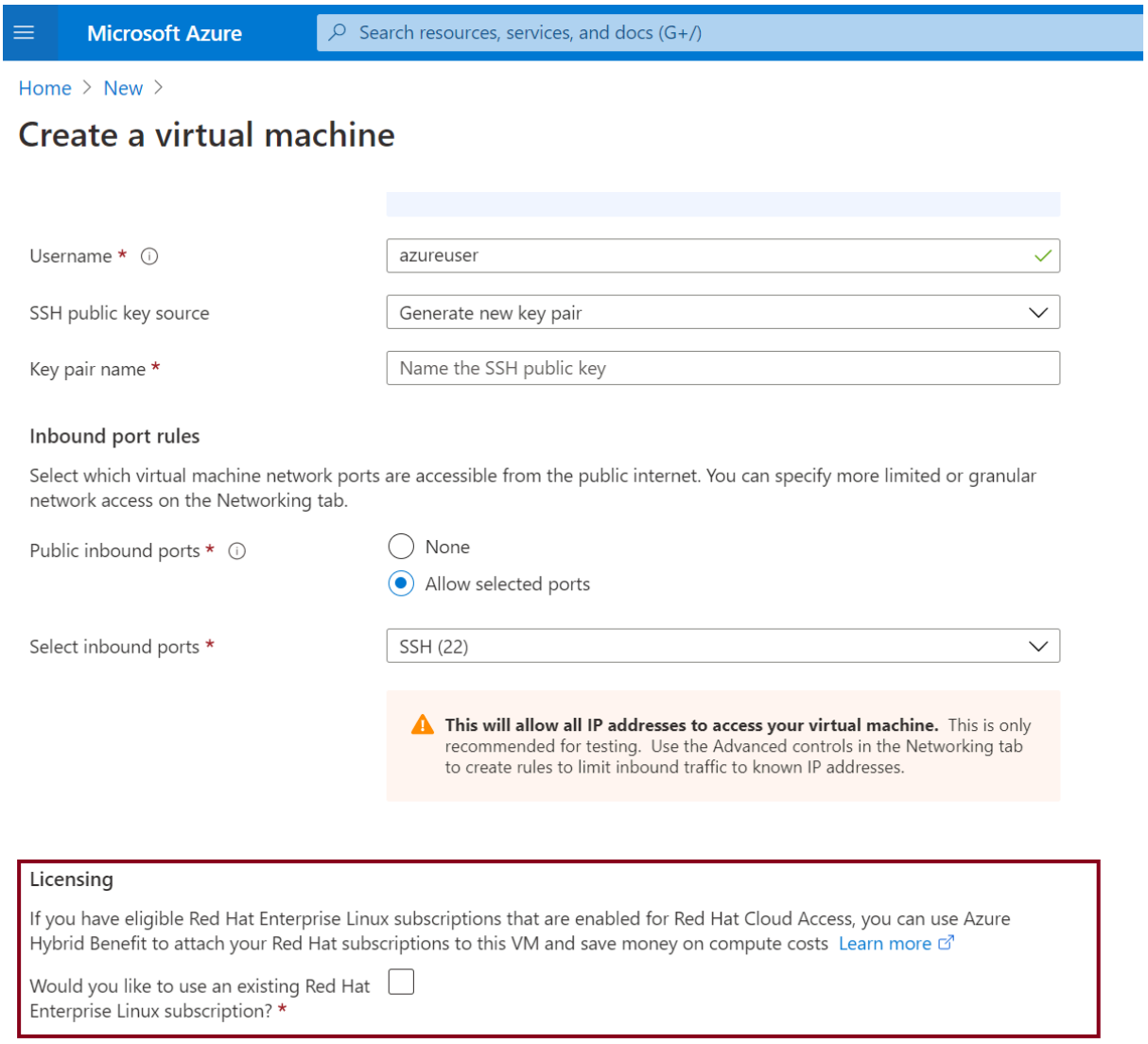
- You can provision both PAYG and BYOS VMs by using the same image and process.
- You can change the licensing mode in the future.
- The PAYG VM is connected to Red Hat Update Infrastructure (RHUI) or SUSE Cloud Update Infrastructure (CUI) by default, to help keep it up to date and secure. You can change the update method after deployment.

Azure portal

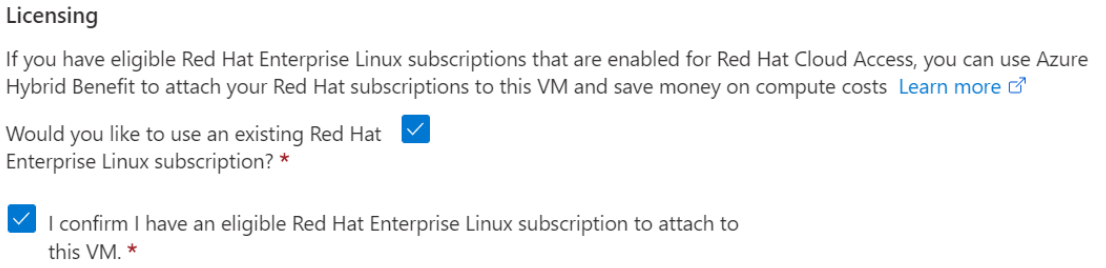
The SUSE workflow is the same as the RHEL example shown here.

To enable Azure Hybrid Benefit when you create a VM:

1. In the Azure portal, go to **Create a virtual machine**.

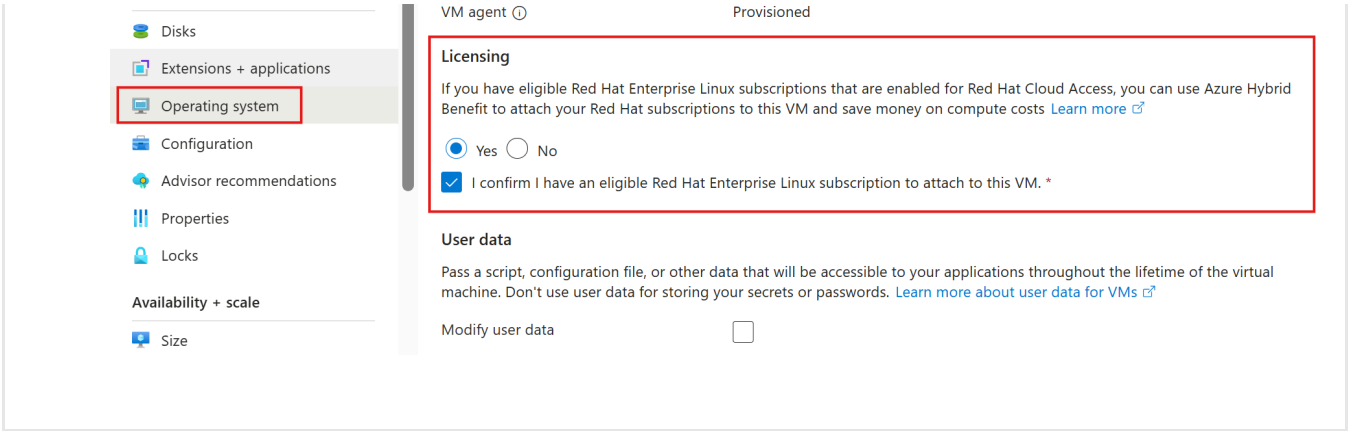


2. In the **Licensing** section, select the checkbox that asks if you want to use an existing RHEL subscription. Select the checkbox to confirm that your subscription is eligible.



3. Create a virtual machine by following the steps that are presented.

4. On the VM service menu, select **Operating System**. Under **Licensing**, verify that the option is enabled.



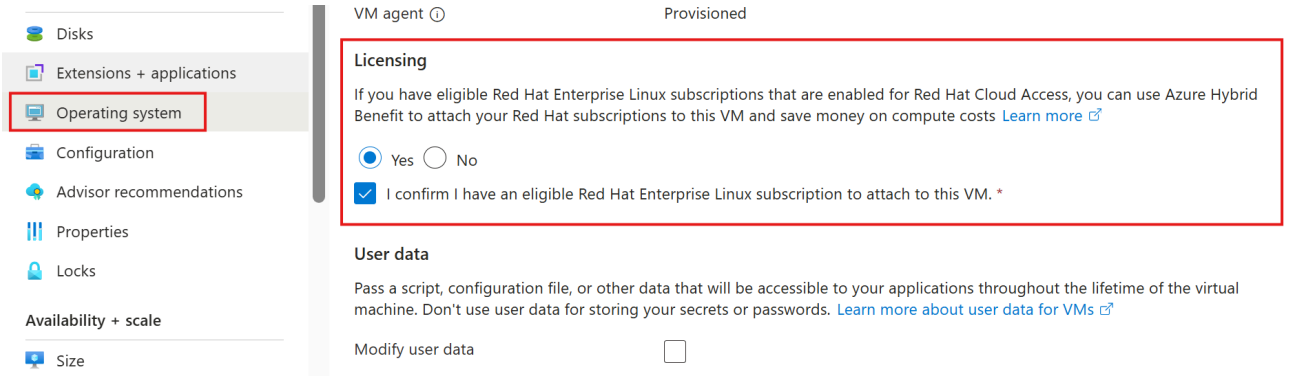
Existing VM

You can enable Azure Hybrid Benefit on an existing VM.

Azure portal

To enable Azure Hybrid Benefit on an existing VM:

1. In the Azure portal, go to the overview pane of the VM that you want to convert.
2. Go to **Operating System** > **Licensing**. To enable the Azure Hybrid Benefit conversion, select **Yes**, and then select the confirmation checkbox.



Check the current licensing model of a VM that has Azure Hybrid Benefit enabled

With Red Hat, the Azure Hybrid Benefit extension must be installed on the VM to switch the licensing model from BYOS to PAYG or vice versa. You can view whether the agent is installed by using the Azure CLI or the Azure Instance Metadata Service.

For SUSE you do not need the extension, the functionality is covered by the SUSE tooling itself.

Azure CLI

1. Red Hat: You can use the `az vm get-instance-view` command to check whether the extension is installed. Look for the `AHBForRHEL` extension. If the corresponding extension is installed, the Azure Hybrid Benefit is enabled. Review the license type to determine which licensing model is applied to your VM.

   Azure CLI

   ```
   az vm get-instance-view -g MyResourceGroup -n myVm --query
   instanceView.extensions
   ```

2. Red Hat and SUSE: Use the following command to review the license type that's applied to the VM:

   Azure CLI

   ```
   az vm get-instance-view -g MyResourceGroup -n myVM --query licenseType
   ```

   The following license types correspond to a **PAYG** subscription model:

   For RHEL:

   - RHEL_BASE
   - RHEL_EUS
   - RHEL_SAPAPPS
   - RHEL_SAPHA
   - RHEL_BASESAPAPPS
   - RHEL_BASESAPHA

   For SLES:

   - SLES
   - SLES_SAP
   - SLES_HPC

   These license types correspond to a **BYOS** subscription model:

   For RHEL:

   - RHEL_BYOS

   For SLES:

   - SLES_BYOS

If the license type of the VM isn't modified, this command returns an empty string and the VM continues to use the billing model of the image that you used to deploy it.

Convert PAYG to BYOS

If you deployed an Azure Marketplace image by using a PAYG licensing model and want to convert licensing to BYOS, complete the following steps.

Convert a PAYG image to BYOS by using the Azure CLI

Use the Azure CLI to convert a PAYG Azure Marketplace image to a BYOS subscription model.

Red Hat (RHEL)

1. Apply the `RHEL_BYOS` license type to the VM:

   Azure CLI

   ```
   # This enables BYOS on a RHEL PAYG VM by using Azure Hybrid Benefit.
   az vm update -g myResourceGroup -n myVmName --license-type RHEL_BYOS
   ```

2. When the PAYG to BYOS conversion is finished, you must register the VM with Red Hat for system updates and usage compliance.

3. If you want to return the original subscription model, set `license-type` to `None`.

   Azure CLI

   ```
   # In order to revert back to the original licensing model, set license-
   type to None.
   az vm update -g myResourceGroup -n myVmName --license-type NONE
   ```

Convert the license type in the VM operating system

Red Hat (RHEL)

To start using Azure Hybrid Benefit for Red Hat:

1. Install the `AHBForRHEL` extension on the VM on which you want to apply the Azure Hybrid Benefit BYOS benefit. You can install the extension by using the Azure CLI or Azure PowerShell.

2. Depending on the software updates that you want, change the license type to a relevant value. Here are the available license type values and the software updates associated with them:

Expand table

License type	Software updates	Allowed VMs
RHEL_BASE	Installs Red Hat regular/base repositories on your VM.	RHEL BYOS VMs, RHEL custom image VMs
RHEL_EUS	Installs Red Hat Extended Update Support (EUS) repositories on your VM.	RHEL BYOS VMs, RHEL custom image VMs
RHEL_SAPAPPS	Installs RHEL for SAP Business Apps repositories on your VM.	RHEL BYOS VMs, RHEL custom image VMs
RHEL_SAPHA	Installs RHEL for SAP with High Availability (HA) repositories on your VM.	RHEL BYOS VMs, RHEL custom image VMs
RHEL_BASESAPAPPS	Installs RHEL regular/base SAP Business Apps repositories on your VM.	RHEL BYOS VMs, RHEL custom image VMs
RHEL_BASESAPHA	Installs regular/base RHEL for SAP with HA repositories on your VM.	RHEL BYOS VMs, RHEL custom image VMs

3. Wait one hour for the extension to read the license type value and install the repositories.

> **Note**
>
> If the extension isn't running by itself, you can run it on demand.

You should now be connected to Azure Red Hat Update. The relevant repositories are installed on your machine.

1. If you want to switch back to the BYOS model, set `license-type` to `None` and run the extension. This action removes all RHUI repositories from your VM and stops associated billing.

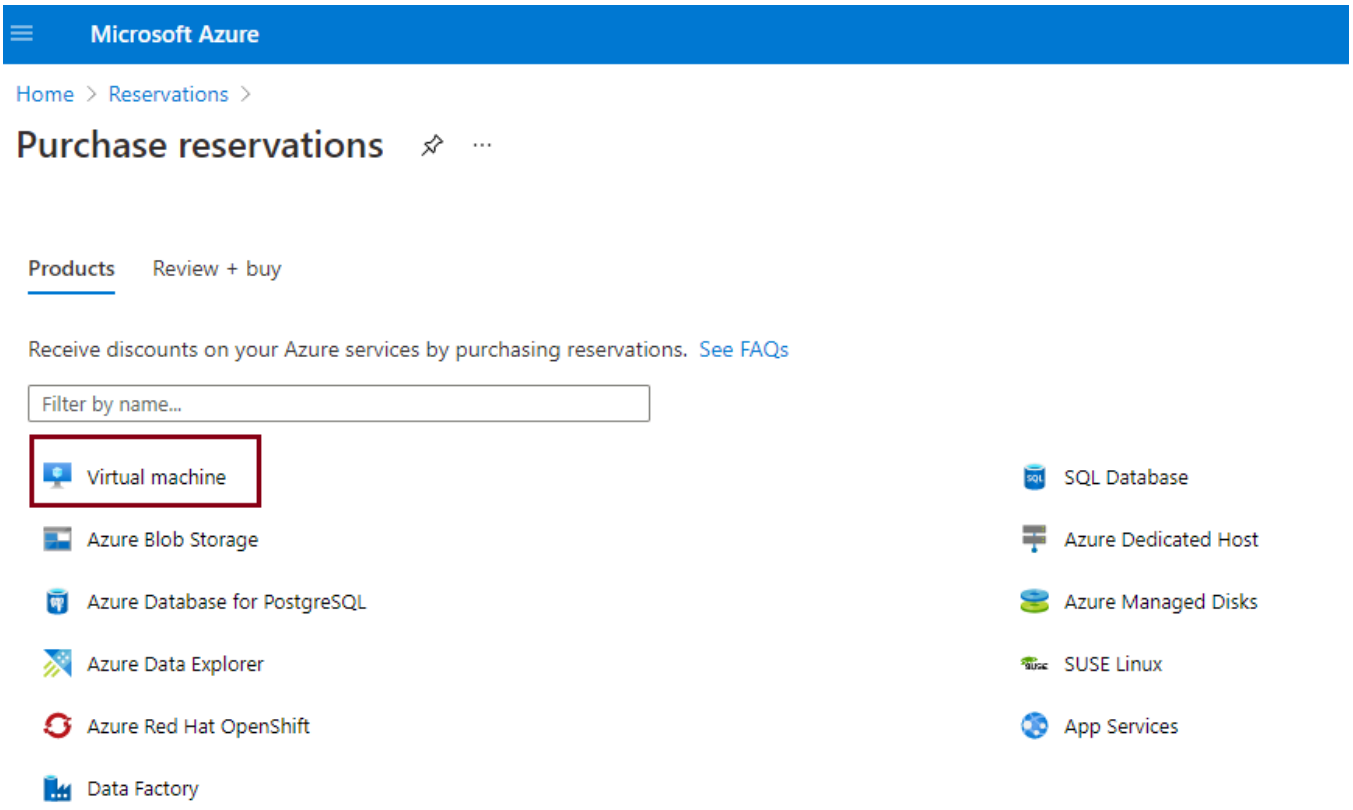
> **Note**

In the unlikely event that the extension can't install repositories or if there are any other issues, switch the license type back to empty and contact Microsoft support. Taking this step ensures that you aren't billed for software updates.

Azure Hybrid Benefit for reserved instance VMs

Azure reservations (Azure Reserved Virtual Machine Instances) help you save money by committing to one-year or three-year plans for multiple products. Azure Hybrid Benefit for PAYG VMs is available for reserved instances.

If you purchased compute costs at a discounted rate by using reserved instances, you can apply Azure Hybrid Benefit on the licensing costs for RHEL and SUSE. The steps to apply Azure Hybrid Benefit for a reserved instance are exactly the same as for a regular VM.



> **Note**
>
> If you already purchased reservations for RHEL or SUSE PAYG software in Azure Marketplace, wait for the reservation tenure to finish before you use Azure Hybrid Benefit for PAYG VMs.

Compliance

Red Hat compliance

Customers who use Azure Hybrid Benefit for PAYG RHEL VMs agree to the standard legal terms and privacy statement associated with Azure Marketplace RHEL offers.

Customers who use Azure Hybrid Benefit for PAYG RHEL VMs have three options for providing software updates and patches to those VMs:

- Red Hat Update Infrastructure (default option)
- Red Hat Satellite Server
- Red Hat Subscription Manager

Customers can use RHUI as the main update source for Azure Hybrid Benefit for PAYG RHEL VMs without attaching subscriptions. Customers who choose the RHUI option are responsible for ensuring RHEL subscription compliance.

Customers using either Red Hat Satellite Server or Red Hat Subscription Manager should remove the RHUI configuration and attach a cloud-access-enabled RHEL subscription to Azure Hybrid Benefit for PAYG RHEL VMs.

For more information about Red Hat subscription compliance, software updates, and sources for Azure Hybrid Benefit for PAYG RHEL VMs, see the Red Hat article about using RHEL subscriptions with Azure Hybrid Benefit.

Customers who use the Azure Hybrid Benefit BYOS to PAYG capability for RHEL agree to the standard legal terms and privacy statement associated with Azure Marketplace RHEL offerings.

Frequently asked questions

- **Q: Can I use a license type of `RHEL_BYOS` with a SLES image or vice versa?**
  - A: No, you can't. Trying to enter a license type that incorrectly matches the distribution running on your VM won't update any billing metadata. But if you accidentally enter the wrong license type, updating your VM again to the correct license type still enables Azure Hybrid Benefit.

- **Q: I've registered with Red Hat Cloud Access but still can't enable Azure Hybrid Benefit on my RHEL VMs. What should I do?**
  - A: It might take some time for your Red Hat Cloud Access subscription registration to propagate from Red Hat to Azure. If you still see the error after one business day, contact Microsoft support.

- **Q: I've deployed a VM by using a RHEL BYOS "golden image." Can I convert the billing on this image from BYOS to PAYG?**
  - A: Yes, you can use Azure Hybrid Benefit for BYOS VMs to convert this subscription model.

- **Q: I've uploaded my own RHEL or SLES image from on-premises (via Azure Migrate, Azure Site Recovery, or otherwise) to Azure. Can I convert the billing on these images from BYOS to PAYG?**
  - A: Yes, you can use Azure Hybrid Benefit for BYOS VMs to convert this subscription model.

- **Q: I've uploaded my own RHEL or SLES image from on-premises (via Azure Migrate, Azure Site Recovery, or otherwise) to Azure. Do I need to do anything to benefit from Azure Hybrid Benefit?**
  - A: No, you don't. RHEL or SLES images that you upload are already considered BYOS, and you're charged only for Azure infrastructure costs. You're responsible for RHEL or SLES subscription costs, just as you are for your on-premises environments.

- **Q: Can I use Azure Hybrid Benefit for PAYG VMs for Azure Marketplace RHEL and SLES SAP images?**
  - A: Yes. You can use the license type of RHEL_BYOS for RHEL VMs and SLES_BYOS for conversions of VMs deployed from Azure Marketplace RHEL and SLES SAP images.

- **Q: Can I use Azure Hybrid Benefit for PAYG VMs on virtual machine scale sets for RHEL and SLES?**
  - A: Yes. Azure Hybrid Benefit on virtual machine scale sets for RHEL and SLES is available to all users. You can learn more about this benefit and how to use it.

- **Q: Can I use Azure Hybrid Benefit for PAYG VMs on reserved instances for RHEL and SLES?**
  - A: Yes. Azure Hybrid Benefit for PAYG VMs on reserved instances for RHEL and SLES is available to all users.

- **Q: Can I use Azure Hybrid Benefit for PAYG VMs on a VM deployed for SQL Server on RHEL images?**
  - A: No, you can't. There's no plan to support these VMs.

- **Q: Can I use Azure Hybrid Benefit on my RHEL for Virtual Datacenters subscription?**
  - A: No. RHEL for Virtual Datacenters isn't supported on Azure at all, including for Azure Hybrid Benefit.

Related content

- Learn how to create and update VMs and add license types (RHEL_BYOS, SLES_BYOS) for Azure Hybrid Benefit by using the Azure CLI
- Learn about Azure Hybrid Benefit on virtual machine scale sets for RHEL and SLES and how to use it

Explore Azure Hybrid Benefit for Windows VMs

Article • 01/02/2025

Maximize your on-premises core licenses for Windows Server to get Windows virtual machines (VMs) on Azure at a reduced cost through Azure Hybrid Benefit for Windows Server. You also can use Azure Hybrid Benefit for Windows Server to deploy new VMs that run the Windows OS.

This article describes the steps to deploy new VMs with Azure Hybrid Benefit for Windows Server and how to update existing running VMs.

To qualify for Azure Hybrid Benefit for Windows Server, you need on-premises core licenses for Windows Server from an applicable program with active Software Assurance or qualifying subscription licenses. Software Assurance and qualifying subscription licenses are available only as part of certain commercial licensing agreements. To learn more about commercial licensing, see Microsoft licensing resources. To learn more about Windows Server core licenses, see Windows Server product licensing.

You can use Azure Hybrid Benefit for Windows Server with any VMs running Windows Server OS in all regions, including VMs that have additional software, such as SQL Server or third-party Azure Marketplace software.

Limitations

To use Azure Hybrid Benefit for Windows Server, you must have a minimum of 8 core licenses (Datacenter edition or Standard edition) per VM. For example, even if you run a 4-core instance, 8 core licenses are required. You also can run instances larger than 8 cores by allocating licenses equal to the core size of the instance. For example, 12 core licenses are required for a 12-core instance.

For customers who have processor licenses, each processor license is equivalent to 16 core licenses.

ⓘ Important

- Workloads that use Azure Hybrid Benefit for Windows Server can run only during the Software Assurance or subscription license term. When the Software Assurance or subscription license term approaches expiration, you must renew your agreement with either Software Assurance or a subscription

license, disable the Azure Hybrid Benefit for Windows Server functionality, or deprovision workloads that use Azure Hybrid Benefit for Windows Server.

- The Microsoft Product Terms for your program take precedence over the information that's presented in this article. For more information, see Microsoft Azure Product Terms and select your program to show the terms.

Classic VMs

For classic VMs, the only supported option is deploying a new VM from an on-premises custom image. To take advantage of the Azure Hybrid Benefit for Windows Server capabilities that this article describes, you must first migrate classic VMs to Azure Resource Manager model VMs.

> **Important**
>
> VMs created through the classic deployment model will be retired on September 1, 2023.
>
> If you use IaaS resources from Azure Service Management, please complete your migration by September 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.
>
> For more information, see Migrate your IaaS resources to Azure Resource Manager by September 1, 2023.

How to use Azure Hybrid Benefit for Windows Server

You have several options to use Windows virtual machines with Azure Hybrid Benefit for Windows Server. You can:

- Deploy VMs from one of the provided Windows Server images on Azure Marketplace.
- Upload a custom VM and deploy by using an Azure Resource Manager template or Azure PowerShell.
- Toggle and convert an existing VM between from running with Azure Hybrid Benefit or the pay-on-demand cost for Windows Server.

- Apply Azure Hybrid Benefit for Windows Server on a virtual machine scale set.

Create a VM that uses Azure Hybrid Benefit for Windows Server

All Windows Server OS-based images are supported for Azure Hybrid Benefit for Windows Server. You can use Azure platform-supported images or upload your own custom Windows Server image.

Azure portal

To create a VM that uses Azure Hybrid Benefit for Windows Server, when you create your VM, on the **Basics** tab under **Licensing**, select the checkbox to use an existing Windows Server license.

Azure PowerShell

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName "myResourceGroup" `
    -Name "myVM" `
    -Location "East US" `
    -ImageName "Win2016Datacenter" `
    -LicenseType "Windows_Server"
```

Azure CLI

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --location eastus \
    --license-type Windows_Server
```

Resource Manager template

In your Resource Manager template, set the `licenseType` parameter. For more information, see Authoring Azure Resource Manager templates.

JSON

```
"properties": {
    "licenseType": "Windows_Server",
    "hardwareProfile": {
        "vmSize": "[variables('vmSize')]"
    }
}
```

Convert an existing VM to use Azure Hybrid Benefit for Windows Server

To convert an existing VM to use Azure Hybrid Benefit for Windows Server, update your VM's license type.

> ⓘ **Note**
>
> Changing the license type on the VM doesn't cause the system to restart, and service is not interrupted. The process changes a metadata licensing flag only.

Azure portal

On the VM service menu, select **Configuration**, and then set **Azure Hybrid Benefit** to **Enable**.

Azure PowerShell

- To convert an existing Windows Server VM to Azure Hybrid Benefit for Windows Server:

  Azure PowerShell

  ```
  $vm = Get-AzVM -ResourceGroup "rg-name" -Name "vm-name"
  $vm.LicenseType = "Windows_Server"
  Update-AzVM -ResourceGroupName rg-name -VM $vm
  ```

- To convert a Windows Server VM that uses Azure Hybrid Benefit for Windows Server back to pay-as-you-go:

  Azure PowerShell

```
$vm = Get-AzVM -ResourceGroup "rg-name" -Name "vm-name"
$vm.LicenseType = "None"
Update-AzVM -ResourceGroupName rg-name -VM $vm
```

Azure CLI

- To convert an existing Windows Server VM to use Azure Hybrid Benefit for Windows Server:

Azure CLI

```
az vm update --resource-group myResourceGroup --name myVM --set
licenseType=Windows_Server
```

Verify that your VM uses the licensing benefit

After you deploy your VM by using either Azure PowerShell, a Resource Manager template, or the Azure portal, you can verify the setting by using one of the following methods.

Azure portal

On the VM service menu, select **Operating system**, and then view the Azure Hybrid Benefit for Windows Server setting.

Azure PowerShell

The following example shows the license type for a single VM:

Azure PowerShell

```
Get-AzVM -ResourceGroup "myResourceGroup" -Name "myVM"
```

Output:

Azure PowerShell

```
Type                    : Microsoft.Compute/virtualMachines
Location                : westus
LicenseType             : Windows_Server
```

The output contrasts with the following VM that's deployed *without* Azure Hybrid Benefit for Windows Server licensing:

Azure PowerShell

```
Type                    : Microsoft.Compute/virtualMachines
Location                : westus
LicenseType             :
```

Azure CLI

Azure CLI

```
az vm get-instance-view -g MyResourceGroup -n MyVM --query "[?
licenseType=='Windows_Server']" -o table
```

> **Note**
>
> Changing the license type on the VM doesn't cause the system to restart, and service is not interrupted. The process changes a metadata licensing flag only.

List all resources that use Azure Hybrid Benefit for Windows Server

To view and get a count of all your VMs and virtual machine scale sets that have Azure Hybrid Benefit for Windows Server enabled, you can use the following options for your subscription.

Azure portal

On the VM or virtual machine scale sets overview pane, get a list of all your VMs and licensing types by setting the table columns to include **OS licensing benefit**. The VM might have the state **Azure Hybrid Benefit for Windows**, **Not enabled**, or **Windows client with multi-tenant hosting**.

Azure PowerShell

For VMs:

Azure PowerShell

```
Get-AzVM | ?{$_.LicenseType -like "Windows_Server"} | select
ResourceGroupName, Name, LicenseType
```

For virtual machine scale sets:

Azure PowerShell

```
Get-AzVmss | Select * -ExpandProperty VirtualMachineProfile | ? LicenseType
-eq 'Windows_Server' | select ResourceGroupName, Name, LicenseType
```

Azure CLI

For VMs:

Azure CLI

```
az vm list --query "[?licenseType=='Windows_Server']" -o table
```

For virtual machine scale sets:

Azure CLI

```
az vmss list --query "[?
virtualMachineProfile.licenseType=='Windows_Server']" -o table
```

Deploy a virtual machine scale set to use Azure Hybrid Benefit for Windows Server

Within your virtual machine scale set Resource Manager templates, the `licenseType` parameter must be set in your `VirtualMachineProfile` property. You can set this parameter when you create or update for your virtual machine scale set by using a Resource Manager template, Azure PowerShell, the Azure CLI, or REST API.

The following example uses a Resource Manager template with a Windows Server 2016 Datacenter image:

JSON

```
"virtualMachineProfile": {
    "storageProfile": {
        "osDisk": {
            "createOption": "FromImage"
```

```
            },
            "imageReference": {
                "publisher": "MicrosoftWindowsServer",
                "offer": "WindowsServer",
                "sku": "2016-Datacenter",
                "version": "latest"
            }
        },
        "licenseType": "Windows_Server",
        "osProfile": {
                "computerNamePrefix": "[parameters('vmssName')]",
                "adminUsername": "[parameters('adminUsername')]",
                "adminPassword": "[parameters('adminPassword')]"
        }
}
```

For more information, see Modify a virtual machine scale set.

Related content

- Save money with Azure Hybrid Benefit
- FAQ for Azure Hybrid Benefit
- Azure Hybrid Benefit for Windows Server licensing detailed guidance
- Azure Hybrid Benefit for Windows Server and Azure Site Recovery make migrating applications to Azure even more cost-effective
- Deploy Windows 11 on Azure with Multitenant Hosting Rights
- Using Resource Manager templates

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What are Azure Reservations?

Azure Reservations help you save money by committing to one-year or three-year plans for multiple products. Committing allows you to get a discount on the resources you use. Reservations can significantly reduce your resource costs by up to 72% from pay-as-you-go prices. Reservations provide a billing discount and don't affect the runtime state of your resources. After you purchase a reservation, the discount automatically applies to matching resources.

You can pay for a reservation up front or monthly. The total cost of up-front and monthly reservations is the same and you don't pay any extra fees when you choose to pay monthly. Monthly payments are for complete month and not prorated based on calendar month. As an example, if you purchased a reservation on January 15, the next charge will be around February 15.

You can buy a reservation in the Azure portal.

Why buy a reservation?

If you have consistent resource usage that supports reservations, buying a reservation gives you the option to reduce your costs. For example, when you continuously run instances of a service without a reservation, you're charged at pay-as-you-go rates. Once a reservation purchase is complete, you get the reservation discount. The resources are no longer charged at the pay-as-you-go rates.

How reservation discount is applied

After purchase, the reservation discount automatically applies to the resource usage that matches the attributes you select when you buy the reservation. Attributes include the SKU, regions (where applicable), and scope. Reservation scope selects where the reservation savings apply.

For more information about how discount is applied, see Reserved instance discount application.

For more information about how reservation scope works, see Scope reservations.

Determine what to purchase

All reservations, except Azure Databricks, are applied on an hourly basis. Consider reservation purchases based on your consistent base usage. You can determine which reservation to

purchase by analyzing your usage data or by using reservation recommendations. Recommendations are available in:

- Azure Advisor (VMs only)
- Reservation purchase experience in the Azure portal
- Cost Management Power BI app
- APIs

For more information, see Determine what reservation to purchase

Buying a reservation

You can purchase reservations from the Azure portal, APIs, PowerShell, and CLI.

Go to the Azure portal to make a purchase.

For more information, see Buy a reservation.

How is a reservation billed?

The reservation is charged to the payment method tied to the subscription. The reservation cost is deducted from your Azure Prepayment (previously called monetary commitment) balance, if available. When your Azure Prepayment balance doesn't cover the cost of the reservation, you're billed the overage. If you have a subscription from an individual plan with pay-as-you-go rates, the credit card you have on your account is billed immediately for up-front purchases. Monthly payments appear on your invoice and your credit card is charged monthly. When you're billed by invoice, you see the charges on your next invoice.

Who can manage a reservation by default

By default, the following users can view and manage reservations:

- The person who buys a reservation and the account administrator of the billing subscription used to buy the reservation are added to the reservation order.
- Enterprise Agreement and Microsoft Customer Agreement billing administrators.

To allow other people to manage reservations, see Manage Reservations for Azure resources.

Get reservation details and utilization after purchase

If you have permission to view to the reservation, you can see it and its use in the Azure portal. You can get the data using APIs, as well.

For more information on how to see reservations in Azure portal, see View reservations in the Azure portal

Manage reservations after purchase

After you buy an Azure reservation, you can update the scope to apply reservation to a different subscription, change who can manage the reservation, split a reservation into smaller parts, or change instance size flexibility.

For more information, see Manage Reservations for Azure resources

Flexibility with Azure reservations

Azure Reservations provide flexibility to help meet your evolving needs. You can exchange a reservation for another reservation of the same type. You can also refund a reservation, up to $50,000 USD in a 12 month rolling window, if you no longer need it. The maximum limit of the refund applies to all reservations in the scope of your agreement with Microsoft.

For more information, see Self-service exchanges and refunds for Azure Reservations

Charges covered by reservation

- **Reserved Virtual Machine Instance** - A reservation only covers the virtual machine and cloud services compute costs. It doesn't cover additional software, Windows, networking, or storage charges.
- **Azure Blob storage reserved capacity** - A reservation covers storage capacity for Blob storage and Azure Data Lake Gen2 storage. The reservation doesn't cover bandwidth or transaction rates.
- **Azure Files reserved capacity** - A reservation covers storage capacity for Azure Files. Reservations for hot and cool tiers don't cover bandwidth or transaction rates.
- **Azure Cosmos DB reserved capacity** - A reservation covers throughput provisioned for your resources. It doesn't cover the storage and networking charges.
- **Azure Data Factory data flows** - A reservation covers integration runtime cost for the compute type and number of cores that you buy.
- **Azure SQL Database** - A reservation only covers the compute costs. It doesn't cover software license, networking, or storage charges. For the General Purpose service tier, reservations are also available to cover compute costs associated with the zone-redundancy add-on.

- **Azure SQL Managed Instance** - A reservation only covers the compute costs. It doesn't cover software license, networking, or storage charges. Reservations are also available to cover compute costs associated with the zone-redundancy add-on.
- **Azure Synapse Analytics** - A reservation covers cDWU usage. It doesn't cover storage or networking charges associated with the Azure Synapse Analytics usage.
- **Azure Databricks** - A reservation covers only the DBU usage. Other charges, such as compute, storage, and networking, are applied separately.
- **App Service stamp fee** - A reservation covers stamp usage. It doesn't apply to workers, so any other resources associated with the stamp are charged separately.
- **Azure Database for MySQL** - Only the compute costs are included with a reservation. A reservation doesn't cover software, networking, or storage charges associated with the MySQL Database server.
- **Azure Database for PostgreSQL** - Only the compute costs are included with a reservation. A reservation doesn't cover software, networking, or storage charges associated with the PostgreSQL Database servers.
- **Azure Data Explorer** - A reservation covers the markup charges. A reservation doesn't apply to compute, networking, or storage charges associated with the clusters.
- **Azure Cache for Redis and Managed Redis** - Only the compute costs are included with a reservation. A reservation doesn't cover networking or storage charges associated with the Redis cache instances.
- **Azure Dedicated Host** - Only the compute costs are included with the Dedicated host.
- **Azure Disk Storage reservations** - A reservation only covers Premium SSDs of P30 size or greater. It doesn't cover any other disk types or sizes smaller than P30.
- **Azure Backup Storage reserved capacity** - A capacity reservation lowers storage costs of backup data in a Recovery Services Vault.
- **Azure NetApp Files** - A capacity reservation covers matching capacity pools in the selected service level and region. When using capacity pools configured with cool access, only "hot" tier consumption is covered by the reservation benefit.

Software plans:

- **SUSE Linux** - A reservation covers the software plan costs. The discounts apply only to SUSE meters and not to the virtual machine usage.
- **Red Hat Plans** (***plans and renewal are temporarily unavailable***) - A reservation covers the software plan costs. The discounts apply only to RedHat meters and not to the virtual machine usage.
- **Azure Red Hat OpenShift** - A reservation applies to the OpenShift costs, not to Azure infrastructure costs.

For Windows virtual machines and SQL Database, the reservation discount doesn't apply to the software costs. You can cover the licensing costs with Azure Hybrid Benefit.

Need help? Contact us.

If you have questions or need help, create a support request.

Next steps

- Learn more about Azure Reservations with the following articles:
  - Manage Azure Reservations
  - Understand reservation usage for your subscription with pay-as-you-go rates
  - Understand reservation usage for your Enterprise enrollment
  - Windows software costs not included with reservations
  - Azure Reservations in Partner Center Cloud Solution Provider (CSP) program
- Learn more about reservations for service plans:
  - Virtual Machines with Azure Reserved VM Instances
  - Azure Cosmos DB resources with Azure Cosmos DB reserved capacity
  - SQL Database compute resources with Azure SQL Database reserved capacity
  - Azure Cache for Redis resources with Azure Cache for Redis reserved capacity
- Learn more about reservations for software plans:
  - Red Hat software plans from Azure Reservations
  - SUSE software plans from Azure Reservations

Last updated on 03/19/2026

Save costs with Azure Reserved VM Instances

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

When you commit to an Azure reserved VM instance you can save money. The reservation discount is applied automatically to the number of running virtual machines that match the reservation scope and attributes. You don't need to assign a reservation to a virtual machine to get the discounts. A reserved instance purchase covers only the compute part of your VM usage. For Windows VMs, the usage meter is split into two separate meters. There's a compute meter, which is same as the Linux meter, and a Windows server license. The charges that you see when you make the purchase are only for the compute costs. Charges don't include Windows software costs. For more information about software costs, see Software costs not included with Azure Reserved VM Instances.

Reserved VM Instances provide a billing discount only and do not reserve or guarantee compute capacity. Azure still needs available capacity in the target region or availability zone when you deploy a VM. A deployment can fail because of capacity constraints even when quota is approved and you already purchased a Reserved VM Instance. To guarantee capacity for your workloads, use On-demand capacity reservation. You can combine Reserved VM Instances with capacity reservations for both cost savings and capacity assurance.

Determine the right VM size before you buy

Before you buy a reservation, you should determine the size of the VM that you need. The following sections will help you determine the right VM size.

Use reservation recommendations

You can use reservation recommendations to help determine the reservations you should purchase.

- Purchase recommendations and recommended quantities are shown when you purchase a VM reserved instance in the Azure portal.
- Azure Advisor provides purchase recommendations for individual subscriptions.
- You can use the APIs to get purchase recommendations for both shared scope and single subscription scope. For more information, see Reserved instance purchase

recommendation APIs for enterprise customers.

- For Enterprise Agreement (EA) and Microsoft Customer Agreement (MCA) customers, purchase recommendations for shared and single subscription scopes are available with the Azure Consumption Insights Power BI content pack.

Services that get VM reservation discounts

Your VM reservations can apply to VM usage emitted from multiple services - not just for your VM deployments. Resources that get reservation discounts change depending on the instance size flexibility setting.

Instance size flexibility setting

The instance size flexibility setting determines which services get the reserved instance discounts.

Whether the setting is on or off, reservation discounts automatically apply to any matching VM usage when the *ConsumedService* is `Microsoft.Compute`. So, check your usage data for the *ConsumedService* value. Some examples include:

- Virtual machines
- Virtual machine scale sets
- Container service
- Azure Batch deployments (in user subscriptions mode)
- Azure Kubernetes Service (AKS)
- Service Fabric

When the setting is on, reservation discounts automatically apply to matching VM usage when the *ConsumedService* is any of the following items:

- Microsoft.Compute
- Microsoft.ClassicCompute
- Microsoft.Batch
- Microsoft.MachineLearningServices
- Microsoft.Kusto

Check the *ConsumedService* value in your usage data to determine if the usage is eligible for reservation discounts.

For more information about instance size flexibility, see Virtual machine size flexibility with Reserved VM Instances.

Analyze your usage information

Analyze your usage information to help determine which reservations you should purchase. Usage data is available in the usage file and APIs. Use them together to determine which reservation to purchase. Check for VM instances that have high usage on daily basis to determine the quantity of reservations to purchase. Avoid the `Meter` subcategory and `Product` fields in usage data. They don't distinguish between VM sizes that use premium storage. If you use these fields to determine the VM size for reservation purchase, you may buy the wrong size. Then you won't get the reservation discount you expect. Instead, refer to the `AdditionalInfo` field in your usage file or usage API to determine the correct VM size.

Your usage file shows your charges by billing period and daily usage. For information about downloading your usage file, see View and download your Azure usage and charges. Then, by using the usage file information, you can determine what reservation to purchase.

Purchase restriction considerations

Reserved VM Instances are available for most VM sizes with some exceptions. Reservation discounts don't apply for the following VMs:

- **VM series** - A-series, or G-series.

  > ⓘ Note
  >
  > A-series VMs differ from Av2 series VMs. Av2 series VMs are available for reservation purchase.

- **Preview or Promo VMs** - Any VM-series or size that is in preview or uses promotional meter.

- **Insufficient quota** - A reservation that is scoped to a single subscription must have vCPU quota available in the subscription for the new RI. For example, if the target subscription has a quota limit of 10 vCPUs for D-Series, then you can't buy a reservation for 11 Standard_D1 instances. The quota check for reservations includes the VMs already deployed in the subscription. For example, if the subscription has a quota of 10 vCPUs for D-Series and has two standard_D1 instances deployed, then you can buy a reservation for 10 standard_D1 instances in this subscription. You can create quote increase request to resolve this issue.

- **Capacity restrictions** - In rare circumstances, Azure limits the purchase of new reservations for subset of VM sizes, because of low capacity in a region.

Buy a Reserved VM Instance

You can buy a reserved VM instance in the Azure portal. Pay for the reservation up front or with monthly payments. These requirements apply to buying a reserved VM instance:

- To buy a reservation, you must have owner role or reservation purchaser role on an Azure subscription.
- For EA subscriptions, the **Add Reserved Instances** option must be enabled in the EA portal. Or, if that setting is disabled, you must be an EA Admin for the subscription.
- For the Cloud Solution Provider (CSP) program, only the admin agents or sales agents can buy reservations.

To buy an instance:

1. Sign in to the Azure portal.
2. Select **All services** > **Reservations**.
3. Select **Add** to purchase a new reservation and then click **Virtual machine**.
4. Enter required fields. Running VM instances that match the attributes you select qualify to get the reservation discount. The actual number of your VM instances that get the discount depend on the scope and quantity selected.

If you have an EA agreement, you can use the **Add more option** to quickly add additional instances. The option isn't available for other subscription types.

Expand table

Field	Description
Subscription	The subscription used to pay for the reservation. The payment method on the subscription is charged the costs for the reservation. The subscription type must be an enterprise agreement (offer numbers: MS-AZR-0017P or MS-AZR-0148P) or Microsoft Customer Agreement or an individual subscription with pay-as-you-go rates (offer numbers: MS-AZR-0003P or MS-AZR-0023P). The charges are deducted from the Azure Prepayment (previously called monetary commitment) balance, if available, or charged as overage. For a subscription with pay-as-you-go rates, the charges are billed to the credit card or invoice payment method on the subscription.
Scope	The reservation's scope can cover one subscription or multiple subscriptions (shared scope). If you select:

Field	Description
	• **Single resource group scope** — Applies the reservation discount to the matching resources in the selected resource group only. • **Single subscription scope** — Applies the reservation discount to the matching resources in the selected subscription. • **Shared scope** — Applies the reservation discount to matching resources in eligible subscriptions that are in the billing context. For EA customers, the billing context is the enrollment. For individual subscriptions with pay-as-you-go rates, the billing scope is all eligible subscriptions created by the account administrator. • **Management group** - Applies the reservation discount to the matching resource in the list of subscriptions that are a part of both the management group and billing scope.
Region	The Azure region that's covered by the reservation.
VM Size	The size of the VM instances.
Optimize for	VM instance size flexibility is selected by default. Click **Advanced settings** to change the instance size flexibility value to apply the reservation discount to other VMs in the same VM size group. Capacity priority prioritizes data center capacity for your deployments. It offers additional confidence in your ability to launch the VM instances when you need them. Capacity priority is only available when the reservation scope is single subscription.
Term	One year or three years. There's also a 5-year term available only for HBv2 VMs.
Quantity	The number of instances being purchased within the reservation. The quantity is the number of running VM instances that can get the billing discount. For example, if you are running 10 Standard_D2 VMs in the East US, then you would specify quantity as 10 to maximize the benefit for all running VMs.

https://learn-video.azurefd.net/vod/player?id=4e688b15-2bc8-47b5-b67e-16ea39397709&locale=en-us&embedUrl=%2Fazure%2Fvirtual-machines%2Fprepay-reserved-vm-instances

Usage data and reservation utilization

Your usage data has an effective price of zero for the usage that gets a reservation discount. You can see which VM instance received the reservation discount for each reservation.

For more information about how reservation discounts appear in usage data, see Understand Azure reservation usage for your Enterprise enrollment if you are an EA customer. If you have an individual subscription, see Understand Azure reservation usage for your Pay-As-You-Go subscription.

Change a reservation after purchase

You can make the following types of changes to a reservation after purchase:

- Update reservation scope
- Instance size flexibility (if applicable)
- Ownership

You can also split a reservation into smaller chunks and merge already split reservations. None of the changes cause a new commercial transaction or change the end date of the reservation.

You can't make the following types of changes after purchase, directly:

- An existing reservation's region
- SKU
- Quantity
- Duration

However, you can *exchange* a reservation if you want to make changes.

Cancel, exchange, or refund reservations

You can cancel, exchange, or refund reservations with certain limitations. For more information, see Self-service exchanges and refunds for Azure Reservations.

Need help? Contact us.

If you have questions or need help, create a support request.

Next steps

- To learn how to manage a reservation, see Manage Azure Reservations.
- To learn more about Azure Reservations, see the following articles:
  - What are Azure Reservations?
  - Manage Reservations in Azure
  - Understand how the reservation discount is applied
  - Understand reservation usage for a subscription with pay-as-you-go rates
  - Understand reservation usage for your Enterprise enrollment
  - Windows software costs not included with reservations
  - Azure Reservations in Partner Center Cloud Solution Provider (CSP) program

Last updated on 04/10/2026

Save costs with Azure Dedicated Host reservations

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

When you commit to a reserved instance of Azure Dedicated Hosts, you can save money. The reservation discount is applied automatically to the number of running dedicated hosts that match the reservation scope and attributes. You don't need to assign a reservation to a dedicated host to get the discounts. A reserved instance purchase covers only the compute part of your usage and does include software licensing costs. See the Overview of Azure Dedicated Hosts for virtual machines.

Determine the right dedicated host SKU before you buy

Before you buy a reservation, you should determine which dedicated host you need. A SKU is defined for a dedicated host representing the VM series and type.

Start by going over the supported sizes for Windows virtual machine or Linux to identify the VM series.

Next, check whether it is supported on Azure Dedicated Hosts. Azure Dedicated Hosts pricing page has the complete list of dedicated hosts SKUs, their CPU information, and various pricing options (including reserved instances).

You may find several SKUs supporting your selected VM series (with different Types). Identify the best SKU by comparing the capacity of the host (number of vCPUs). Note that you will be able to apply your reservation to multiple dedicated hosts SKUs supporting the same VM series (for example DSv3_Type1 and DSv3_Type2) but not across different VM series (like DSv3 and ESv3). Over the lifetime of the reservation, you may need to switch between different dedicated host SKUs that support the same VM size, depending on the hardware usage fluctuations and lifecycle.

Purchase restriction considerations

Reserved instances are available for most dedicated host sizes, with some exceptions.

Reservation discounts don't apply for the following:

- **Clouds** - Reservations aren't available for purchase in Germany or China regions.
- **Insufficient quota** - A reservation that is scoped to a single subscription must have vCPU quota available in the subscription for the new reserved instance. For example, if the target subscription has a quota limit of 10 vCPUs for DSv3-Series, then you can't buy a reservation dedicated hosts supporting this series. The quota check for reservations includes the VMs and dedicated hosts already deployed in the subscription. You can create quota increase request to resolve this issue.
- **Capacity restrictions** - In rare circumstances, Azure limits the purchase of new reservations for subset of dedicated host SKUs, because of low capacity in a region.

Buy a reservation

You can buy a reserved instance of an Azure Dedicated Host instance in the Azure portal.

Pay for the reservation up front or with monthly payments. These requirements apply to buying a reserved Dedicated Host instance:

- To buy a reservation, you must have owner role or reservation purchaser role on an Azure subscription.
- For EA subscriptions, the **Add Reserved Instances** option must be enabled in the EA portal. Or, if that setting is disabled, you must be an EA Admin for the subscription.
- For the Cloud Solution Provider (CSP) program, only the admin agents or sales agents can buy reservations.

To buy an instance:

1. Sign in to the Azure portal.
2. Select **All services** > **Reservations**.
3. Select **Add** to purchase a new reservation and then click **Dedicated Hosts**.
4. Enter required fields. Running Dedicated Hosts instances that match the attributes you select qualify to get the reservation discount. The actual number of your Dedicated Host instances that get the discount depend on the scope and quantity selected.

If you have an EA agreement, you can use the **Add more option** to quickly add additional instances. The option isn't available for other subscription types.

Expand table

Field	Description
Subscription	The subscription used to pay for the reservation. The payment method on the subscription is charged the costs for the reservation. The subscription type must be an enterprise agreement (offer numbers: MS-AZR-0017P or MS-AZR-0148P) or Microsoft Customer Agreement or an individual subscription with pay-as-you-go rates (offer numbers: MS-AZR-0003P or MS-AZR-0023P). The charges are deducted from the Azure Prepayment (previously called monetary commitment) balance, if available, or charged as overage. For a subscription with pay-as-you-go rates, the charges are billed to the credit card or invoice payment method on the subscription.
Scope	The reservation's scope can cover one subscription or multiple subscriptions (shared scope). If you select:
Region	The Azure region that's covered by the reservation.
Dedicated Host Size	The size of the Dedicated Host instances.
Term	One year or three years.
Quantity	The number of instances being purchased within the reservation. The quantity is the number of running Dedicated Host instances that can get the billing discount.

- **Single resource group scope** — Applies the reservation discount to the matching resources in the selected resource group only.
- **Single subscription scope** — Applies the reservation discount to the matching resources in the selected subscription.
- **Shared scope** — Applies the reservation discount to matching resources in eligible subscriptions that are in the billing context. For EA customers, the billing context is the enrollment. For individual subscriptions with pay-as-you-go rates, the billing scope is all eligible subscriptions created by the account administrator.
- **Management group** — Applies the reservation discount to the matching resource in the list of subscriptions that are a part of both the management group and billing scope.

Usage data and reservation utilization

Your usage data has an effective price of zero for the usage that gets a reservation discount. You can see which VM instance received the reservation discount for each reservation.

For more information about how reservation discounts appear in usage data, see Understand Azure reservation usage for your Enterprise enrollment if you are an EA customer. If you have an individual subscription, see Understand Azure reservation usage for your Pay-As-You-Go subscription.

Change a reservation after purchase

You can make the following types of changes to a reservation after purchase:

- Update reservation scope
- Ownership

You can also split a reservation into smaller chunks and merge already split reservations. None of the changes cause a new commercial transaction or change the end date of the reservation.

You can't make the following types of changes after purchase, directly:

- An existing reservation's region
- SKU
- Quantity
- Duration

However, you can *exchange* a reservation if you want to make changes.

Cancel, exchange, or refund reservations

You can cancel, exchange, or refund reservations with certain limitations. For more information, see Self-service exchanges and refunds for Azure Reservations.

Need help? Contact us.

If you have questions or need help, create a support request.

Next steps

To learn how to manage a reservation, see Manage Azure Reservations.

To learn more about Azure Reservations, see the following articles:

- What are Azure Reservations?
- Using Azure Dedicated Hosts
- Dedicated Hosts Pricing
- Manage Reservations in Azure
- Understand how the reservation discount is applied
- Understand reservation usage for a subscription with pay-as-you-go rates
- Understand reservation usage for your Enterprise enrollment
- Windows software costs not included with reservations
- Azure Reservations in Partner Center Cloud Solution Provider (CSP) program

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Virtual machine size flexibility with Reserved VM Instances

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

When you buy a Reserved VM Instance, you can choose to optimize for instance size flexibility or capacity priority. For more information about setting or changing the optimize setting for reserved VM instances, see Change the optimize setting for reserved VM instances.

With a reserved virtual machine instance that's optimized for instance size flexibility, the reservation you buy can apply to the virtual machines (VMs) sizes in the same instance size flexibility group. In other words, when you buy a reserved VM instance of any size within an instance flexibility group, the instance applies to all sizes within the group. For example, if you buy a reservation for a VM size that's listed in the DSv2 Series, like Standard_DS3_v2, the reservation discount can apply to the other sizes that are listed in that same instance size flexibility group:

- Standard_DS1_v2
- Standard_DS2_v2
- Standard_DS3_v2
- Standard_DS4_v2

But that reservation discount doesn't apply to VMs sizes that are listed in different instance size flexibility groups, like SKUs in DSv2 Series High Memory: Standard_DS11_v2, Standard_DS12_v2, and so on.

Within the instance size flexibility group, the number of VMs the reservation discount applies to depends on the VM size you pick when you buy a reservation. It also depends on the sizes of the VMs that you have running. The ratio column compares the relative footprint for each VM size in that instance size flexibility group. Use the ratio value to calculate how the reservation discount applies to the VMs you have running.

Examples

The following examples use the sizes and ratios in the DSv2-series table.

You buy a reserved VM instance with the size Standard_DS4_v2 where the ratio or relative footprint compared to the other sizes in that series is 8.

- Scenario 1: Run eight Standard_DS1_v2 sized VMs with a ratio of 1. Your reservation discount applies to all eight of those VMs.
- Scenario 2: Run two Standard_DS2_v2 sized VMs with a ratio of 2 each. Also run a Standard_DS3_v2 sized VM with a ratio of 4. The total footprint is 2+2+4=8. So your reservation discount applies to all three of those VMs.
- Scenario 3: Run one Standard_DS5_v2 with a ratio of 16. Your reservation discount applies to half that VM's compute cost.
- Scenario 4: Run one Standard_DS5_v2 with a ratio of 16 and purchase an additional Standard_DS4_v2 reservation with a ratio of 8. Both reservations combine and apply the discount to entire VM.

The following sections show what sizes are in the same size series group when you buy a reserved VM instance optimized for instance size flexibility. SKUs that have same ratio and are in same size series group, would have no additional cost if you purchase reservation for any of those SKUs for same number of VMs running.

Instance size flexibility ratio

CSV below has the instance size flexibility groups, ArmSkuName and the ratios.

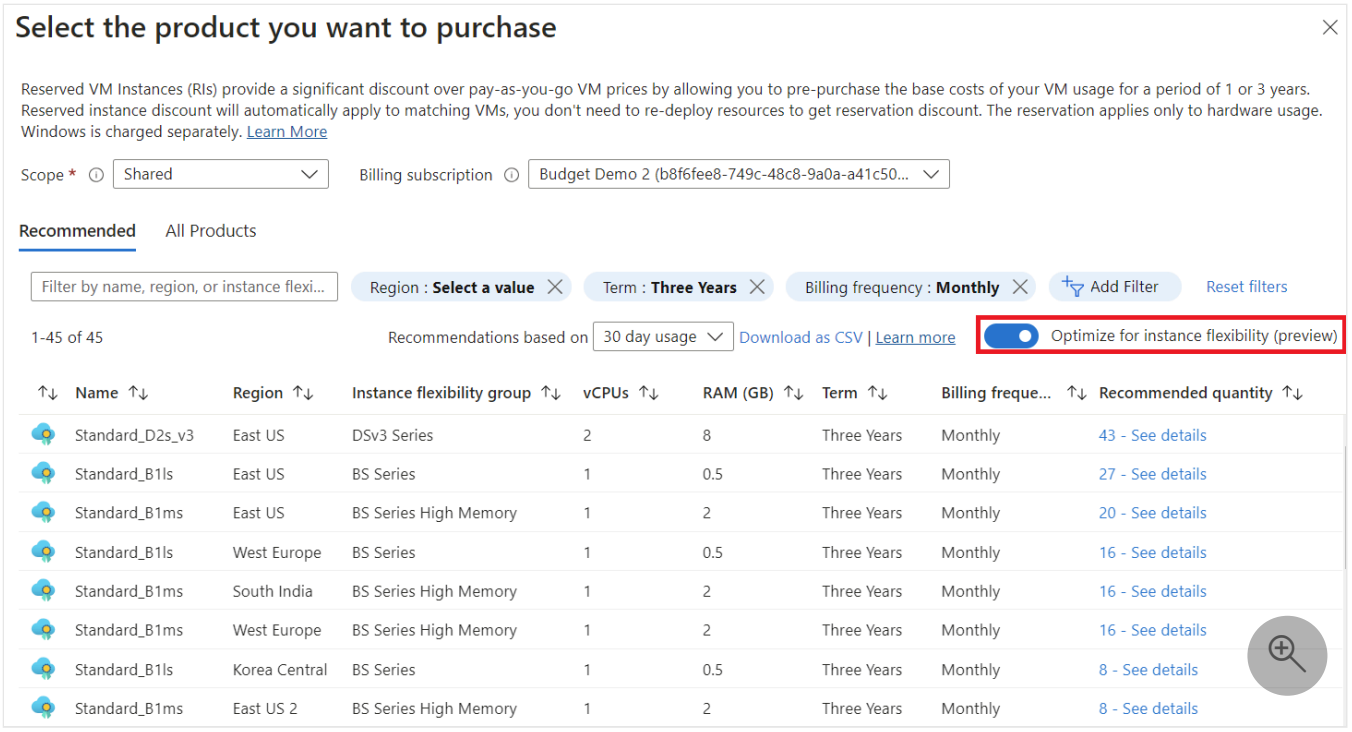
Instance size flexibility ratios

You can also use API or PowerShell to fetch the ratios. The CSV link and schema are kept up to date so you can use them programmatically. Read more here.

> **⊗ Caution**
>
> Beginning May 9, 2026, the CSV file will no longer receive updates and will be removed on August 30, 2026, in favor of accessing data through the API or PowerShell. Read more here.

View VM size recommendations

Azure shows VM size recommendations in the purchase experience. When enabled, the **Optimize for instance size flexibility (preview)** option groups and sorts recommendations by instance size flexibility.



Next steps

For more information, see What are Azure Reservations.

Last updated on 04/13/2026

On-demand capacity reservation

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale set ✔ Flexible scale sets

On-demand capacity reservation enables you to reserve compute capacity in an Azure region or an availability zone for any duration of time. Unlike reserved instances, you don't have to sign up for a one-year or three-year term commitment. You can create and delete reservations at any time and have full control over how you want to manage your reservations.

After you create the capacity reservation, you can use the resources immediately. Capacity is reserved for you until you delete the reservation.

Capacity reservation has some basic properties that are always defined at the time of creation:

- **VM size**: Each reservation is for one virtual machine (VM) size. An example is `Standard_D2s_v3`.
- **Location**: Each reservation is for one location (region). If that location has availability zones, the reservation can also specify one of the zones.
- **Quantity**: Each reservation has a quantity of instances to be reserved.

To create a capacity reservation, the parameters are passed to Azure as a capacity request. If Azure doesn't have capacity available that meets the request, the reservation deployment fails. Your deployment fails if you don't have an adequate subscription quota. Request a higher quota or try a different VM size, location, or zone combination.

Quota and capacity are separate checks. Quota is your subscription's permission to deploy resources, while capacity is the underlying infrastructure available in a specific region or zone.

Capacity reservations and Reserved Instances are also different. Capacity reservation secures capacity availability for deployment. Reserved Instances provide a billing discount only and don't guarantee capacity.

After Azure accepts your reservation request, it's available for VMs with matching configurations. To consume capacity reservation, the VM has to specify the reservation in its properties. Otherwise, the capacity reservation isn't used. One benefit of this design is that you can target only critical workloads to reservations and other noncritical workloads can run without reserved capacity.

Benefits of capacity reservation

- After deployment, capacity is reserved for your use and is always available within the scope of applicable service-level agreements (SLAs).
- Capacity can be deployed and deleted at any time with no term commitment.
- Capacity can be combined automatically with reserved instances to use term-commitment discounts.

SLA for capacity reservation

Read the SLA details in the SLA for capacity reservation .

Any claim against the SLA requires that you calculate the Minutes Not Available for the reserved capacity. Here's an example of how to calculate Minutes Not Available:

- An on-demand capacity reservation has a total capacity of five reserved units. The on-demand capacity reservation starts in the Unused Capacity state with zero VMs allocated.
- A supported deployment with a quantity of 5 is allocated to the on-demand capacity reservation. Three VMs succeed and two fail with a VM capacity error. The result is that two reserved units begin to accumulate Minutes Not Available.
- No action is taken for 20 minutes. The result is that two reserved units each accumulate 15 Minutes Not Available.
- At 20 minutes, a supported deployment with a quantity of 2 is attempted. One VM succeeds and the other VM fails with a VM capacity error. The result is one reserved unit stays at 15 accumulated Minutes Not Available. Another reserved unit resumes accumulating Minutes Not Available.
- Four more supported deployments with a quantity of 1 are made at 10-minute intervals. On the fourth attempt (60 minutes after the first capacity error), the VM is deployed. The result is that the last reserved unit adds 40 minutes of Minutes Not Available (four attempts x 10 minutes between attempts) for a total of 55 Minutes Not Available.

From this example accumulation of Minutes Not Available, here's the calculation of service credit:

- One reserved unit accumulated 15 minutes of downtime. The percentage uptime is 99.97%. This reserved unit doesn't qualify for a service credit.
- Another reserved unit accumulated 55 minutes of downtime. The percentage uptime is 99.87. This reserved unit qualifies for a service credit of 10%.

Limitations and restrictions

- Creating capacity reservations requires a quota in the same manner as when you create VMs.
- Creating capacity reservations is currently limited to certain VM series and sizes. The compute Resource SKUs list advertises the set of supported VM sizes.
- Reserved capacity for these VM series supports high availability using multi-zone or fault domains. The fault domain count for a Virtual Machine Scale Set is limited to 3, enforced at deployment.

  The following VM series support the creation of capacity reservations:

Expand table

Type	VM Series
General Purpose (Burstable)	B (Intel) Bsv2 (Intel), Basv2 (AMD), Bpsv2 (ARM)
General Purpose (General use A, D)	Av2 (Intel) D and Ds series, v2 and newer (Intel) Dd and Dds, v4 and newer (Intel, local disk) Dls and Dlds, v5 and newer (Intel, low memory) Da and Das, v4 series and newer (AMD) Dads, v5 and newer (AMD, local disk) Dals and Dalds, v6 and newer (AMD) Dps and Dpds series, v5 and newer (ARM) Dpls and Dplds series, v5 and newer (ARM, low memory)
General Purpose (General use, high memory, E)	E and Es, all versions (Intel) Ed and Eds series, v4 and newer (Intel, local disk) Ebdsv5 and Ebsv5 series (Intel, block storage) Ea and Eas, v4 and newer (AMD) Eads, v5 and newer series (AMD, local disk) Eps and Epds series, v5 and newer (ARM)
General Purpose (General use, compute optimized, F)	Fsv2 series (Intel) Fas and Fals, v6 and newer series (AMD) Famsv6 and newer (AMD) FX series (Intel) FXmsv2 (Intel) FXmdsv2 (Intel) Fadsv7 (AMD) Famdsv7 (AMD) Faldsv7 (AMD)

Type	VM Series
Storage optimized	Lsv3 (Intel) Lasv3 (AMD) Lasv4 (AMD) Laosv4 (AMD)
Confidential compute	DCsv2 series (Intel) DCasv5 and DCadsv5 series (AMD) DCesv5 and DCedsv5 series (Intel) ECasv5 and ECadsv5 series (AMD) ECesv5 and ECedsv5 series (Intel)

Reserved capacity for these VM series supports only multi-zone high availability. Virtual Machine Scale Sets with a fault domain availability construct is not supported.

Expand table

Type	VM Series
GPU-accelerated compute	NC-series, v3 (Intel) NCasT4_v3 series (AMD) NCADSA10_v4 series (AMD) NC_A100_v4 series (AMD) NV-series, v3 and newer (Intel) NVadsA10_v5 series (AMD) NVads V710 v5 series (AMD) NGads V620_v1 series (AMD)
Memory optimized	Msv2 Medium Memory series (Intel) Mdsv2 Medium Memory series (Intel) Msv2 High Memory (Intel) Mdsv3 Medium Memory series (Intel) Msv3 Medium Memory series (Intel) Mbsv3 and Mbdsv3 series (Intel)
Storage optimized	Lsv2 (Intel)-Preview

- Support for other VM series isn't currently available:

Expand table

Type	VM Series
GPU-accelerated compute	ND-series NCads H100 v5-series Hb-series Hc-series

Type	VM Series
Memory optimized	M series, v1 Msv3 HM Mdsv3 HM and Mdsv3 VHM

- The following deployment types are supported:
  - Single VM
  - Virtual Machine Scale Sets with Uniform Orchestration
  - Virtual Machine Scale Sets with Flexible Orchestration
- The following deployment types aren't supported:
  - Spot VMs
  - Azure Dedicated Host nodes or VMs deployed to dedicated hosts
  - Availability sets
- Other deployment constraints aren't supported. For example:
  - Proximity placement group
  - Update domains
  - Virtual Machine Scale Sets with single placement group set to `true`
  - Azure Ultra Disk Storage (formerly UltraSSD)
  - VMs resuming from hibernation
- A pinned subscription can't use the feature.
- Deployments using On demand capacity reservations are compatible with vNet Encryption when deployed using a supported VM size. For more information on which VM sizes this feature supports, see Virtual Network Encryption.
- Reservations are only available to paid Azure customers. Sponsored accounts such as Free Trial and Azure for Students aren't eligible to use this feature.
- Clouds supported for capacity reservation:
  - Azure Cloud
  - Azure for Government
  - Azure in China

Pricing and billing

Capacity reservations are priced at the same rate as the underlying VM size. For example, if you create a reservation for 10 D2s_v3 VMs, you start getting billed for 10 D2s_v3 VMs, even if the

reservation isn't being used.

If you then deploy a D2s_v3 VM and specify the reservation property, the capacity reservation gets used. After the VM is in use, you pay for only the VM and not the capacity reservation. Let's say you deploy six D2s_v3 VMs against the previously mentioned capacity reservation. You see a bill for six D2s_v3 VMs and four unused capacity reservations, both charged at the same rate as a D2s_v3 VM.

Both used and unused capacity reservations are eligible for Savings Plan and Reserved Instances term commitment discounts. In the previous example, if you have reserved instances for two D2s_v3 VMs in the same Azure region, the billing for two resources (either VM or unused capacity reservation) is zeroed out. The remaining eight D2s_v3 are billed normally. The term commitment discounts could be applied on either the VM or the unused capacity reservation.

Differences between on-demand capacity reservations and reserved instances

Expand table

Differences	On-demand capacity reservation	Reserved instances
Term	No term commitment required. Can be created and deleted as per the customer requirement.	Fixed-term commitment of either one year or three years.
Billing discount	Charged at pay-as-you-go rates for the underlying VM size.*	Significant cost savings over pay-as-you-go rates.
Capacity SLA	Provides capacity guarantee in the specified location (region or availability zone).	Doesn't provide a capacity guarantee. Customers can choose **Capacity priority** to gain better access, but that option doesn't carry an SLA.
Region vs. availability zones	Can be deployed per region or per availability zone.	Only available at the regional level.

*Eligible for the reserved instances discount if purchased separately.

Work with capacity reservation

Capacity reservation is created for a specific VM size in an Azure region or an availability zone. All reservations are created and managed as part of a capacity reservation group.

The group specifies the Azure location:

- The group sets the region in which all reservations are created. Examples are East US, North Europe, or Southeast Asia.
- The group sets the eligible zones. Examples are AZ1, AZ2, and AZ3 in any combination.
- If no zones are specified, then Azure will select the best zone available upon initial capacity reservation creation. All VMs using the capacity reservation group must be deployed without specifying a zone; each associated VM will be deployed to the zone Azure selected for the group.

Each reservation in a group is for one VM size. If eligible zones were selected for the group, the reservation must be for one of the supported zones.

A group can have only one reservation per VM size per zone, or just one reservation per VM size if no zones are selected.

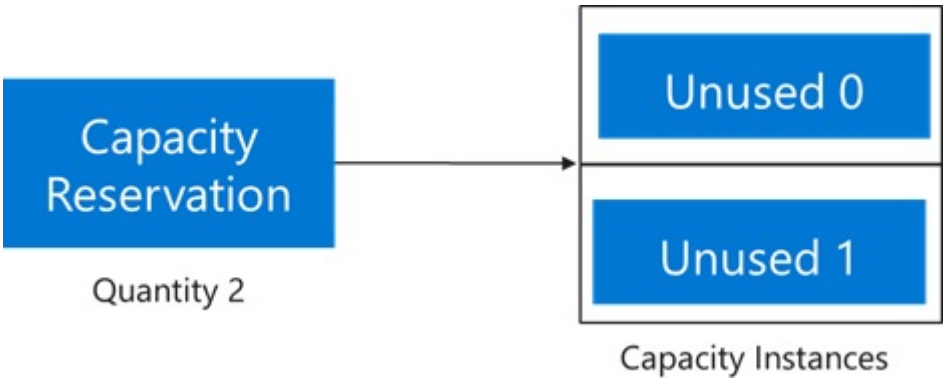
To consume capacity reservation, specify the capacity reservation group as one of the VM properties. If the group doesn't have a reservation matching the size and location, Azure returns an error message.

You can adjust the quantity reserved for reservation after initial deployment by changing the capacity property. Other changes to capacity reservation, such as VM size or location, aren't permitted. We recommend that you create a new reservation, migrate any existing VMs, and then delete the old reservation if it's no longer needed.

Capacity reservation doesn't create limits on the number of VM deployments. Azure supports allocating as many VMs as desired against the reservation. Because the reservation itself requires a quota, the quota checks are omitted for VM deployment up to the reserved quantity. Allocating VMs beyond the reserved quantity is called overallocating the reservation. Overallocating VMs isn't covered by the SLA, and the VMs are subject to quota checks and Azure fulfilling the extra capacity. After they're deployed, these extra VM instances can cause the quantity of VMs allocated against the reservation to exceed the reserved quantity. To learn more, see Overallocate capacity reservation.

Capacity reservation lifecycle

When a reservation is created, Azure sets aside the requested number of capacity instances in the specified location.

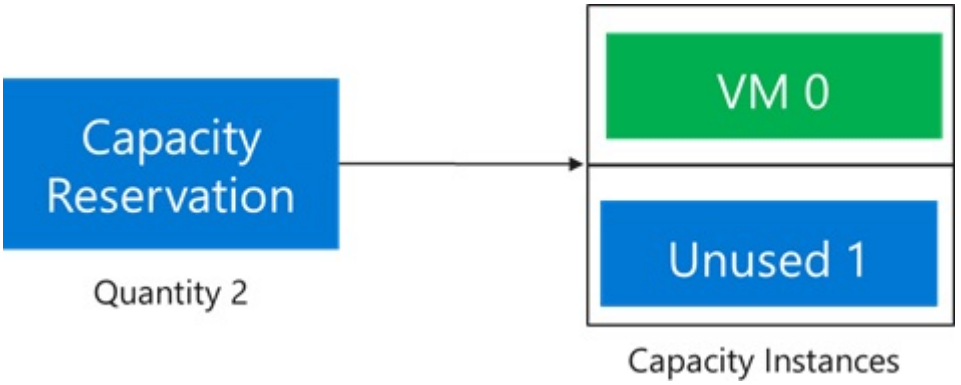


Track the state of the overall reservation through the following properties:

- `capacity`: Total quantity of instances reserved by the customer.
- `virtualMachinesAllocated`: List of VMs allocated against the capacity reservation and count toward consuming the capacity. These VMs are either **Running** or **Stopped** (**Allocated**), or they're in a transitional state such as **Starting** or **Stopping**. This list doesn't include the VMs that are in a deallocated state, which is referred to as **Stopped** (deallocated).
- `virtualMachinesAssociated`: List of VMs associated to the capacity reservation. This list has all the VMs that were configured to use the reservation, including the ones that are in a deallocated state.

The previous example starts with `capacity` as 2 and the length of `virtualMachinesAllocated` and `virtualMachinesAssociated` as 0.

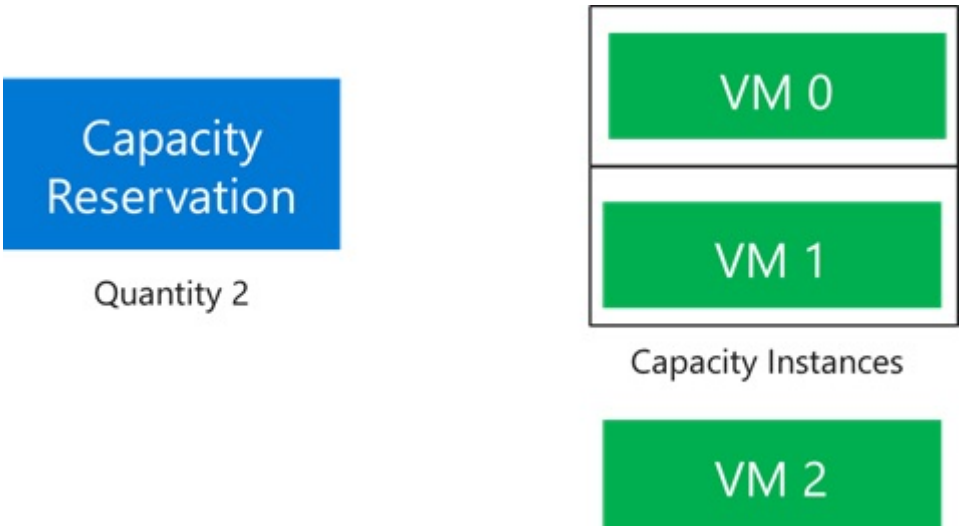
When a VM is then allocated against the capacity reservation, it consumes one of the reserved capacity instances.



The status of the capacity reservation shows `capacity` as 2 and the length of `virtualMachinesAllocated` and `virtualMachinesAssociated` as 1.

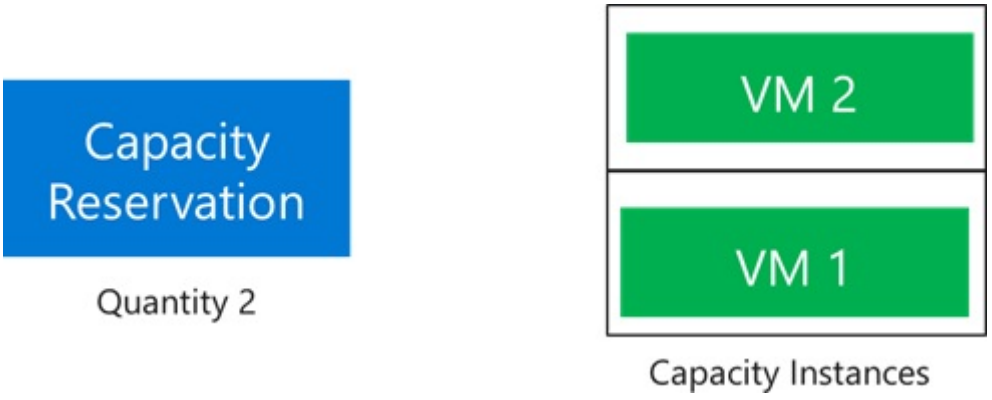
Allocations against the capacity reservation succeed if the VMs have matching properties and there's at least one empty capacity instance.

As shown in our example, when a third VM is allocated against the capacity reservation, the reservation enters the overallocated state. This third VM requires unused quota and extra capacity fulfillment from Azure. After the third VM is allocated, the capacity reservation now looks like this diagram:



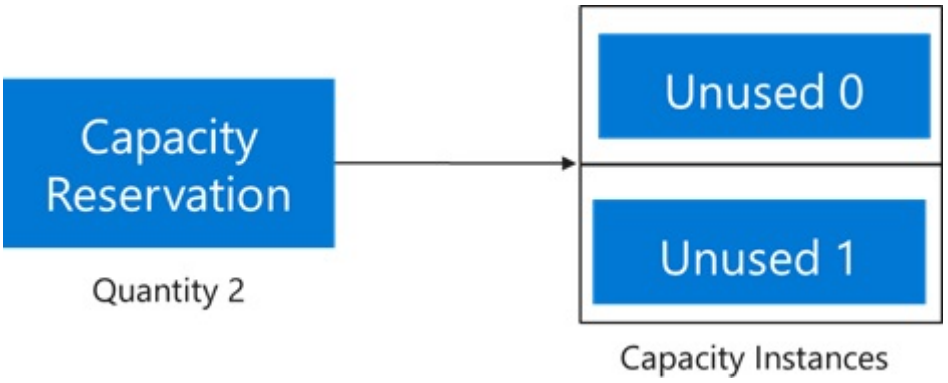
The `capacity` is 2 and the length of `virtualMachinesAllocated` and `virtualMachinesAssociated` is 3.

Now suppose the application scales down to the minimum of two VMs. Because the VM 0 needs an update, it's chosen for deallocation. The reservation automatically shifts to this state:

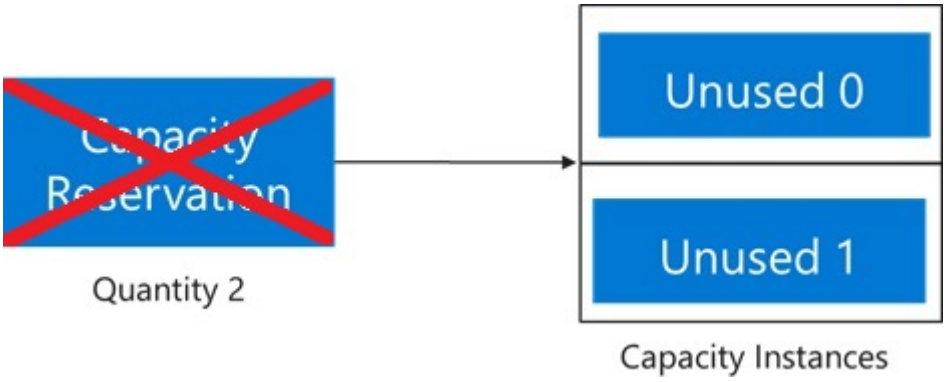


The `capacity` and the length of `virtualMachinesAllocated` are both 2. However, the length for `virtualMachinesAssociated` is still 3 because VM 0, although deallocated, is still associated to the capacity reservation. To prevent a quota overrun, the deallocated VM 0 still counts against the quota allocated to the reservation. If you have enough unused quota, you can deploy new VMs to the capacity reservation and receive the SLA from any unused reserved capacity. Or you can delete VM 0 to remove its use of the quota.

The capacity reservation exists until it's explicitly deleted. To delete a capacity reservation, the first step is to dissociate all the VMs in the `virtualMachinesAssociated` property. After disassociation is finished, the capacity reservation should look like this diagram:



The status of the capacity reservation shows `capacity` as 2 and the length of `virtualMachinesAssociated` and `virtualMachinesAllocated` as 0. From this state, you can delete the capacity reservation. After it's deleted, you don't pay for the reservation anymore.

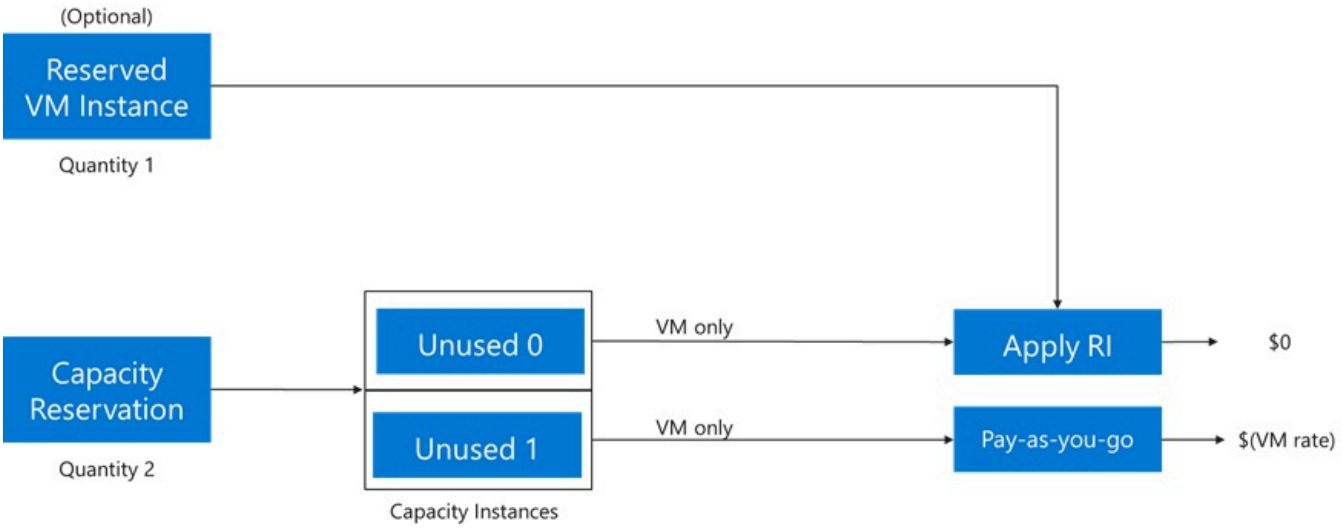


Usage and billing

When a capacity reservation is empty, VM usage is reported for the corresponding VM size and the location. VM reserved instances can cover some or all of the capacity reservation usage even when VMs aren't deployed.

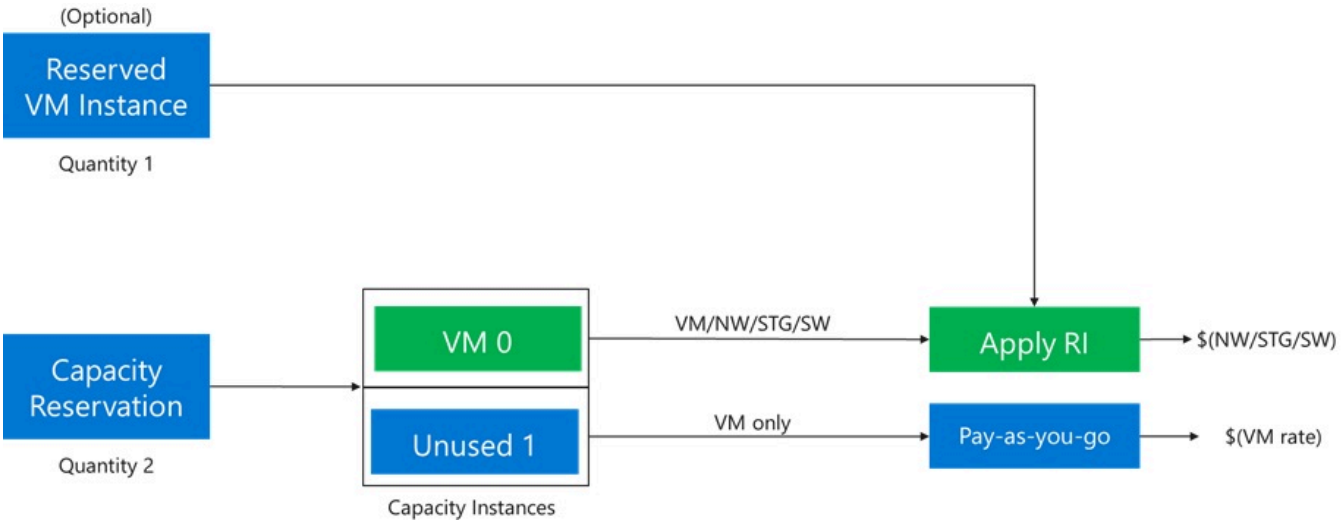
Example

For example, let's say that a capacity reservation with the reserved quantity of 2 was created. The subscription has access to one matching reserved VM instance of the same size. The result is two usage streams for the capacity reservation, one of which is covered by the reserved instance.



In the previous diagram, a reserved VM instance discount is applied to one of the unused instances and the cost for that instance is zeroed out. For the other instance, the pay-as-you-go rate is charged for the VM size reserved.

When a VM is allocated against the capacity reservation, the other VM components such as disks, network, extensions, and any other requested components must also be allocated. In this state, the VM usage reflects one allocated VM and one unused capacity instance. The reserved VM instance will zero out the cost of either the VM or the unused capacity instance. The other charges for disks, networking, and other components associated to the allocated VM also appear on the bill.



In the previous image, the VM reserved instance discount is applied to VM 0, which is only charged for other components, such as disks and networking. The other unused instance is being charged at a pay-as-you-go rate for the VM size reserved.

Frequently asked questions

- **What's the price of on-demand capacity reservation?**

The price of your on-demand capacity reservation is the same as the price of the underlying VM size associated to the reservation. When you use capacity reservation, you're charged for the VM size you selected at pay-as-you-go rates, whether the VM was provisioned or not. For more information, see the Windows and Linux VM pricing pages.

- **Will I get charged twice for the cost of on-demand capacity reservation and for the actual VM when I finally provision it?**

  No, you only get charged once for on-demand capacity reservation.

- **Can I apply Azure Reserved Virtual Machine Instances to on-demand capacity reservations to lower my costs?**

  Yes, you can apply existing or future reserved instances to on-demand capacity reservations and receive reserved instance discounts. Available reserved instances are applied automatically to capacity reservations the same way they're applied to VMs.

- **What's the difference between Reserved Virtual Machine Instances and on-demand capacity reservations?**

  Both reserved instances and on-demand capacity reservations are applicable to Azure VMs. However, reserved instances provide discounted reservation rates for your VMs compared to pay-as-you-go rates as a result of a one-year or three-year term commitment. Conversely, on-demand capacity reservations don't require a commitment.

  You can create or cancel a capacity reservation at any time. However, no discounts are applied, and you incur charges at pay-as-you-go rates after your capacity reservation is successfully provisioned. Unlike reserved instances, which prioritize capacity but don't guarantee it, when you purchase an on-demand capacity reservation, Azure sets aside compute capacity for your VM and provides an SLA guarantee.

- **Which scenarios would benefit the most from on-demand capacity reservations?**

  Typical scenarios include business continuity, disaster recovery, and scale-out of mission-critical applications.

Related content

Get started reserving compute capacity. Check out other capacity reservation articles:

- Create a capacity reservation

- Overallocating capacity reservation
- Modify a capacity reservation
- Associate a VM
- Remove a VM
- Associate a virtual machine scale set - Flexible Orchestration
- Associate a virtual machine scale set - Uniform Orchestration
- Remove a virtual machine scale set

Last updated on 02/17/2026

Create a capacity reservation

Article • 09/09/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale set ✔ Flexible scale sets

Capacity reservation is always created as part of a capacity reservation group. The first step is to create a group if a suitable one doesn't exist already, and then create reservations. After reservations are successfully created, they're immediately available for use with virtual machines (VMs). The capacity is reserved for your use as long as the reservation isn't deleted.

A well-formed request for a capacity reservation group should always succeed because it doesn't reserve any capacity. It just acts as a container for reservations. However, a request for capacity reservation could fail if you don't have the required quota for the VM series or if Azure doesn't have enough capacity to fulfill the request. Either request more quota or try a different VM size, location, or zone combination.

Capacity reservation creation succeeds or fails in its entirety. For a request to reserve 10 instances, success is returned only if all 10 instances could be allocated. Otherwise, the capacity reservation creation fails.

Considerations

The capacity reservation must meet the following rules:

- The location parameter must match the location property for the parent capacity reservation group. A mismatch results in an error.
- The VM size must be available in the target region. Otherwise, the reservation creation fails.
- The subscription must have available quota equal to or more than the quantity of VMs being reserved for the VM series and for the region overall. If needed, request more quota.
    - As needed to satisfy existing quota limits, you can do single VMs in stages. Create a capacity reservation with a smaller quantity and reallocate that quantity of VMs. This approach frees up quota to increase the quantity reserved and add more VMs. Alternatively, if the subscription uses different VM sizes in the same series, reserve and redeploy VMs for the first size. Then add a reservation to the group for another size and redeploy the VMs for the new size to the reservation group. Repeat the process until it's complete.
    - For scale sets, available quota is required unless the scale set or you delete its VM instances, capacity is reserved, and the scale set instances are added by

using reserved capacity. If the scale set is updated by using blue green deployment, then reserve the capacity and deploy the new scale set to the reserved capacity at the next update.

- Each capacity reservation group can have exactly one reservation for a specific VM size. For example, you can create only one capacity reservation for the VM size `Standard_D2s_v3`. Attempting to create a second reservation for `Standard_D2s_v3` in the same capacity reservation group results in an error. However, you can create another reservation in the same group for other VM sizes, such as `Standard_D4s_v3` and `Standard_D8s_v3`.
- For a capacity reservation group that supports zones, each reservation type is defined by the combination of **VM size** and **zone**. For example, one capacity reservation for `Standard_D2s_v3` in `Zone 1`, another capacity reservation for `Standard_D2s_v3` in `Zone 2`, and a third capacity reservation for `Standard_D2s_v3` in `Zone 3` are supported.

Check VM sizes that are available for capacity reservation in a region

Before you create a capacity reservation, you can check the VM sizes that are available for the reservation for a particular region.

Portal

1. Open the Azure portal.
2. In the search bar, enter **capacity reservation groups**.
3. Select **Capacity reservation groups** from the options.
4. Select **Create**.
5. On the **Basics** tab, create a capacity reservation group:
   a. Select a **Subscription**.
   b. Select or create a resource group.
   c. Name your group.
   d. Select a region.
   e. Optionally, select **Availability zones** or allow Azure to choose for you.
6. Select **Next**.
7. On **VM size**, select **See all sizes** to check what VM sizes are available for capacity reservation.

Create a capacity reservation

API

1. Create a capacity reservation group.

   To create a capacity reservation group, construct the following `PUT` request on the `Microsoft.Compute` provider:

   ```rest
   PUT
   https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{CapacityReservationGroupName}&api-version=2021-04-01
   ```

   In the request body, include the following parameter:

   ```JSON
   {
     "location":"eastus"
   }
   ```

   This group is created to contain reservations for the US East location.

   The group in the following example only supports regional reservations because zones weren't specified at the time of creation. To create a zonal group, pass an extra parameter `zone` in the request body:

   ```JSON
   {
     "location":"eastus",
     "zones": ["1", "2", "3"]
   }
   ```

2. Create a capacity reservation.

   To create a reservation, construct the following `PUT` request on the `Microsoft.Compute` provider:

   ```rest
   ```

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{CapacityReservationGroupName}/capacityReservations/{capacityReservationName}?api-version=2021-04-01
```

In the request body, include the following parameters:

JSON

```
{
  "location": "eastus",
  "sku": {
    "name": "Standard_D2s_v3",
    "capacity": 5
  },
 "tags": {
        "environment": "testing"
}
```

The preceding request creates a reservation in the East US location for five quantities of the D2s_v3 VM size.

Check on your capacity reservation

After the capacity reservation is successfully created, it's immediately available for use with VMs.

API

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{CapacityReservationGroupName}/capacityReservations/{capacityReservationName}?api-version=2021-04-01
```

JSON

```
{
    "name": "<CapacityReservationName>",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/capacityReservationGroups/{CapacityRes
```

```
ervationGroupName}/capacityReservations/{CapacityReservationName}",
    "type": "Microsoft.Compute/capacityReservationGroups/capacityReserva
tions",
    "location": "eastus",
    "tags": {
        "environment": "testing"
    },
    "sku": {
        "name": "Standard_D2s_v3",
        "capacity": 5
    },
    "properties": {
        "reservationId": "<reservationId>",
         "provisioningTime": "<provisioningTime>",
         "provisioningState": "Updating"
    }
}
```

Next step

Learn how to modify your capacity reservation

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Overallocate capacity reservation

Article • 09/09/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale set ✔ Flexible scale sets

Azure permits the association of extra virtual machines (VMs) above the number of capacity reservations. These VMs are available to allow for burst and other scale-out scenarios without the limits of reserved capacity. The only difference is that the count of VMs beyond the quantity reserved doesn't receive the capacity availability service-level agreement (SLA) benefit. As long as Azure has available capacity that meets the VM requirements, the extra allocation succeeds.

The Instance View of a capacity reservation group provides a snapshot of usage for each member capacity reservation. You can use the Instance View to see how overallocation works.

This article assumes you created a capacity reservation group (`myCapacityReservationGroup`), a member capacity reservation (`myCapacityReservation`), and a VM (*myVM1*) that's associated to the group. For more information, see Create a capacity reservation and Associate a VM to a capacity reservation.

Instance View for a capacity reservation group

The Instance View for a capacity reservation group looks like this example:

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/myC
apacityReservationGroup?$expand=instanceview&api-version=2021-04-01
```

JSON

```
{
    "name": "myCapacityReservationGroup",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName
}/providers/Microsoft.Compute/capacityReservationGroups/myCapacityReservatio
nGroup",
    "type": "Microsoft.Compute/capacityReservationGroups",
    "location": "eastus",
    "properties": {
        "capacityReservations": [
            {
```

```
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{resou
rceGroupName}/providers/Microsoft.Compute/capacityReservationGroups/MYCAPACI
TYRESERVATIONGROUP/capacityReservations/MYCAPACITYRESERVATION"
            }
        ],
        "virtualMachinesAssociated": [
            {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{resou
rceGroupName}/providers/Microsoft.Compute/virtualMachines/myVM1"
            }
        ],
        "instanceView": {
            "capacityReservations": [
                {
                    "name": "myCapacityReservation",
"utilizationInfo": {
                        "virtualMachinesAllocated": [
                            {
                                "id": "/subscriptions/{subscriptionId}/resou
rceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/my
VM1"
                            }
                        ]
                    },
                    "statuses": [
                        {
                            "code": "ProvisioningState/succeeded",
                            "level": "Info",
                            "displayStatus": "Provisioning succeeded",
                            "time": "<time>"
                        }
                    ]
                }
            ]
        }
    }
}
```

Let's say we create another VM named *myVM2* and associate it to the preceding capacity reservation group.

The Instance View for the capacity reservation group now looks like this example:

JSON

```
{
    "name": "myCapacityReservationGroup",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName
}/providers/Microsoft.Compute/capacityReservationGroups/myCapacityReservatio
nGroup",
    "type": "Microsoft.Compute/capacityReservationGroups",
    "location": "eastus",
    "properties": {
```

```
        "capacityReservations": [
            {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{resou
rceGroupName}/providers/Microsoft.Compute/capacityReservationGroups/MYCAPACI
TYRESERVATIONGROUP/capacityReservations/MYCAPACITYRESERVATION"
            }
        ],
        "virtualMachinesAssociated": [
            {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{resou
rceGroupName}/providers/Microsoft.Compute/virtualMachines/myVM1"
            },
 {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{resou
rceGroupName}/providers/Microsoft.Compute/virtualMachines/myVM2"
            }
        ],
        "instanceView": {
            "capacityReservations": [
                {
                    "name": "myCapacityReservation",
"utilizationInfo": {
                        "virtualMachinesAllocated": [
                            {
                                "id": "/subscriptions/{subscriptionId}/resou
rceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/my
VM1"
                            },
{
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{resou
rceGroupName}/providers/Microsoft.Compute/virtualMachines/myVM2"
            }
                        ]
                    },
                    "statuses": [
                        {
                            "code": "ProvisioningState/succeeded",
                            "level": "Info",
                            "displayStatus": "Provisioning succeeded",
                            "time": "<time>"
                        }
                    ]
                }
            ]
        }
    }
}
```


The length of `virtualMachinesAllocated` (2) is greater than `capacity` (1). This valid state is referred to as *overallocated*.

> **ⓘ Important**

Azure won't stop allocations because a capacity reservation is fully consumed. Autoscale rules, temporary scale-out, and related requirements work beyond the quantity of reserved capacity if Azure has available capacity and other constraints like available quota are met.

States and considerations

There are three valid states for a specific capacity reservation:

Expand table

State	Status	Considerations
Reserved capacity available	Length of `virtualMachinesAllocated` < `capacity`	Is all the reserved capacity needed? Optionally, reduce the capacity to reduce costs.
Reservation consumed	Length of `virtualMachinesAllocated` == `capacity`	More VMs won't receive the capacity SLA unless some existing VMs are deallocated. Optionally, try to increase the capacity so that extra planned VMs receive an SLA.
Reservation overallocated	Length of `virtualMachinesAllocated` > `capacity`	More VMs won't receive the capacity SLA. Also, the quantity of VMs (length of `virtualMachinesAllocated` – `capacity`) won't receive a capacity SLA if deallocated. Optionally, increase the capacity to add capacity SLA to more of the existing VMs.

Next step

Learn how to remove VMs from a capacity reservation

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Modify a capacity reservation

Article • 09/09/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale set ✔ Flexible scale sets

After you create a capacity reservation group and a capacity reservation, you might want to modify your reservations. This article explains how to do the following actions by using the API, the Azure portal, and PowerShell.

- ✔ Update the number of instances reserved in a capacity reservation.
- ✔ Resize VMs associated to a capacity reservation group.
- ✔ Delete the capacity reservation group and capacity reservation.

Update the number of instances reserved

Update the number of virtual machine (VM) instances reserved in a capacity reservation.

> **ⓘ Important**
>
> In rare cases when Azure can't fulfill the request to increase the quantity reserved for existing capacity reservations, a reservation might go into a *Failed* state and become unavailable until the quantity is restored to the original amount.

API

rest

```
    PATCH
https://management.azure.com/subscriptions/{subscriptionId}/resourceGrou
ps/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGr
oups/{CapacityReservationGroupName}/capacityReservations/{capacityReserv
ationName}?api-version=2021-04-01
```

In the request body, update the `capacity` property to the new count that you want to reserve:

JSON

```
{
    "sku":
    {
        "capacity": 5
```

```
    }
}
```

In this example, the `capacity` property is now set to `5`.

Resize VMs associated to a capacity reservation group

You must do one of the following options if the VM being resized is currently attached to a capacity reservation group and that group doesn't have a reservation for the target size:

- Create a new reservation for that size.
- Remove the VM from the reservation group before you resize.

Check if the target size is part of the reservation group.

API

1. Get the names of all capacity reservations within the group.

rest

```
    GET
https://management.azure.com/subscriptions/{subscriptionId}/resourc
eGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityRes
ervationGroups/{CapacityReservationGroupName}?api-version=2021-04-
01
```

JSON

```
{
    "name": "<CapacityReservationGroupName>",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resource
GroupName}/providers/Microsoft.Compute/capacityReservationGroups/{C
apacityReservationGroupName}",
    "type": "Microsoft.Compute/capacityReservationGroups",
    "location": "eastUS",
    "zones": [
        "1"
    ],
    "properties": {
        "capacityReservations": [
            {
                "id": "/subscriptions/{subscriptionId}/resourceGrou
```

```
ps/{resourceGroupName}/providers/Microsoft.Compute/capacityReservat
ionGroups/{CapacityReservationGroupName}/capacityReservations/{capa
cityReservationName1}"
            },
{
                "id": "/subscriptions/{subscriptionId}/resourceGrou
ps/{resourceGroupName}/providers/Microsoft.Compute/capacityReservat
ionGroups/{CapacityReservationGroupName}/capacityReservations/{capa
cityReservationName2}"
            }
        ]
    }
}
```

2. Find out the VM size reserved for each reservation. The following example is for `capacityReservationName1`, but you can repeat this step for other reservations.

rest

```
    GET
https://management.azure.com/subscriptions/{subscriptionId}/resourc
eGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityRes
ervationGroups/{CapacityReservationGroupName}/capacityReservations/
{capacityReservationName1}?api-version=2021-04-01
```

JSON

```
{
    "name": "capacityReservationName1",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resource
GroupName}/providers/Microsoft.Compute/capacityReservationGroups/{C
apacityReservationGroupName}/capacityReservations/{capacityReservat
ionName1}",
    "type": "Microsoft.Compute/capacityReservationGroups/capacityRe
servations",
    "location": "eastUS",
    "sku": {
        "name": "Standard_D2s_v3",
        "capacity": 3
    },
    "zones": [
        "1"
    ],
    "properties": {
        "reservationId": "<reservationId>",
        "provisioningTime": "<provisioningTime>",
        "provisioningState": "Succeeded"
    }
}
```

3. Consider the following scenarios:

   - If the target VM size isn't part of the group, create a new capacity reservation for the target VM.
   - If the target VM size already exists in the group, resize the VM.

Delete a capacity reservation group and capacity reservation

Azure allows a group to be deleted when all the member capacity reservations are deleted and no VMs are associated to the group.

To delete a capacity reservation, first find out all of the VMs that are associated to it. The list of VMs is available under the `virtualMachinesAssociated` property.

API

First, find all the VMs associated to the capacity reservation group and dissociate them:

rest

```
    GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGrou
ps/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGr
oups/{CapacityReservationGroupName}?$expand=instanceView&api-
version=2021-04-01
```

JSON

```
{
    "name": "<capacityReservationGroupName>",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroup
Name}/providers/Microsoft.Compute/capacityReservationGroups/{capacityRes
ervationGroupName}",
    "type": "Microsoft.Compute/capacityReservationGroups",
    "location": "eastus",
    "properties": {
        "capacityReservations": [
            {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{r
esourceGroupName}/providers/Microsoft.Compute/capacityReservationGroups/
{capacityReservationGroupName}/capacityReservations/{capacityReservation
Name}"
            }
```

```
        ],
        "virtualMachinesAssociated": [
            {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{r
esourceGroupName}/providers/Microsoft.Compute/virtualMachines/{VirtualMa
chineName1}"
            },
            {
                "id": "/subscriptions/{subscriptionId}/resourceGroups/{r
esourceGroupName}/providers/Microsoft.Compute/virtualMachines/{VirtualMa
chineName2}"
            }
        ],
        "instanceView": {
            "capacityReservations": [
                {
                    "name": "{capacityReservationName}",
                    "utilizationInfo": {
                        "virtualMachinesAllocated": [
                            {
                                "id": "/subscriptions/{subscriptionId}/r
esourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMac
hines/{VirtualMachineName1}"
                            }
                        ]
                    },
                    "statuses": [
                        {
                            "code": "ProvisioningState/succeeded",
                            "level": "Info",
                            "displayStatus": "Provisioning succeeded",
                            "time": "<time>"
                        }
                    ]
                }
            ]
        }
    }
}
```


From the preceding response, find the names of all the VMs under the `virtualMachinesAssociated` property. Remove them from the capacity reservation group by using the steps in Remove a VM association to a capacity reservation.

After all the VMs are removed from the capacity reservation group, delete the member capacity reservations:

rest

```
DELETE
https://management.azure.com/subscriptions/{subscriptionId}/resourceGrou
ps/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGr
```

```
oups/{CapacityReservationGroupName}/capacityReservations/{capacityReserv
ationName}?api-version=2021-04-01
```

Lastly, delete the parent capacity reservation group:

rest

```
DELETE
https://management.azure.com/subscriptions/{subscriptionId}/resourceGrou
ps/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGr
oups/{CapacityReservationGroupName}?api-version=2021-04-01
```

Restore instance quantity

A well-formed request for reducing the reserved quantity should always succeed no matter the number of VMs associated to the reservation. However, increasing the reserved quantity might require more quota and for Azure to fulfill the request for more capacity. In a rare scenario in which Azure can't fulfill the request to increase the reserved quantity for existing reservations, the reservation might go into a *Failed* state and become unavailable until the reserved quantity is restored to the original amount.

> **Note**
>
> If a reservation is in a *Failed* state, all the VMs that are associated to the reservation continue to work as normal.

For example, let's say `myCapacityReservation` has 5 instances reserved. You request 5 extra instances, which makes the total reserved quantity equal to 10. However, because of a constrained capacity situation in the region, Azure can't fulfill the extra 5 instances that you requested. In this case, `myCapacityReservation` fails to meet its intended state of 10 reserved instances and goes into a *Failed* state.

To resolve this failure, follow these steps to locate the old quantity reserved value:

1. In the Azure portal, go to Application Change Analysis.
2. Select the applicable **Subscription**, **Resource group**, and **Time range** settings in the filters. You can only go back up to 14 days in the past in the **Time range** filter.
3. Search for the name of the capacity reservation.
4. Look for the change in the `sku.capacity` property for that reservation. The old quantity reserved is the value under the **Old Value** column.

Update `myCapacityReservation` to the old quantity reserved. After it's updated, the reservation is available immediately for use with your VMs.

Next step

Learn how to remove VMs from a capacity reservation

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Share a Capacity Reservation Group (Preview)

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Uniform scale set ✔ Flexible scale sets

> ⓘ **Important**
>
> This feature is currently in **Preview**. See the Preview Terms of Use for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

On-demand Capacity Reservation Group (CRG) can be shared with other subscriptions. Using this option can make it easier to manage some common configuration needs:

- **Reuse of capacity reserved for disaster recovery.** Reserved capacity is the primary means to obtain capacity assurance in another region or zone in the event of the primary region or zone becoming unavailable. Reservation sharing supports reuse of disaster recovery capacity by subscriptions, hosting less critical workloads such as development and testing, or subscriptions used to run job-oriented workloads. Capacity reuse can save on total capacity costs and drive more value from the reserved capacity.
- **Central management of capacity.** Quota requests and term commitments are often administered by a central operations team as part of cost management. Now, reserved capacity needs can be assessed and managed more centrally to align on capacity and cost management.
- **Separate security and capacity concerns.** Applications implemented with multiple subscriptions for security reasons can operate from a common pool of capacity. This pattern is common with service providers serving their own end customers.
- **More cost-effective scale-out with capacity assurance.** Applications that scale at different rates and times can share one pool of reserved capacity.

How to share a Capacity Reservation Group

Sharing reserved capacity requires at least two subscriptions:

- **Provider subscription** - the subscription that creates and hosts the Capacity Reservation Group and member Capacity Reservations.
- **Consumer subscription** - another subscription that is granted access to the reserved capacity, obtaining the ability to deploy virtual machines (VMs) with the Capacity

Reservation Service Level Agreement (SLA).

A given Capacity Reservation Group can be shared with up to 100 consumer subscriptions. All the member Capacity Reservations in the Group will be accessible from consumer subscriptions.

Configuring a Capacity Reservation Group sharing relationship has three steps:

1. In the Consumer subscription, configure an On Demand capacity Reservation (ODCR) owner from the Producer subscription with the rights `Microsoft.Compute/capacityReservationGroups/share/action`.
2. In the Producer subscription, add the Consumer subscription ID to the Capacity Reservation Group *shared* list. See Share a capacity Reservation Group to learn how to add a Consumer subscription to the sharing profile.
3. In the Producer subscription, configure at least one VM owner in the Consumer subscription with the following rights:

    - `Microsoft.Compute/capacityReservationGroups/read`
    - `Microsoft.Compute/capacityReservationGroups/deploy`
    - `Microsoft.Compute/capacityReservationGroups/capacityReservations/read`
    - `Microsoft.Compute/capacityReservationGroups/capacityReservations/deploy`

If the ODCR owner and the VM owner already have either Owner or Contributor Azure role in both the Provider and Consumer subscriptions, then no further action is needed for granting share, read, and deploy permissions. To learn more on how to assign an Azure role, see Role Assignment Steps or Azure custom role

Once complete, a VM owner in the Consumer subscription can enumerate the shared CRG. View the list of Capacity Reservation Groups for a subscription and deploy VMs by setting the `capacityReservationGroup` property on Virtual Machines or Virtual Machine Scale Sets. Keep in mind, the ability to modify the Capacity Reservation Group (CRG) remains only with the ODCR administrator in the producer subscription.

> **Note**
>
> There are no extra charges for using the shared Capacity Reservation Group feature. Unused reservations are charged to the subscription that owns the reservation. VM usage is charged to the subscription that uses the capacity reservation as it does today. For details on how Reserved Instance (RI) applies to the feature, see Use of Reserved Instances with shared Capacity Reservation Groups section.

Usage patterns

The Provider subscription sharing a Capacity Reservation Group can allow:

- Consumer subscriptions to access a specific Capacity Reservation Group in the Provider subscription
- Consumer subscription to access all Capacity Reservation Groups created in the Provider subscription
- All Consumer subscriptions in a specific Management Group to access the Capacity Reservation Group

> **Note**
>
> Azure strongly recommends using one main Provider subscription to host Capacity Reservation Groups for each application, workload, or usage scope to share across other subscriptions. Creating Capacity Reservation Groups in many different subscriptions and then cross-sharing in a matrix fashion will create management challenges and lead to confusion at VM deployment.

Prerequisites for sharing a Capacity Reservation Group

- An ODCR owner in the Provider subscription must have sufficient rights to be able to share a CRG
- A VM owner in the Consumer subscription must have sufficient rights to be able to make deployments in capacity reservation (CR) in shared CRG
- A VM being deployed in the shared CRG must match the VM SKU, region, and zone if applicable

Limitations of sharing a Capacity Reservation Group

Limitations by design:

- Sharing works with an explicit list of target Consumer subscriptions. Azure doesn't support wildcard or tenant level sharing.
- A CRG can be shared with a maximum of 100 Consumer subscriptions.
- Sharing is done per Capacity Reservation Group, which grants access to all member Capacity Reservations. Individual Capacity Reservations can't be shared. To isolate specific

Capacity Reservations, create multiple Capacity Reservation Groups and share only those Capacity Reservations that contain shared capacity.
- By default, Capacity Reservation Group administrators in the subscription owning a Capacity Reservation Group can't modify VM instances deployed by other subscriptions. If such VM access is desired, more rights to VMs on the shared subscriptions must be granted separately.

Limitations for Preview:

- Azure portal support isn't available; API and other Azure clients are available.
- Reprovisioning of Virtual Machine Scale Set VMs using a shared Capacity Reservation Group isn't supported during a zone outage.
- There is a known issue of Capacity Reservation Groups - List by Subscription ID not giving the right response if there is no CRG created by the subscription making the `GET` call to list shared CRGs in the region. To get the correct response, ensure you have a local CRG created in the subscription making the API call in the same region where you would like to enumerate the shared CRGs. Alternatively, use the Azure Resource Graph query provided to get the list of CRGs shared with your subscription.

Share a Capacity Reservation Group

Capacity Reservation Groups are shared by adding Consumer subscriptions in the sharing profile of newly created or existing Capacity Reservation Groups. Once shared, the subscriptions which are part of the sharing profile can deploy VMs or Virtual Machine Scale Sets in the shared Capacity Reservation Group.

Add sharing profile to a new Capacity Reservation Group

Share a Capacity Reservation Group during creation by adding subscriptions in the sharing profile.

API

To share a Capacity Reservation group on creation, construct the following PUT request:

rest

```
PUT https://management.azure.com/subscriptions/{provider-subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/capacityReservationGroups/myCapacityReservationGroup?api-version=2024-03-01
```

In the request body, include the subscription ID to share the Capacity Reservation Group in the `sharing profile`:

JSON

```
{
    "location": "westus",
    "tags": {
        "department": "finance"
    },
    "zones": [
        "1"
    ],
    "properties": {
        "sharingProfile": {
            "subscriptionIds": [{
                    "id": "/subscriptions/{consumer-subscription-id1}"
                }, {
                    "id": "/subscriptions/{consumer-subscription-id2}"
                }
            ]
        }
    }
}
```

This example is to create a shared CRG in zone 1 of Region West US.

To learn more, see Capacity Reservation Groups - Create Or Update.

Add sharing profile to an existing Capacity Reservation Group

Add a sharing profile and share with subscriptions for an existing Capacity Reservation Group.

API

To add a sharing profile to an existing CRG, construct the following PUT request.

The following example adds sharing profile to an existing Capacity Reservation Group called `myCapacityReservationGroup` and is shared with three subscription IDs.

rest

```
PUT https://management.azure.com/subscriptions/{provider-subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/capacityReservationGroups/myCapacityReservationGroup?api-version=2024-03-01
```

In the request body, include the consumer subscription IDs to share the Capacity Reservation Group in the `sharing profile`.

JSON

```
{
    "location": "westus",
    "tags": {
        "department": "finance"
    },
    "zones": [
        "1"
    ],
    "properties": {
        "sharingProfile": {
            "subscriptionIds": [
                {
                    "id": "/subscriptions/{consumer-subscription-id1}"
                },
                {
                    "id": "/subscriptions/{consumer-subscription-id2}"
                },
                {
                    "id": "/subscriptions/{consumer-subscription-id3}"
                }
            ]
        }
    }
}
```

To learn more, see Capacity Reservation Groups - Create Or Update.

Stop sharing a Capacity Reservation Group

The Capacity Reservation Group owner can stop sharing a Capacity Reservation Group with a subscription or all subscriptions at any time.

Unsharing of a Capacity Reservation Group with a cross subscription ID is done while a VM or Virtual Machine Scale Set from cross subscriptions remains associated with the Capacity Reservation Group. The associated resources get SLA until deallocation or reallocation.

Once unsharing happens, any VM or scale set previously associated to the CRG would fail to associate upon deallocation or reallocation. Avoid this failure by removing the association from the CRG.

Unsharing a Capacity Reservation Group with a subscription

To unshare a Capacity Reservation Group with a subscription from the sharing profile, remove the subscription from the sharing profile.

In the following examples, a Capacity Reservation Group was shared with Consumer Subscription ID 1, Consumer Subscription ID 2, and Consumer Subscription ID 3.

API

To remove a subscription from the sharing profile of an existing Capacity Reservation Group, construct the following PUT request.

The following example removes Consumer Subscription ID 3 from the sharing profile to an existing Capacity Reservation Group called `myCapacityReservationGroup` that was previously shared with three subscription IDs.

rest

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/capacityReservationGroups/myCapacityReservationGroup?api-version=2024-03-01
```

In the request body, include the subscription ID to share the Capacity Reservation Group in the `sharing profile`.

JSON

```
{
    "location": "westus",
    "tags": {
        "department": "finance"
    },
    "zones": [
        "1"
    ],
    "properties": {
        "sharingProfile": {
            "subscriptionIds": [
                {
                    "id": "/subscriptions/{consumer-subscription-id1}"
                },
                {
                    "id": "/subscriptions/{consumer-subscription-id2}"
                }
            ]
        }
    }
}
```

To learn more, see Capacity Reservation Groups - Create Or Update.

Unsharing a Capacity Reservation Group with all subscriptions

To unshare a Capacity Reservation Group with all consumer subscriptions, remove all subscriptions from the sharing profile.

API

To remove all consumer subscriptions from the sharing profile of an existing Capacity Reservation group, construct the following PUT request.

The following example removes all consumer subscriptions from the sharing profile to an existing CRG called `myCapacityReservationGroup`.

rest

```
PUT https://management.azure.com/subscriptions/{provider-subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/capacityReservationGroups/myCapacityReservationGroup?api-version=2024-03-01
```

In the request body, include the subscription ID to share the Capacity Reservation Group in the `sharing profile`.

JSON

```
{
    "location": "westus",
    "tags": {
        "department": "finance"
    },
    "zones": [
        "1"
    ],
    "properties": {
        "sharingProfile": {
            "subscriptionIds": []
        }
    }
}
```

To learn more, see Capacity Reservation Groups - Create Or Update.

Delete a shared Capacity Reservation Group

See Modify a Capacity Reservation for deletion guidance.

- Users with sufficient rights can delete the shared Capacity Reservation Group.
- Azure allows a Capacity Reservation Group to be deleted when all the member capacity reservations are deleted.
- Azure allows a Capacity Reservation to be deleted when no VMs are associated to the Capacity Reservation.
- Unsharing of a CRG with a shared subscription happens as part of the shared Capacity Reservation Group deletion process.

Using a shared Capacity Reservation Group

Once the Capacity Reservation Group is successfully shared, users with sufficient rights from a Consumer subscription can deploy a VM or Virtual Machine Scale Set in a shared Capacity Reservation Group.

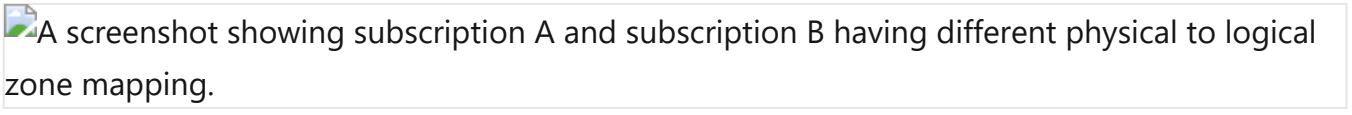
> **Note**
>
> The Provider subscription deploying the shared CRG will need to hold its own quota for deploying the CRG. The Consumer subscription making a deployment in the shared reservation group will need to hold its own quota.

Availability zone mapping with shared zonal Capacity Reservation Groups

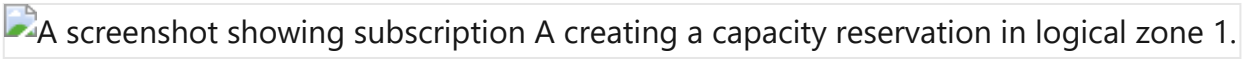
The availability zones visible to each Azure subscription represent a logical mapping to underlying groups of physical data centers that constitute a physical zone. To promote efficient distribution of resources across availability zones, each Azure subscription gets a random logical to physical mapping of zones. For example, subscription A (Provider subscription) and subscription B (Consumer subscription) can have different logical mappings.

Consider the following example:

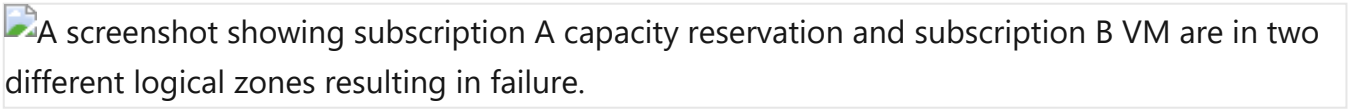
If subscription A deploys to AZ1, the deployment goes to physical zone 1. But if subscription B deploys to AZ1, the deployment goes to physical zone 2:



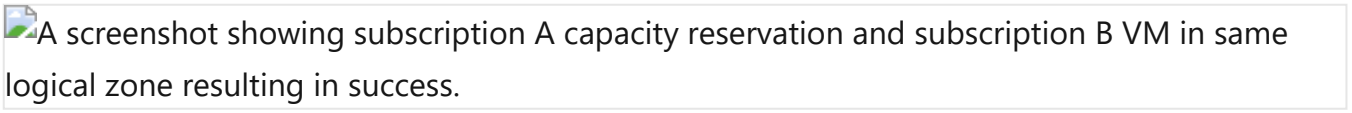
Now consider a Capacity Reservation deployed by Subscription A to logical AZ1. The result is reserved capacity in physical zone 1.



If subscription B deployed a VM to logical AZ1 using the shared Capacity Reservation Group, the deployment would fail because subscription B AZ1 resolves to physical zone 2, and there's no reserved capacity in physical zone 2.


A screenshot showing subscription A capacity reservation and subscription B VM are in two different logical zones resulting in failure.

The solution is to reconcile the logical to physical mappings for subscription A and subscription B. Subscription B should deploy to AZ2 to access reserved capacity in physical zone 1.


A screenshot showing subscription A capacity reservation and subscription B VM in same logical zone resulting in success.

When using a shared Capacity Reservation Group with zones, all subscriptions have the logical view of availability zones from the CRG subscription, which is likely different than the logical view of availability zones from the target subscription. When deploying a VM or scale set to a shared CRG, you must remap the zones. To ensure that the Azure resources are efficiently distributed across Availability zones in a Region, each subscription has independent logical zone mapping for Availability zones. This means that logical to physical zone mapping may or may not be the same across subscriptions.

To check the Physical Zone and Logical Zone mapping for your subscription, see Subscriptions - List Locations - REST API (Azure Resource Management). For more information, see Physical and Logical availability zones.

Secure existing zonal workloads using zero size reservation in Shared capacity Reservation Group

Zonally deployed virtual machines or virtual machine scale sets from Consumer subscription can be converted to using a shared Capacity Reservation without reallocation. If you do not have capacity in an existing shared capacity reservation group, you can start by creating a zero size matching reservation to secure your workloads. The basic process involves 3 steps:

1. Create a shared Capacity Reservation Group and then matching capacity reservations in each target zone with the reserved quantity set to zero. This requires no additional quota or capacity for the Provider subscription. For more information on how to create a reservation, see Create a capacity reservation.

2. Associate existing running zonal virtual machine or virtual machine scale set from the Consumer subscription to the shared capacity reservation group. Set the Virtual Machine or Virtual Machine Scale Set (VMSS) capacityReservationGroup property to the desired shared Capacity Reservation Group. When complete, each target capacity reservation will be overallocated.

3. Increase the reserved quantity of each capacity reservation (CR) to match the allocated Virtual Machine (VM) count. The Provider subscription must have sufficient quota to accomodate the workloads from the Consumer subscription for this step. See Using a shared Capacity Reservation Group. For more information on how to update the reserved count, see Capacity reservation modify

After the quantity increase, you should see the CR in a fully allocated state with all the virtual machines or virtual machine scale sets allocated. See View VM allocation with the Instance View.

For more information, see Zonal Virtual Machine or Zonal Virtual Machine Scale Set.

Use of Reserved Instances with shared Capacity Reservation Groups

Sharing a Capacity Reservation Group doesn't alter the scope of any Reserved Instances or Savings Plans. If either the CRG or the VM is deployed from a scope not covered by prepaid discounts, the pay-as-you-go price is charged.

To share Reserved Instance discounts between a Capacity Reservation Group and VMs deployed from a Consumer subscription, the Provider subscription and the Consumer subscription must share the same Reserved Instance scope. If the two subscriptions share an enrollment or a management group, then Reserved Instances set to the corresponding scope works automatically.

Associate or create a single Virtual Machine with shared Capacity Reservation Group

Single Virtual Machine can be deployed in shared Capacity Reservation Group using PowerShell, CLI, or REST API. See Associate a virtual machine to a Capacity Reservation group.

Remove a single Virtual Machine from Shared Capacity Reservation Group

Single Virtual Machine can be removed from Shared Capacity Reservation Group using PowerShell, CLI, or REST API. See Remove a virtual machine association from a Capacity Reservation group.

Associate or create a Virtual Machine Scale Set with shared Capacity Reservation Group

Virtual Machine Scale Set in Flexible and Uniform orchestration mode can be deployed in shared Capacity Reservation Group using PowerShell, CLI, or REST API. To learn more, see Associate a scale set -Flexible and Associate a scale set - Uniform.

Remove Virtual Machine Scale Set from Shared Capacity Reservation Group

Virtual Machine Scale Set in Flexible and Uniform orchestration mode can be removed from shared Capacity Reservation Group using PowerShell, CLI, or REST API. To learn more, see Remove a scale set.

View shared Capacity Reservation Group

Once a Capacity Reservation Group is shared successfully, the reservations are immediately available for use with single Virtual Machines and Virtual Machine Scale Sets.

View the subscription IDs the Capacity Reservation Group are shared with from the sharing profile.

To learn more, see Create a Capacity Reservation.

API

rest

```
GET https://management.azure.com/subscriptions/{provider-subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/capacityReservationGroups/{capacityReservationGroupName}?api-version=2023-07-01
```

JSON

```
{
    "name": "SharedCRG",
    "id": "/subscriptions/{Provider-subscriptionID1}/resourceGroups/{MyRG}/providers/Microsoft.Compute/capacityReservationGroups/SharedCRG",
    "type": "Microsoft.Compute/capacityReservationGroups",
    "location": "eastus2",
    "properties": {
        "capacityReservations": [{
                "id": "/subscriptions/{Provider-subscriptionID1}/resourceGroups/{MyRG}/providers/Microsoft.Compute/capacityReservationGroups/{SharedCRG}/capacityReservations/{CR1}"
            },
        ],
```

```
        "sharingProfile": {
            "subscriptionIds": [{
                    "id": "/subscriptions/{consumer-subscriptionID2}"
                }
            ]
        },
        "provisioningState": "Succeeded"
    }
}
```

To learn more, see Capacity Reservation Group-GET.

View the list of Capacity Reservation Groups for a subscription

The list of all Capacity Reservation Groups that are created locally or shared with by other subscriptions can be viewed for a given subscription. Extra parameter `resourceIdsonly` needs to be passed to view the shared Capacity Reservation Groups.

Capacity Reservation Groups - List by Subscription ID

By default, obtaining a list of Capacity Reservation Groups returns only those owned by the subscription. To add the Capacity Reservation Groups shared to the subscription, the additional parameter of `resourceIdsOnly` must be set to `sharedwithsubscription`.

API

Enable fetching Resource IDs for all CRG resources shared with the subscription:

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.
Compute/capacityReservationGroups?api-version=2023-09-
01&resourceIdsOnly=sharedwithsubscription
```

Enable fetching Resource IDs for all CRG resources shared with the subscription and created in the subscription:

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.
Compute/capacityReservationGroups?api-version=2023-09-01&resourceIdsOnly=All
```

Enable fetching Resource IDs for all CRG resources created in the subscription:

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.
Compute/capacityReservationGroups?api-version=2023-09-
01&resourceIdsOnly=CreatedInSubscription
```

JSON


```
{
    "value": [
        {
            "id":
"/subscriptions/{subscriptionId1}/resourceGroups/{resourceGroupName1}
/providers/Microsoft.Compute/capacityReservationGroups/{CapacityReservationGroup
Name1} ",
            "type": "Microsoft.Compute/capacityReservationGroups",
            "location": "EastUS2"
        },
        {
            "id":
"/subscriptions/{subscriptionId2}/resourceGroups/{resourceGroupName2}
/providers/Microsoft.Compute/capacityReservationGroups/{CapacityReservationGroup
Name2} ",
            "type": "Microsoft.Compute/capacityReservationGroups",
            "location": "EastUS2"
        }
    ]
}
```


To learn more, see Capacity Reservation Group - List by subscription.

Azure Resource Graph

Use the Azure Resource Graph to view the list of all Capacity Reservation Groups that are shared with created locally within a given subscription.

Portal

To view the CRG list, go to Azure Resource Graph Explorer and try this query.

Enable fetching Resource IDs for all CRG resources shared with subscription ID 1 and created within subscription ID 1:

Kusto

```
resources
|where type == "microsoft.compute/capacityreservationgroups"
|where properties["sharingProfile"] contains "{subscriptionId1}" or
subscriptionId == "{subscriptionId1}"
|project name, id
```

Enable fetching Resource IDs for all CRG resources shared with subscription ID 1:

Kusto

```
resources
|where type == "microsoft.compute/capacityreservationgroups"
|where properties["sharingProfile"] contains "{subscriptionId}"
|project name, id
```

Next steps

Learn how to associate a VM to a capacity reservation group

Last updated on 11/25/2025

Associate a VM to a capacity reservation group

Applies to: ✔ Windows Virtual Machines ✔ Linux Virtual Machines

You can use capacity reservation groups with new or existing virtual machines (VMs). To learn more about capacity reservations, see the Capacity reservation overview.

Associate a new virtual machine

To associate a new virtual machine to the capacity reservation group, the group must be explicitly referenced as a property of the VM. This reference protects the matching reservation in the group for applications and workloads intended to use it.

API

To add the `capacityReservationGroup` property to a VM, construct the following `PUT` request to the `Microsoft.Compute` provider:

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{VirtualMachineName}?api-version=2021-04-01
```

In the request body, include the `capacityReservationGroup` property:

JSON

```
{
  "location": "eastus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    …
   "capacityReservation":{
    "capacityReservationGroup":{

"id":"subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{CapacityReservationGroupName}"
    }
    "storageProfile": {
    …
    },
```

```
    "osProfile": {
    …
    },
    "networkProfile": {
     …
    }
  }
}
```

Associate an existing virtual machine

Existing zonal virtual machines can simply be updated with Capacity Reservation Group property without the need of deallocation and it will automatically consume the capacity reservation. Existing regional virtual machines must be reallocated.

Regional Virtual Machine

For the initial release of a capacity reservation, a virtual machine must be allocated to a capacity reservation:

1. Follow guidance to create a capacity reservation group and capacity reservation, if necessary. Or increment the quantity of an existing capacity reservation so there's unused reserved capacity.
2. Deallocate the virtual machine.
3. Update the capacity reservation group property on the VM.
4. Restart the VM.

Follow the steps to associate the virtual machine to a capacity reservation group.

API

1. Deallocate the virtual machine:

   rest

   ```
   PUT
   https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
   {resourcegroupname}/providers/Microsoft.Compute/virtualMachines/{VirtualMac
   hineName}/deallocate?api-version=2021-04-01
   ```

2. Add the `capacityReservationGroup` property to the VM. Construct the following `PUT` request to `Microsoft.Compute` provider:

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{VirtualMac
hineName}?api-version=2021-04-01
```

In the request body, include the `capacityReservationGroup` property:

JSON

```
{
"location": "eastus",
"properties": {
    "capacityReservation": {
        "capacityReservationGroup": {
            "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/provide
rs/Microsoft.Compute/capacityReservationGroups/{capacityReservationGroupNam
e}"
        }
    }
}
}
```

Zonal Virtual Machine

A zonal virtual machine can be allocated to a capacity reservation without the need of reallocation. If you have capacity in an existing capacity reservation group, you can add an existing zonal virtual machine to the capacity reservation group and update the VM to use a matching capacity reservation in the capacity reservation group.

> **ⓘ Important**
>
> The feature to associate an existing zonal VM to a Capacity Reservation Group without the need to deallocate, is currently in **Preview**. See the Preview Terms of Use for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

1. Follow guidance to create a capacity reservation group and capacity reservation, if necessary. Or increment the quantity of an existing capacity reservation so there's unused reserved capacity.
2. Update the capacity reservation group property on the VM.

Follow the steps to associate the virtual machine to a capacity reservation group.

API

Add the `capacityReservationGroup` property to the VM. Construct the following `PUT` request to `Microsoft.Compute` provider:

rest

```
    PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{reso
urceGroupName}/providers/Microsoft.Compute/virtualMachines/{VirtualMachineName}?
api-version=2021-04-01
```

In the request body, include the `capacityReservationGroup` property:

JSON

```
    {
    "location": "eastus",
    "properties": {
        "capacityReservation": {
            "capacityReservationGroup": {
                "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Mi
crosoft.Compute/capacityReservationGroups/{capacityReservationGroupName}"
            }
        }
    }
    }
```

Secure existing zonal virtual machine using zero size reservation

Zonally deployed virtual machines can be converted to using an On Demand Capacity Reservation without reallocation. If you do not have capacity in an existing capacity reservation group, you can start by creating a zero size matching reservation to secure your workloads. The basic process involves 3 steps:

1. Create a Capacity Reservation Group and then matching capacity reservations in each target zone with the reserved quantity set to zero. This requires no additional quota or capacity. For more information on how to create a reservation, see Create a capacity reservation.

2. Associate existing running zonal virtual machine to the capacity reservation. Set the Virtual Machine capacityReservationGroup property to the desired Capacity Reservation Group. When complete, each target capacity reservation will be overallocated. See Zonal Virtual Machine.

3. Increase the reserved quantity of each capacity reservation (CR) to match the allocated Virtual Machine (VM) count. Since each CR is already overallocated, this step requires no additional quota or capacity. For more information on how to update the reserved count, see Capacity reservation modify.

After the quantity increase, you should see the CR in a fully allocated state with all the virtual machines allocated. See View VM allocation with the Instance View.

For information on using shared capacity reservation group, see Secure existing zonal workloads using zero size reservation in Shared capacity Reservation Group.

View VM allocation with the Instance View

After the `capacityReservationGroup` property is set, an association now exists between the VM and the group. Azure automatically finds the matching capacity reservation in the group and consumes a reserved slot. The capacity reservation's Instance View reflects the new VM in the `virtualMachinesAllocated` property:

API

rest

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{reso
urceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{capacityRe
servationGroupName}?$expand=instanceView&api-version=2021-04-01
```

JSON

```
{
   "name":"{CapacityReservationGroupName}",

"id":"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/provide
rs/Microsoft.Compute/capacityReservationGroups/{CapacityReservationGroupName}",
   "type":"Microsoft.Compute/capacityReservationGroups",
   "location":"eastus",
   "properties":{
      "capacityReservations":[
         {
```

```
"id":"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/capacityReservationGroups/{CapacityReservationGroupName}/capacityReservations/{CapacityReservationName}"
         }
      ],
      "virtualMachinesAssociated":[
         {

"id":"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{myVM}"
         }
      ],
      "instanceView":{
         "capacityReservations":[
            {
               "name":"{CapacityReservationName}",
               "utilizationInfo":{
                  "virtualMachinesAllocated":[
                     {

"id":"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{myVM}"
                     }
                  ]
               },
               "statuses":[
                  {
                     "code":"ProvisioningState/succeeded",
                     "level":"Info",
                     "displayStatus":"Provisioning succeeded",
                     "time":"2021-05-25T15:12:10.4165243+00:00"
                  }
               ]
            }
         ]
      }
   }
}
```


Next step

Remove a VM's association to a capacity reservation group

Last updated on 11/24/2025

Remove a VM association from a capacity reservation group

This article walks you through the steps of removing a virtual machine (VM) association to a capacity reservation group. To learn more about capacity reservations, see the Capacity reservation overview.

Because both the VM and the underlying capacity reservation logically occupy capacity, Azure imposes some constraints on this process to avoid ambiguous allocation states and unexpected errors.

There are three ways to change an association:

- Deallocate the virtual machine, change the capacity reservation group property, and, optionally, restart the VM.
- Update the reserved quantity to zero and then change the capacity reservation group property.
- Delete a VM.

Deallocate the virtual machine

The first option is to deallocate the virtual machine, change the capacity reservation group property, and, optionally, restart the VM.

API

1. Deallocate the virtual machine:

```rest
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{virtualMachineName}/deallocate?api-version=2021-04-01
```

2. Update the VM to remove the association with the capacity reservation group:

```rest
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{virtualMachineName}/update?api-version=2021-04-01
```

In the request body, set the `capacityReservationGroup` property to `null` to remove the VM association to the group:

JSON

```
 {
"location": "eastus",
"properties": {
    "capacityReservation": {
        "capacityReservationGroup": {
            "id":null
        }
    }
}
}
```

Update the reserved quantity to zero

The second option involves updating the reserved quantity to zero and then changing the capacity reservation group property.

This option works well when the VM can't be deallocated and when a reservation is no longer needed. For example, you might create a capacity reservation to temporarily assure capacity during a large-scale deployment. After it's finished, the reservation is no longer needed.

API

1. Update the reserved quantity to zero:

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{
CapacityReservationGroupName}/CapacityReservations/{CapacityReservationName
}?api-version=2021-04-01
```

In the request body, include the following parameters:

JSON

```
{
"sku":
    {
```

```
    "capacity": 0
    }
}
```

Note that the `capacity` property is set to `0`.

2. Update the VM to remove the association with the capacity reservation group.

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{VirtualMac
hineName}/update?api-version=2021-04-01
```

In the request body, set the `capacityReservationGroup` property to `null` to remove the association:

JSON

```
{
"location": "eastus",
"properties": {
    "capacityReservation": {
        "capacityReservationGroup": {
            "id":null
        }
    }
}
}
```

Delete a VM

The virtual machine deletion process will remove a VM association from a capacity reservation. A VM delete must complete before Azure removes it from the capacity reservation. Some latency can occur between the delete request and the corresponding change in capacity reservation allocation state. See Delete a VM for more information. Use Capacity Reservation Instance View to check the allocation status as needed.

Next step

Learn how to associate a scale set to a capacity reservation group

Last updated on 10/24/2025

Associate a virtual machine scale set with Flexible Orchestration to a capacity reservation group

Article • 09/09/2024

Applies to: ✔ Flexible scale sets

Azure Virtual Machine Scale Sets has two modes:

- **Uniform Orchestration**: In this mode, virtual machine scale sets use a virtual machine (VM) profile or a template to scale up to the capacity you want. Although there's some ability to manage or customize individual VM instances, Uniform Orchestration uses identical VM instances. These instances are exposed through the virtual machine scale set's VM APIs. They aren't compatible with the API commands that are standard for Azure infrastructure as a service (IaaS) VMs. Because the scale set performs all the actual VM operations, reservations are associated to the virtual machine scale set directly. After the scale set is associated to the reservation, all the subsequent VM allocations are done against the reservation.
- **Flexible Orchestration**: In this mode, you get more flexibility to manage the individual instances of virtual machine scale set VMs. They can use the standard Azure IaaS VM APIs instead of using the scale set interface. To use reservations with Flexible Orchestration mode, define both the virtual machine scale set property and the capacity reservation property on each VM.

To learn more about these modes, see Orchestration modes for Virtual Machine Scale Sets.

This content applies to the Flexible Orchestration mode. For more information about Uniform Orchestration mode, see Associate a virtual machine scale set with Uniform Orchestration to a capacity reservation group.

Associate a new virtual machine scale set to a capacity reservation group

You have two options:

- **Add to the VM profile:** If the scale set with Flexible Orchestration includes a VM profile, add the capacity reservation group property to the profile during scale set

creation. Follow the same process that's used for a scale set that uses Uniform Orchestration. For sample code, see Associate a virtual machine scale set with Uniform Orchestration to a capacity reservation group.

- **Add to the first deployed VM:** If the scale set omits a VM profile, you must add the capacity reservation group to the first VM that you deployed by using the virtual machine scale set. Follow the same process that's used to associate a VM. For sample code, see Associate a VM to a capacity reservation group.

Associate an existing virtual machine scale set to a capacity reservation group

There are two steps:

- **Add to the virtual machine scale set:** For sample code, see Associate a virtual machine scale set with Uniform Orchestration to a capacity reservation group.
- **Add to the deployed VMs:** You must add the capacity reservation group to the VMs that you deployed by using the scale set, depending on the upgrade mode. Follow the same process used to associate a VM. For sample code, see Associate a VM to a capacity reservation group.

Next step

Learn how to remove a scale set association from a capacity reservation

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Associate a virtual machine scale set to Uniform Orchestration to a capacity reservation group

Applies to: ✔ Uniform scale set

Azure Virtual Machine Scale Sets has two modes:

- **Uniform Orchestration**: In this mode, virtual machine scale sets use a virtual machine (VM) profile or a template to scale up to the capacity you want. Although there's some ability to manage or customize individual VM instances, Uniform Orchestration uses identical VM instances. These instances are exposed through the virtual machine scale set's VM APIs and aren't compatible with the API commands that are standard for Azure infrastructure as a service (IaaS) VMs. Because the scale set performs all the actual VM operations, reservations are associated to the virtual machine scale set directly. After the scale set is associated to the reservation, all the subsequent VM allocations are done against the reservation.
- **Flexible Orchestration**: In this mode, you get more flexibility to manage the individual virtual machine scale set VM instances. They can use the standard Azure IaaS VM APIs instead of by using the scale set interface. To use reservations with Flexible Orchestration mode, define both the virtual machine scale set property and the capacity reservation property on each VM.

To learn more about these modes, see Virtual Machine Scale Sets orchestration modes.

This content applies to the Uniform Orchestration mode. For Flexible Orchestration mode, see Associate a virtual machine scale set with Flexible Orchestration to a capacity reservation group.

Limitations of scale sets in Uniform Orchestration

- For virtual machine scale sets in Uniform Orchestration to be compatible with capacity reservation, the `singlePlacementGroup` property must be set to `False`.
- The **Static Fixed Spreading** availability option for multizone uniform scale sets isn't supported with capacity reservation. This option requires the use of five fault domains. However, the reservations only support up to three fault domains for general purpose sizes. The approach we recommend is to use the **Max Spreading** option that spreads VMs across as many fault domains as possible within each zone. If needed, configure a custom fault domain configuration of three or less.

There are some other restrictions when you use capacity reservations. For the complete list, see the overview of capacity reservations.

Associate a new virtual machine scale set to a capacity reservation group

> **Important**
>
> Starting November 2023, virtual machine scale sets created by using PowerShell and the Azure CLI default to Flexible Orchestration mode if no orchestration mode is specified. For more information about this change and what actions you should take, see Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub.

API

To associate a new uniform virtual machine scale set to a capacity reservation group, construct the following `PUT` request to the `Microsoft.Compute` provider:

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{VMScaleSetName}?api-version=2021-04-01
```

Add the `capacityReservationGroup` property in the `virtualMachineProfile` property:

JSON

```
{
    "name": "<VMScaleSetName>",
    "id": "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{VMScaleSetName}",
    "type": "Microsoft.Compute/virtualMachineScaleSets",
    "location": "eastus",
    "sku": {
        "name": "Standard_D2s_v3",
        "tier": "Standard",
        "capacity": 3
},
"properties": {
    "virtualMachineProfile": {
        "capacityReservation": {
            "capacityReservationGroup":{
```

```
"id":"subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroup/{CapacityReservationGroupName}"
            }
         },
        "osProfile": {
            …
        },
        "storageProfile": {
            …
        },
        "networkProfile": {
            …,
            "extensionProfile": {
                …
            }
        }
    }
```

Associate an existing virtual machine scale set to a capacity reservation group

Existing zonal virtual machine scale set can simply be updated with Capacity Reservation Group property without the need of deallocation. Existing regional virtual machine scale set can include similar process but must be reallocated.

Regional Virtual Machine Scale Set

To add an existing regional capacity reservation group to an existing uniform scale set:

- Stop the scale set to deallocate the VM instances.
- Update the scale set to use a matching capacity reservation group.
- Start the scale set.

This process ensures that the placement for the capacity reservations and scale set in the region are compatible.

Important notes on upgrade policies

- **Automatic upgrade**: In this mode, the scale set VM instances are automatically associated to the capacity reservation group without any further action from you. When the scale set VMs are reallocated, they start consuming the reserved capacity.
- **Rolling upgrade**: In this mode, scale set VM instances are associated to the capacity reservation group without any further action from you. However, they're updated in

batches with an optional pause time between them. When the scale set VMs are reallocated, they start consuming the reserved capacity.

- **Manual upgrade**: In this mode, nothing happens to the scale set VM instances when the virtual machine scale set is attached to a capacity reservation group. You need to update to each scale set VM by upgrading it with the latest scale set model.

API

1. Deallocate the virtual machine scale set:

   rest

   ```
   POST
   https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
   {resourcegroupname}/providers/Microsoft.Compute/virtualMachineScaleSets/{VM
   ScaleSetName}/deallocate?api-version=2021-04-01
   ```

2. Add the `capacityReservationGroup` property to the scale set model. Construct the following `PUT` request to `Microsoft.Compute` provider:

   rest

   ```
   PUT
   https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
   {resourcegroupname}/providers/Microsoft.Compute/virtualMachineScaleSets/{VM
   ScaleSetName}?api-version=2021-04-01
   ```

   In the request body, include the `capacityReservationGroup` property:

   JSON

   ```
   "location": "eastus",
   "properties": {
       "virtualMachineProfile": {
            "capacityReservation": {
                     "capacityReservationGroup": {
                           "id":
   "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/provide
   rs/Microsoft.Compute/capacityReservationGroups/{capacityReservationGroupNam
   e}"
                     }
               }
       }
   }
   ```

Zonal Virtual Machine Scale Set

If you have capacity in an existing capacity reservation group and you want to add an existing zonal uniform scale set to it, simply update the scale set to the capacity reservation group to use a matching capacity reservation.

> **Important**
>
> The feature to associate an existing zonal scale set to a reservation group without the need to deallocate, is currently in **Preview**. See the Preview Terms of Use for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability. The feature is currently not available in Microsoft Azure Government and Microsoft Azure operated by 21Vianet.

Important notes on upgrade policies

- **Automatic upgrade**: In this mode, the scale set VM instances are automatically associated to the capacity reservation group without any further action from you. If already running, the scale set VM instances start consuming the reserved capacity. If the existing scale set VMs are in deallocated state, they start consuming the reserved capacity when reallocated.
- **Rolling upgrade**: In this mode, scale set VM instances are associated to the capacity reservation group without any further action from you. However, they're updated in batches with an optional pause time between them. If already running, the scale set VM instances start consuming the reserved capacity. If the existing scale set VMs are in deallocated state, they start consuming the reserved capacity when reallocated.
- **Manual upgrade**: In this mode, nothing happens to the scale set VM instances when the virtual machine scale set is attached to a capacity reservation group. You need to update to each scale set VM by upgrading it with the latest scale set model.

API

Add the `capacityReservationGroup` property to the scale set model. Construct the following `PUT` request to `Microsoft.Compute` provider:

rest

```
    PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{reso
urcegroupname}/providers/Microsoft.Compute/virtualMachineScaleSets/{VMScaleSetNa
me}?api-version=2021-04-01
```

In the request body, include the `capacityReservationGroup` property:

JSON

```
    "location": "eastus",
    "properties": {
        "virtualMachineProfile": {
             "capacityReservation": {
                      "capacityReservationGroup": {
                            "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Mi
crosoft.Compute/capacityReservationGroups/{capacityReservationGroupName}"
                      }
                 }
        }
    }
```

Secure existing zonal virtual machine scale sets using zero size reservation

Zonally deployed virtual machines scale sets can be converted to using an On Demand Capacity Reservation without reallocation. If you do not have capacity in an existing capacity reservation group, you can start by creating a zero size matching reservation to secure your workloads. The basic process involves 3 steps:

1. Create a Capacity Reservation Group and then matching capacity reservations in each target zone with the reserved quantity set to zero. This requires no additional quota or capacity. For more information on how to create a reservation, see Create a capacity reservation.

2. Associate existing running zonal virtual machine scale set to the capacity reservation. Set the Virtual Machine Scale Set (VMSS) capacityReservationGroup property to the desired Capacity Reservation Group, then execute a model update on the VMSS (VMSS uniform) or update each member VM with the capacityReservationGroup property (VMSS Flex). When complete, each target capacity reservation will be overallocated See Zonal Virtual Machine Scale Set

3. Increase the reserved quantity of each capacity reservation (CR) to match the allocated Virtual Machine (VM) count. Since each CR is already overallocated, this step requires no additional quota or capacity. For more information on how to update the reserved count, see Capacity reservation modify

After the quantity increase, you should see the CR in a fully allocated state with all the VMSS virtual machines allocated. See View VM allocation with the Instance View.

For information on using shared capacity reservation group, see Secure existing zonal workloads using zero size reservation in Shared capacity Reservation Group.

Region and availability zone considerations

You can create virtual machine scale sets regionally or in one or more availability zones to help protect them from datacenter-level failure. To learn more about multizonal virtual machine scale sets, see Virtual machine scale sets that use availability zones.

> **Important**
>
> The location (region and availability zones) of the virtual machine scale set and the capacity reservation group must match for the association to succeed. For a regional scale set, the region must match between the scale set and the capacity reservation group. For a zonal scale set, both the regions and the zones must match between the scale set and the capacity reservation group.

When a scale set is spread across multiple zones, it always attempts to deploy evenly across the included availability zones. Because of that even deployment, a capacity reservation group should always have the same quantity of reserved VMs in each zone. As an illustration of why this even deployment is important, consider the following example.

In this example, each zone has a different quantity reserved. Let's say that the virtual machine scale set scales out to 75 instances. Because a scale set always attempts to deploy evenly across zones, the VM distribution should look like this example:

Expand table

Zone	Quantity reserved	Number of scale set VMs in each zone	Unused quantity reserved	Overallocated
1	40	25	15	0
2	20	25	0	5
3	15	25	0	10

In this case, the scale set incurs extra cost for 15 unused instances in Zone 1. The scale-out is also relying on 5 VMs in Zone 2 and 10 VMs in Zone 3 that aren't protected by capacity reservation. If each zone had 25 capacity instances reserved, then all 75 VMs would be protected by capacity reservation and the deployment wouldn't incur any extra cost for unused instances.

Because the reservations can be overallocated, the scale set can continue to scale normally beyond the limits of the reservation. The only difference is that the VMs allocated above the quantity reserved aren't covered by capacity reservation service-level agreement. To learn more, see Overallocate capacity reservation.

Next step

Learn how to remove a scale set association from a capacity reservation

---

Last updated on 11/24/2025

Remove a virtual machine scale set association from a capacity reservation group

Applies to: ✔ Uniform scale set ✔ Flexible scale sets

This article walks you through removing a virtual machine scale set association from a capacity reservation group. To learn more about capacity reservations, see the overview of capacity reservations.

Because both the virtual machine (VM) and the underlying capacity reservation logically occupy capacity, Azure imposes some constraints on this process to avoid ambiguous allocation states and unexpected errors.

There are three ways to change an association:

- Deallocate the virtual machine scale set, change the capacity reservation group property at the scale set level, and then update the underlying VMs.
- Update the reserved quantity to zero and then change the capacity reservation group property.
- Delete the virtual machine scale set

Deallocate the virtual machine scale set

The first option is to deallocate the virtual machine scale set, change the capacity reservation group property at the scale set level, and then update the underlying VMs.

For more information about automatic, rolling, and manual upgrades, see Upgrade policies.

API

1. Deallocate the virtual machine scale set:

   rest

   ```
   POST
   https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
   {resourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{VM
   ScaleSetName}/deallocate?api-version=2021-04-01
   ```

2. Update the virtual machine scale set to remove the association with the capacity reservation group:

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{VM
ScaleSetName}/update?api-version=2021-04-01
```

In the request body, set the `capacityReservationGroup` property to `null` to remove the virtual machine scale set association to the group:

JSON

```
{
"location": "eastus",
"properties": {
    "virtualMachineProfile": {
        "capacityReservation": {
            "capacityReservationGroup":{
                "id":null
            }
        }
    }
}
}
```

Update the reserved quantity to zero

The second option involves updating the reserved quantity to zero and then changing the capacity reservation group property.

This option works well when the scale set can't be deallocated and when a reservation is no longer needed. For example, you might create a capacity reservation to temporarily assure capacity during a large-scale deployment. After deployment is finished, the reservation is no longer needed.

For more information about automatic, rolling, and manual upgrades, see Upgrade policies.

API

1. Update the reserved quantity to zero:

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
```

```
{resourceGroupName}/providers/Microsoft.Compute/CapacityReservationGroups/{
CapacityReservationGroupName}/CapacityReservations/{CapacityReservationName
}?api-version=2021-04-01
```

In the request body, include the following parameters:

JSON

```
{
"sku":
    {
    "capacity": 0
    }
}
```

Note that the `capacity` property is set to `0`.

2. Update the virtual machine scale set to remove the association with the capacity reservation group.

rest

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.Compute/virtualMachineScaleSets/{VM
ScaleSetName}/update?api-version=2021-04-01
```

In the request body, set the `capacityReservationGroup` property to `null` to remove the association:

JSON

```
{
"location": "eastus",
"properties": {
    "virtualMachineProfile": {
        "capacityReservation": {
            "capacityReservationGroup":{
                "id":null
            }
        }
    }
}
}
```

Delete the virtual machine scale set

Deletion process of a scale set will remove its association from a capacity reservation. The delete request must complete before Azure removes it from the capacity reservation. Some latency can occur between the delete request and the corresponding change in capacity reservation allocation state. See Delete a VM for more information. Use Capacity Reservation Instance View to check the allocation status as needed.

Upgrade policies

- **Automatic upgrade**: In this mode, the scale set VM instances are automatically dissociated from the capacity reservation group without any further action from you.
- **Rolling upgrade**: In this mode, the scale set VM instances are dissociated from the capacity reservation group without any further action from you. However, they're updated in batches with an optional pause time between them.
- **Manual upgrade**: In this mode, nothing happens to the scale set VM instances when the virtual machine scale set is updated. You need to individually remove each scale set VM by upgrading it with the latest scale set model.

Next step

Learn about overallocating a capacity reservation

Last updated on 10/24/2025

Create a complete Linux virtual machine with the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs

To quickly create a virtual machine (VM) in Azure, you can use a single Azure CLI command that uses default values to create any required supporting resources. Resources such as a virtual network, public IP address, and network security group rules are automatically created. For more control of your environment in production use, you may create these resources ahead of time and then add your VMs to them. This article guides you through how to create a VM and each of the supporting resources one by one.

Make sure that you have installed the latest Azure CLI and logged to an Azure account in with az login.

In the following examples, replace example parameter names with your own values. Example parameter names include *myResourceGroup*, *myVnet*, and *myVM*.

Create resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine and supporting virtual network resources. Create the resource group with az group create. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

By default, the output of Azure CLI commands is in JSON (JavaScript Object Notation). To change the default output to a list or table, for example, use az config set core.output=table. You can also add `--output` to any command for a one time change in output format. The following example shows the JSON output from the `az group create` command:

JSON

```
{
  "id": "/subscriptions/guid/resourceGroups/myResourceGroup",
  "location": "eastus",
  "name": "myResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```


Create a virtual network and subnet

Next you create a virtual network in Azure and a subnet in to which you can create your VMs. Use az network vnet create to create a virtual network named *myVnet* with the *192.168.0.0/16* address prefix. You also add a subnet named *mySubnet* with the address prefix of *192.168.1.0/24*:

Azure CLI

```
az network vnet create \
    --resource-group myResourceGroup \
    --name myVnet \
    --address-prefix 192.168.0.0/16 \
    --subnet-name mySubnet \
    --subnet-prefix 192.168.1.0/24
```

The output shows the subnet is logically created inside the virtual network:

JSON


```
{
  "addressSpace": {
    "addressPrefixes": [
      "192.168.0.0/16"
    ]
  },
  "dhcpOptions": {
    "dnsServers": []
  },
  "etag": "W/\"e95496fc-f417-426e-a4d8-c9e4d27fc2ee\"",
  "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/virtualNetworks/myVnet",
  "location": "eastus",
  "name": "myVnet",
  "provisioningState": "Succeeded",
  "resourceGroup": "myResourceGroup",
  "resourceGuid": "ed62fd03-e9de-430b-84df-8a3b87cacdbb",
```

```
  "subnets": [
    {
      "addressPrefix": "192.168.1.0/24",
      "etag": "W/\"e95496fc-f417-426e-a4d8-c9e4d27fc2ee\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/virtualNetworks/myVnet/subnets/mySubnet",
      "ipConfigurations": null,
      "name": "mySubnet",
      "networkSecurityGroup": null,
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null
    }
  ],
  "tags": {},
  "type": "Microsoft.Network/virtualNetworks",
  "virtualNetworkPeerings": null
}
```


Create a public IP address

Now let's create a public IP address with az network public-ip create. This public IP address enables you to connect to your VMs from the Internet. Because the default address is dynamic, create a named DNS entry with the `--domain-name-label` parameter. The following example creates a public IP named *myPublicIP* with the DNS name of *mypublicdns*. Because the DNS name must be unique, provide your own unique DNS name:

Azure CLI

```
az network public-ip create \
    --resource-group myResourceGroup \
    --name myPublicIP \
    --dns-name mypublicdns
```

Output:

JSON


```
{
  "publicIp": {
    "dnsSettings": {
      "domainNameLabel": "mypublicdns",
      "fqdn": "mypublicdns.eastus.cloudapp.azure.com",
      "reverseFqdn": null
    },
```

```
    "etag": "W/\"2632aa72-3d2d-4529-b38e-b622b4202925\"",
    "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/publicIPAddresses/myPublicIP",
    "idleTimeoutInMinutes": 4,
    "ipAddress": null,
    "ipConfiguration": null,
    "location": "eastus",
    "name": "myPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Dynamic",
    "resourceGroup": "myResourceGroup",
    "resourceGuid": "4c65de38-71f5-4684-be10-75e605b3e41f",
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses"
  }
}
```


Create a network security group

To control the flow of traffic in and out of your VMs, apply a network security group to a virtual NIC or subnet. The following example uses az network nsg create to create a network security group named *myNetworkSecurityGroup*:

Azure CLI

```
az network nsg create \
    --resource-group myResourceGroup \
    --name myNetworkSecurityGroup
```

You define rules that allow or deny specific traffic. To allow inbound connections on port 22 (to enable SSH access, create an inbound rule with az network nsg rule create. The following example creates a rule named *myNetworkSecurityGroupRuleSSH*:

Azure CLI

```
az network nsg rule create \
    --resource-group myResourceGroup \
    --nsg-name myNetworkSecurityGroup \
    --name myNetworkSecurityGroupRuleSSH \
    --protocol tcp \
    --priority 1000 \
    --destination-port-range 22 \
    --access allow
```

To allow inbound connections on port 80 (for web traffic), add another network security group rule. The following example creates a rule named *myNetworkSecurityGroupRuleHTTP*:

Azure CLI

```
az network nsg rule create \
    --resource-group myResourceGroup \
    --nsg-name myNetworkSecurityGroup \
    --name myNetworkSecurityGroupRuleWeb \
    --protocol tcp \
    --priority 1001 \
    --destination-port-range 80 \
    --access allow
```

Examine the network security group and rules with az network nsg show:

Azure CLI

```
az network nsg show --resource-group myResourceGroup --name
myNetworkSecurityGroup
```

Output:

JSON


```
{
  "defaultSecurityRules": [
    {
      "access": "Allow",
      "description": "Allow inbound traffic from all VMs in VNET",
      "destinationAddressPrefix": "VirtualNetwork",
      "destinationPortRange": "*",
      "direction": "Inbound",
      "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/defaultSecurityRules/AllowV
netInBound",
      "name": "AllowVnetInBound",
      "priority": 65000,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "VirtualNetwork",
      "sourcePortRange": "*"
    },
    {
      "access": "Allow",
      "description": "Allow inbound traffic from azure load balancer",
```

```
      "destinationAddressPrefix": "*",
      "destinationPortRange": "*",
      "direction": "Inbound",
      "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/defaultSecurityRules/AllowA
zureLoadBalancerInBou",
      "name": "AllowAzureLoadBalancerInBound",
      "priority": 65001,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "AzureLoadBalancer",
      "sourcePortRange": "*"
    },
    {
      "access": "Deny",
      "description": "Deny all inbound traffic",
      "destinationAddressPrefix": "*",
      "destinationPortRange": "*",
      "direction": "Inbound",
      "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/defaultSecurityRules/DenyAl
lInBound",
      "name": "DenyAllInBound",
      "priority": 65500,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "*",
      "sourcePortRange": "*"
    },
    {
      "access": "Allow",
      "description": "Allow outbound traffic from all VMs to all VMs in
VNET",
      "destinationAddressPrefix": "VirtualNetwork",
      "destinationPortRange": "*",
      "direction": "Outbound",
      "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/defaultSecurityRules/AllowV
netOutBound",
      "name": "AllowVnetOutBound",
      "priority": 65000,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "VirtualNetwork",
      "sourcePortRange": "*"
    },
```

```
    {
      "access": "Allow",
      "description": "Allow outbound traffic from all VMs to Internet",
      "destinationAddressPrefix": "Internet",
      "destinationPortRange": "*",
      "direction": "Outbound",
      "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/defaultSecurityRules/AllowI
nternetOutBound",
      "name": "AllowInternetOutBound",
      "priority": 65001,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "*",
      "sourcePortRange": "*"
    },
    {
      "access": "Deny",
      "description": "Deny all outbound traffic",
      "destinationAddressPrefix": "*",
      "destinationPortRange": "*",
      "direction": "Outbound",
      "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/defaultSecurityRules/DenyAl
lOutBound",
      "name": "DenyAllOutBound",
      "priority": 65500,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "*",
      "sourcePortRange": "*"
    }
  ],
  "etag": "W/\"3371b313-ea9f-4687-a336-a8ebdfd80523\"",
  "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup",
  "location": "eastus",
  "name": "myNetworkSecurityGroup",
  "networkInterfaces": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "myResourceGroup",
  "resourceGuid": "47a9964e-23a3-438a-a726-8d60ebbb1c3c",
  "securityRules": [
    {
      "access": "Allow",
      "description": null,
      "destinationAddressPrefix": "*",
      "destinationPortRange": "22",
```

```
      "direction": "Inbound",
      "etag": "W/\"9e344b60-0daa-40a6-84f9-0ebbe4a4b640\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/securityRules/myNetworkSecu
rityGroupRuleSSH",
      "name": "myNetworkSecurityGroupRuleSSH",
      "priority": 1000,
      "protocol": "Tcp",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "*",
      "sourcePortRange": "*"
    },
    {
      "access": "Allow",
      "description": null,
      "destinationAddressPrefix": "*",
      "destinationPortRange": "80",
      "direction": "Inbound",
      "etag": "W/\"9e344b60-0daa-40a6-84f9-0ebbe4a4b640\"",
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup/securityRules/myNetworkSecu
rityGroupRuleWeb",
      "name": "myNetworkSecurityGroupRuleWeb",
      "priority": 1001,
      "protocol": "Tcp",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "*",
      "sourcePortRange": "*"
    }
  ],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/networkSecurityGroups"
}
```


Create a virtual NIC

Virtual network interface cards (NICs) are programmatically available because you can apply rules to their use. Depending on the VM size, you can attach multiple virtual NICs to a VM. In the following az network nic create command, you create a NIC named *myNic* and associate it with your network security group. The public IP address *myPublicIP* is also associated with the virtual NIC.

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic \
    --vnet-name myVnet \
    --subnet mySubnet \
    --public-ip-address myPublicIP \
    --network-security-group myNetworkSecurityGroup
```

Output:

JSON


```
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix":
"brqlt10lvoxedgkeuomc4pm5tb.bx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\"04b5ab44-d8f4-422a-9541-e5ae7de8466d\"",
    "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkInterfaces/myNic",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "etag": "W/\"04b5ab44-d8f4-422a-9541-e5ae7de8466d\"",
        "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkInterfaces/myNic/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.4",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": {
          "dnsSettings": null,
          "etag": null,
          "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/publicIPAddresses/myPublicIP",
          "idleTimeoutInMinutes": null,
          "ipAddress": null,
          "ipConfiguration": null,
```

```
          "location": null,
          "name": null,
          "provisioningState": null,
          "publicIpAddressVersion": null,
          "publicIpAllocationMethod": null,
          "resourceGroup": "myResourceGroup",
          "resourceGuid": null,
          "tags": null,
          "type": null
        },
        "resourceGroup": "myResourceGroup",
        "subnet": {
          "addressPrefix": null,
          "etag": null,
          "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/virtualNetworks/myVnet/subnets/mySubnet",
          "ipConfigurations": null,
          "name": null,
          "networkSecurityGroup": null,
          "provisioningState": null,
          "resourceGroup": "myResourceGroup",
          "resourceNavigationLinks": null,
          "routeTable": null
        }
      }
    ],
    "location": "eastus",
    "macAddress": null,
    "name": "myNic",
    "networkSecurityGroup": {
      "defaultSecurityRules": null,
      "etag": null,
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Netw
ork/networkSecurityGroups/myNetworkSecurityGroup",
      "location": null,
      "name": null,
      "networkInterfaces": null,
      "provisioningState": null,
      "resourceGroup": "myResourceGroup",
      "resourceGuid": null,
      "securityRules": null,
      "subnets": null,
      "tags": null,
      "type": null
    },
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "myResourceGroup",
    "resourceGuid": "b3dbaa0e-2cf2-43be-a814-5cc49fea3304",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
```

```
  }
}
```

Create an availability set

Availability sets help spread your VMs across fault domains and update domains. Even though you only create one VM right now, it's best practice to use availability sets to make it easier to expand in the future.

Fault domains define a grouping of virtual machines that share a common power source and network switch. By default, the virtual machines that are configured within your availability set are separated across up to three fault domains. A hardware issue in one of these fault domains does not affect every VM that is running your app.

Update domains indicate groups of virtual machines and underlying physical hardware that can be rebooted at the same time. During planned maintenance, the order in which update domains are rebooted might not be sequential, but only one update domain is rebooted at a time.

Azure automatically distributes VMs across the fault and update domains when placing them in an availability set. For more information, see managing the availability of VMs.

Create an availability set for your VM with az vm availability-set create. The following example creates an availability set named *myAvailabilitySet*:

Azure CLI

```
az vm availability-set create \
    --resource-group myResourceGroup \
    --name myAvailabilitySet
```

The output notes fault domains and update domains:

JSON

```
{
  "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Comp
ute/availabilitySets/myAvailabilitySet",
  "location": "eastus",
  "managed": null,
  "name": "myAvailabilitySet",
  "platformFaultDomainCount": 2,
  "platformUpdateDomainCount": 5,
  "resourceGroup": "myResourceGroup",
  "sku": {
```

```
    "capacity": null,
    "managed": true,
    "tier": null
  },
  "statuses": null,
  "tags": {},
  "type": "Microsoft.Compute/availabilitySets",
  "virtualMachines": []
}
```

Create a VM

You've created the network resources to support Internet-accessible VMs. Now create a VM and secure it with an SSH key. In this example, let's create an Ubuntu VM based on the most recent LTS. You can find additional images with az vm image list, as described in finding Azure VM images.

Specify an SSH key to use for authentication. If you do not have an SSH public key pair, you can create them or use the `--generate-ssh-keys` parameter to create them for you. If you already have a key pair, this parameter uses existing keys in `~/.ssh`.

Create the VM by bringing all the resources and information together with the az vm create command. The following example creates a VM named *myVM*:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --location eastus \
    --availability-set myAvailabilitySet \
    --nics myNic \
    --image Ubuntu2204 \
    --admin-username azureuser \
    --generate-ssh-keys
```

SSH to your VM with the DNS entry you provided when you created the public IP address. This `fqdn` is shown in the output as you create your VM:

JSON

```
{
  "fqdns": "mypublicdns.eastus.cloudapp.azure.com",
  "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Comp
ute/virtualMachines/myVM",
  "location": "eastus",
```

```
  "macAddress": "00-0D-3A-13-71-C8",
  "powerState": "VM running",
  "privateIpAddress": "192.168.1.5",
  "publicIpAddress": "13.90.94.252",
  "resourceGroup": "myResourceGroup"
}
```

Bash

```
ssh azureuser@mypublicdns.eastus.cloudapp.azure.com
```

Output:

Bash

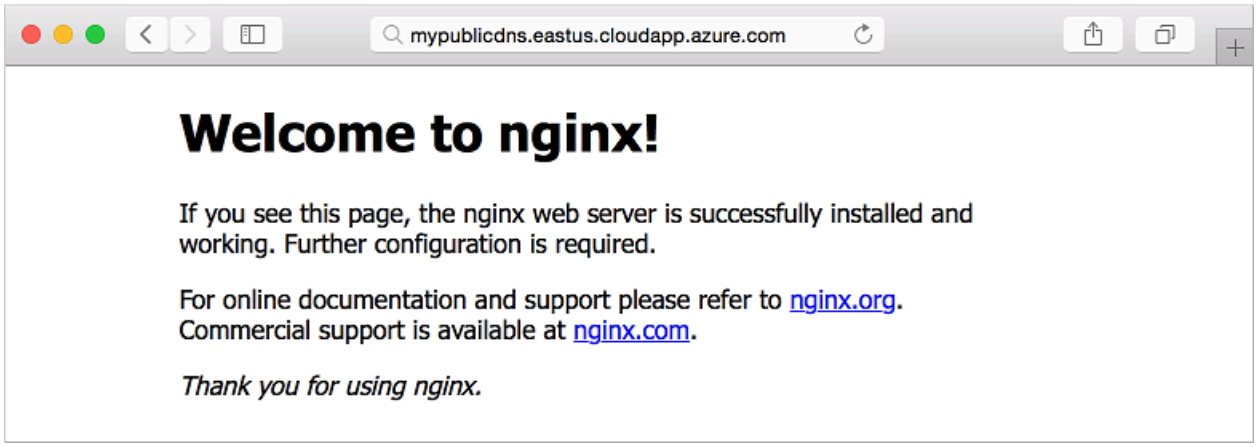
```
The authenticity of host 'mypublicdns.eastus.cloudapp.azure.com
(13.90.94.252)' can't be established.
ECDSA key fingerprint is SHA256:SylINP80Um6XRTvWiFaNz+H+1jcrKB1IiNgCDDJRj6A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added
'mypublicdns.eastus.cloudapp.azure.com,13.90.94.252' (ECDSA) to the list of
known hosts.
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.11.0-1016-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    https://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@myVM:~$
```

You can install NGINX and see the traffic flow to the VM. Install NGINX as follows:

Bash

```
sudo apt-get install -y nginx
```

To see the default NGINX site in action, open your web browser and enter your FQDN:



Export as a template

What if you now want to create an additional development environment with the same parameters, or a production environment that matches it? Resource Manager uses JSON templates that define all the parameters for your environment. You build out entire environments by referencing this JSON template. You can build JSON templates manually or export an existing environment to create the JSON template for you. Use az group export to export your resource group as follows:

Azure CLI

```
az group export --name myResourceGroup > myResourceGroup.json
```

This command creates the `myResourceGroup.json` file in your current working directory. When you create an environment from this template, you are prompted for all the resource names. You can populate these names in your template file by adding the `--include-parameter-default-value` parameter to the `az group export` command. Edit your JSON template to specify the resource names, or create a parameters.json file that specifies the resource names.

To create an environment from your template, use az deployment group create as follows:

Azure CLI

```
az deployment group create \
    --resource-group myNewResourceGroup \
```

```
    --template-file myResourceGroup.json
```

You might want to read more about how to deploy from templates. Learn about how to incrementally update environments, use the parameters file, and access templates from a single storage location.

Next steps

Now you're ready to begin working with multiple networking components and VMs. You can use this sample environment to build out your application by using the core components introduced here.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to create a Linux virtual machine with Azure Resource Manager templates

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Learn how to create a Linux virtual machine (VM) by using an Azure Resource Manager template and the Azure CLI from the Azure Cloud shell. To create a Windows virtual machine, see Create a Windows virtual machine from a Resource Manager template.

An alternative is to deploy the template from the Azure portal. To open the template in the portal, select the **Deploy to Azure** button.



Templates overview

Azure Resource Manager templates are JSON files that define the infrastructure and configuration of your Azure solution. By using a template, you can repeatedly deploy your solution throughout its lifecycle and have confidence your resources are deployed in a consistent state. To learn more about the format of the template and how you construct it, see Quickstart: Create and deploy Azure Resource Manager templates by using the Azure portal. To view the JSON syntax for resources types, see Define resources in Azure Resource Manager templates.

Quickstart template

> **Note**
>
> The provided template creates an Azure Generation 2 VM by default.

> **Note**
>
> Only SSH authentication is enabled by default when using the quickstart template. When prompted, provide the value of your own SSH public key, such as the contents of *~/.ssh/id_rsa.pub*.

If you don't have an SSH key pair, create and use an SSH key pair for Linux VMs in Azure.

Click **Copy** to add the quickstart template to your clipboard:

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "projectName": {
      "type": "string",
      "metadata": {
        "description": "Specifies a name for generating resource names."
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Specifies the location for all resources."
      }
    },
    "adminUsername": {
      "type": "string",
      "metadata": {
        "description": "Specifies a username for the Virtual Machine."
      }
    },
    "adminPublicKey": {
      "type": "string",
      "metadata": {
        "description": "Specifies the SSH rsa public key file as a string.
Use \"ssh-keygen -t rsa -b 2048\" to generate your SSH key pairs."
      }
    },
    "vmSize": {
      "type": "string",
      "defaultValue": "Standard_D2s_v3",
      "metadata": {
        "description": "description"
      }
    }
  },
  "variables": {
    "vNetName": "[concat(parameters('projectName'), '-vnet')]",
    "vNetAddressPrefixes": "10.0.0.0/16",
    "vNetSubnetName": "default",
    "vNetSubnetAddressPrefix": "10.0.0.0/24",
    "vmName": "[concat(parameters('projectName'), '-vm')]",
    "publicIPAddressName": "[concat(parameters('projectName'), '-ip')]",
```

```
    "networkInterfaceName": "[concat(parameters('projectName'), '-nic')]",
    "networkSecurityGroupName": "[concat(parameters('projectName'), '-
nsg')]",
    "networkSecurityGroupName2": "[concat(variables('vNetSubnetName'), '-
nsg')]"
  },
  "resources": [
    {
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2020-05-01",
      "name": "[variables('networkSecurityGroupName')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "ssh_rule",
            "properties": {
              "description": "Locks inbound down to ssh default port 22.",
              "protocol": "Tcp",
              "sourcePortRange": "*",
              "destinationPortRange": "22",
              "sourceAddressPrefix": "*",
              "destinationAddressPrefix": "*",
              "access": "Allow",
              "priority": 123,
              "direction": "Inbound"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/publicIPAddresses",
      "apiVersion": "2020-05-01",
      "name": "[variables('publicIPAddressName')]",
      "location": "[parameters('location')]",
      "properties": {
        "publicIPAllocationMethod": "Dynamic"
      },
      "sku": {
        "name": "Basic"
      }
    },
    {
      "comments": "Simple Network Security Group for subnet
[variables('vNetSubnetName')]",
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2020-05-01",
      "name": "[variables('networkSecurityGroupName2')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "default-allow-22",
            "properties": {
```

```
              "priority": 1000,
              "access": "Allow",
              "direction": "Inbound",
              "destinationPortRange": "22",
              "protocol": "Tcp",
              "sourceAddressPrefix": "*",
              "sourcePortRange": "*",
              "destinationAddressPrefix": "*"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2020-05-01",
      "name": "[variables('vNetName')]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName2'))]"
      ],
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('vNetAddressPrefixes')]"
          ]
        },
        "subnets": [
          {
            "name": "[variables('vNetSubnetName')]",
            "properties": {
              "addressPrefix": "[variables('vNetSubnetAddressPrefix')]",
              "networkSecurityGroup": {
                "id": "
[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName2'))]"
              }
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/networkInterfaces",
      "apiVersion": "2020-05-01",
      "name": "[variables('networkInterfaceName')]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName'))]",
        "[resourceId('Microsoft.Network/virtualNetworks',
variables('vNetName'))]",
        "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
```

```
      ],
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": {
                "id": "[resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName'))]"
              },
              "subnet": {
                "id": "
[resourceId('Microsoft.Network/virtualNetworks/subnets',
variables('vNetName'), variables('vNetSubnetName'))]"
              }
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2021-11-01",
      "name": "[variables('vmName')]",
      "location": "[parameters('location')]",
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
      ],
      "properties": {
        "hardwareProfile": {
          "vmSize": "[parameters('vmSize')]"
        },
        "osProfile": {
          "computerName": "[variables('vmName')]",
          "adminUsername": "[parameters('adminUsername')]",
          "linuxConfiguration": {
            "disablePasswordAuthentication": true,
            "ssh": {
              "publicKeys": [
                {
                  "path": "[concat('/home/', parameters('adminUsername'),
'/.ssh/authorized_keys')]",
                  "keyData": "[parameters('adminPublicKey')]"
                }
              ]
            }
          }
        },
        "storageProfile": {
          "imageReference": {
            "publisher": "Canonical",
            "offer": "0001-com-ubuntu-server-jammy",
            "sku": "22_04-lts-gen2",
```

```
            "version": "latest"
          },
          "osDisk": {
            "createOption": "fromImage"
          }
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
            }
          ]
        }
      }
    }
  ]
}
```

You can also download or create a template and specify the local path with the `--template-file` parameter.

Create a quickstart template VM with Azure CLI

After acquiring or creating a quickstart template, create a VM with it using the Azure CLI.

The following command requests several pieces of input from the user. These include:

- Name of the Resource Group (resourceGroupName)
- Location of the Azure datacenter that hosts the VM (location)
- A name for resources related to the VM (projectName)
- Username for the administrator user (username)
- A public SSH key for accessing the VM's terminal (key)

Creating an Azure virtual machine requires a resource group. Quickstart templates include resource group creation as part of the process.

To run the CLI script, click **Open Cloudshell**. Once you have access to the Azure Cloudshell, click **Copy** to copy the command, right-click the shell, then select **Paste**.

Azure CLI

```
echo "Enter the Resource Group name:" &&
read resourceGroupName &&
echo "Enter the location (i.e. centralus):" &&
read location &&
echo "Enter the project name (used for generating resource names):" &&
read projectName &&
```

```
echo "Enter the administrator username:" &&
read username &&
echo "Enter the SSH public key:" &&
read key &&
az group create --name $resourceGroupName --location "$location" &&
az deployment group create --resource-group $resourceGroupName --template-
uri https://raw.githubusercontent.com/azure/azure-quickstart-
templates/master/quickstarts/microsoft.compute/vm-sshkey/azuredeploy.json --
parameters projectName=$projectName adminUsername=$username
adminPublicKey="$key" &&
az vm show --resource-group $resourceGroupName --name "$projectName-vm" --
show-details --query publicIps --output tsv
```

The last line in the command shows the public IP address of the newly created VM. You need the public IP address to connect to the virtual machine.

Connect to virtual machine

You can then SSH to your VM as normal. Provide your own public IP address from the preceding command:

Bash

```
ssh <adminUsername>@<ipAddress>
```

Other templates

In this example, you created a basic Linux VM. For more Resource Manager templates that include application frameworks or create more complex environments, browse the Azure Quickstart Templates.

To learn more about creating templates, view the JSON syntax and properties for the resources types you deployed:

- Microsoft.Network/networkSecurityGroups
- Microsoft.Network/publicIPAddresses
- Microsoft.Network/virtualNetworks
- Microsoft.Network/networkInterfaces
- Microsoft.Compute/virtualMachines

Next steps

- To learn how to develop Resource Manager templates, see Azure Resource Manager documentation.
- To see the Azure virtual machine schemas, see Azure template reference.
- To see more virtual machine template samples, see Azure Quickstart templates.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Virtual Machines - Create Or Update

Service: Compute
API Version: 2025-11-01

The operation to create or update a virtual machine. Please note some properties can be set only during virtual machine creation.

HTTP

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmName}?api-version=2025-11-01
```

URI Parameters

Expand table

Name	In	Required	Type	Description
`resourceGroupName`	path	True	string minLength: 1 maxLength: 90	The name of the resource group. The name is case insensitive.
`subscriptionId`	path	True	string minLength: 1	The ID of the target subscription.
`vmName`	path	True	string	The name of the virtual machine.
`api-version`	query	True	string minLength: 1	The API version to use for this operation.

Request Header

Expand table

Name	Required	Type	Description
If-Match		string	The ETag of the transformation. Omit this value to always overwrite the current resource. Specify the last-seen ETag value to prevent accidentally overwriting concurrent changes.
If-None-Match		string	Set to '*' to allow a new record set to be created, but to prevent updating an existing record set. Other values will result in error from server as they are not supported.

Request Body

Expand table

Name	Required	Type	Description
location	True	string	The geo-location where the resource lives
extendedLocation		Extended Location	The extended location of the Virtual Machine.
identity		Virtual Machine Identity	The identity of the virtual machine, if configured.
placement		Placement	Placement section specifies the user-defined constraints for virtual machine hardware placement. This property cannot be changed once VM is provisioned. Minimum api-version: 2024-11-01.

plan	Plan	Specifies information about the marketplace image used to create the virtual machine. This element is only used for marketplace images. Before you can use a marketplace image from an API, you must enable the image for programmatic use. In the Azure portal, find the marketplace image that you want to use and then click **Want to deploy programmatically, Get Started ->**. Enter any required information and then click **Save**.
properties.additionalCapabilities	Additional Capabilities	Specifies additional capabilities enabled or disabled on the virtual machine.
properties.applicationProfile	Application Profile	Specifies the gallery applications that should be made available to the VM/VMSS.
properties.availabilitySet	Sub Resource	Specifies information about the availability set that the virtual machine should be assigned to. Virtual machines specified in the same availability set are allocated to different nodes to maximize availability. For more information about availability sets, see Availability sets overview. For more information on Azure planned maintenance, see Maintenance and updates for Virtual Machines in Azure. Currently, a VM can only be added to availability set at creation time. The availability set to which the VM is being added should be under the same resource group as the availability set resource. An existing VM cannot be added to an availability set. This property cannot exist along with a non-null properties.virtualMachineScaleSet reference.
properties.billingProfile	Billing Profile	Specifies the billing related details of a Azure Spot virtual machine. Minimum api-version: 2019-03-01.
properties.capacityReservation	Capacity Reservation Profile	Specifies information about the capacity reservation that is used to allocate virtual machine. Minimum api-version: 2021-04-01.
properties.diagnosticsProfile	Diagnostics Profile	Specifies the boot diagnostic settings state. Minimum api-version: 2015-06-15.
properties.evictionPolicy	Virtual Machine Eviction PolicyTypes	Specifies the eviction policy for the Azure Spot virtual machine and Azure Spot scale set. For Azure Spot virtual machines, both 'Deallocate' and 'Delete' are supported and the minimum api-version is 2019-03-01. For Azure Spot scale sets, both 'Deallocate' and 'Delete' are supported and the minimum api-version is 2017-10-30-preview.
properties.extensionsTimeBudget	string	Specifies the time alloted for all extensions to start. The time duration should be between 15 minutes and 120 minutes (inclusive) and should be specified in ISO 8601 format. The default value is 90 minutes (PT1H30M). Minimum api-version: 2020-06-01.
properties.hardwareProfile	Hardware Profile	Specifies the hardware settings for the virtual machine.
properties.host	Sub Resource	Specifies information about the dedicated host that the virtual machine resides in. Minimum api-version: 2018-10-01.
properties.hostGroup	Sub Resource	Specifies information about the dedicated host group that the virtual machine resides in. **Note:** User cannot specify both host and hostGroup properties. Minimum api-version: 2020-06-01.

properties.licenseType	string	Specifies that the image or disk that is being used was licensed on-premises. Possible values for Windows Server operating system are: Windows_Client Windows_Server Possible values for Linux Server operating system are: RHEL_BYOS (for RHEL) SLES_BYOS (for SUSE) For more information, see Azure Hybrid Use Benefit for Windows Server Azure Hybrid Use Benefit for Linux Server Minimum api-version: 2015-06-15
properties.networkProfile	Network Profile	Specifies the network interfaces of the virtual machine.
properties.osProfile	OSProfile	Specifies the operating system settings used while creating the virtual machine. Some of the settings cannot be changed once VM is provisioned.
properties.platformFaultDomain	integer (int32)	Specifies the scale set logical fault domain into which the Virtual Machine will be created. By default, the Virtual Machine will by automatically assigned to a fault domain that best maintains balance across available fault domains. This is applicable only if the 'virtualMachineScaleSet' property of this Virtual Machine is set. The Virtual Machine Scale Set that is referenced, must have 'platformFaultDomainCount' greater than 1. This property cannot be updated once the Virtual Machine is created. Fault domain assignment can be viewed in the Virtual Machine Instance View. Minimum api-version: 2020-12-01.
properties.priority	Virtual Machine Priority Types	Specifies the priority for the virtual machine. Minimum api-version: 2019-03-01
properties.proximityPlacementGroup	Sub Resource	Specifies information about the proximity placement group that the virtual machine should be assigned to. Minimum api-version: 2018-04-01.
properties.resiliencyProfile	Resiliency Profile	Resiliency profile for the virtual machine.
properties.scheduledEventsPolicy	Scheduled Events Policy	Specifies Redeploy, Reboot and ScheduledEventsAdditionalPublishingTargets Scheduled Event related configurations for the virtual machine.
properties.scheduledEventsProfile	Scheduled Events Profile	Specifies Scheduled Event related configurations.
properties.securityProfile	Security Profile	Specifies the Security related profile settings for the virtual machine.
properties.storageProfile	Storage Profile	Specifies the storage settings for the virtual machine disks.

properties.userData	string	UserData for the VM, which must be base-64 encoded. Customer should not pass any secrets in here. Minimum api-version: 2021-03-01.
properties.virtualMachineScaleSet	Sub Resource	Specifies information about the virtual machine scale set that the virtual machine should be assigned to. Virtual machines specified in the same virtual machine scale set are allocated to different nodes to maximize availability. Currently, a VM can only be added to virtual machine scale set at creation time. An existing VM cannot be added to a virtual machine scale set. This property cannot exist along with a non-null properties.availabilitySet reference. Minimum api-version: 2019-03-01.
tags	object	Resource tags.
zones	string[]	The availability zones.

Responses

Expand table

Name	Type	Description
200 OK	Virtual Machine	Resource 'VirtualMachine' update operation succeeded
201 Created	Virtual Machine	Resource 'VirtualMachine' create operation succeeded Headers • Location: string • Retry-After: integer
Other Status Codes	CloudError	An unexpected error response.

Security

azure_auth

Azure Active Directory OAuth2 Flow.

Type: oauth2
Flow: implicit
Authorization URL: https://login.microsoftonline.com/common/oauth2/authorize

Scopes

Expand table

Name	Description
user_impersonation	impersonate your user account

Examples

Expand table

Create a custom-image vm from an unmanaged generalized os image.
Create a Linux vm with a patch setting assessmentMode of ImageDefault.

Create a Linux vm with a patch setting patchMode of AutomaticByPlatform and AutomaticByPlatform Settings.
Create a Linux vm with a patch setting patchMode of ImageDefault.
Create a Linux vm with a patch settings patchMode and assessmentMode set to AutomaticByPlatform.
Create a platform-image vm with unmanaged os and data disks.
Create a VM from a community gallery image
Create a vm from a custom image.
Create a vm from a generalized shared image.
Create a VM from a shared gallery image
Create a vm from a specialized shared image.
Create a vm in a Virtual Machine Scale Set with customer assigned platformFaultDomain.
Create a vm in an availability set.
Create a vm with a marketplace image plan.
Create a vm with an extensions time budget.
Create a vm with Application Profile.
Create a VM with automatic zone placement
Create a vm with boot diagnostics.
Create a vm with data disks using 'Copy' and 'Restore' options.
Create a VM with Disk Controller Type
Create a vm with DiskEncryptionSet resource id in the os disk and data disk.
Create a vm with empty data disks.
Create a VM with encryption identity
Create a vm with ephemeral os disk and enableFullCaching set to true.
Create a vm with ephemeral os disk provisioning in Cache disk using placement property.
Create a vm with ephemeral os disk provisioning in Nvme disk using placement property.
Create a vm with ephemeral os disk provisioning in Resource disk using placement property.
Create a vm with ephemeral os disk.
Create a VM with FIPS 140-3 Enabled
Create a VM with HibernationEnabled
Create a vm with Host Encryption using encryptionAtHost property.
Create a vm with managed boot diagnostics.
Create a VM with network interface configuration
Create a VM with network interface configuration with public ip address dns settings
Create a vm with password authentication.
Create a vm with premium storage.
Create a VM with ProxyAgent Settings of enabled and mode.
Create a vm with resiliency profile and zone movement enabled.
Create a vm with Scheduled Events Profile
Create a VM with securityType ConfidentialVM with Customer Managed Keys
Create a VM with securityType ConfidentialVM with NonPersistedTPM securityEncryptionType
Create a VM with securityType ConfidentialVM with Platform Managed Keys

Create a vm with ssh authentication.
Create a VM with Uefi Settings of secureBoot and vTPM.
Create a VM with UserData
Create a VM with VM Size Properties
Create a Windows vm with a patch setting assessmentMode of ImageDefault.
Create a Windows vm with a patch setting patchMode of AutomaticByOS.
Create a Windows vm with a patch setting patchMode of AutomaticByPlatform and AutomaticByPlatform Settings.
Create a Windows vm with a patch setting patchMode of AutomaticByPlatform and enableHotpatching set to true.
Create a Windows vm with a patch setting patchMode of Manual.
Create a Windows vm with patch settings patchMode and assessmentMode set to AutomaticByPlatform.
Create or update a VM with capacity reservation

Create a custom-image vm from an unmanaged generalized os image.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-name}?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "osDisk": {
        "name": "myVMosdisk",
        "image": {
          "uri": "http://{existing-storage-account-name}.blob.core.windows.net/{existing-container-name}/{existing-generalized-os-image-blob-name}.vhd"
        },
        "osType": "Windows",
        "createOption": "FromImage",
        "caching": "ReadWrite",
        "vhd": {
          "uri": "http://{existing-storage-account-name}.blob.core.windows.net/{existing-container-name}/myDisk.vhd"
        }
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
```

```
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "osDisk": {
        "name": "myVMosdisk",
        "image": {
          "uri": "https://{existing-storage-account-
name}.blob.core.windows.net/system/Microsoft.Compute/Images/vhds/{existing-
generalized-os-image-blob-name}.vhd"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Windows",
        "vhd": {
          "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk.vhd"
        }
      },
      "dataDisks": []
    },
    "vmId": "926cd555-a07c-4ff5-b214-4aa4dd09d79b",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
```

```
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "osDisk": {
        "name": "myVMosdisk",
        "image": {
          "uri": "https://{existing-storage-account-
name}.blob.core.windows.net/system/Microsoft.Compute/Images/vhds/{existing-
generalized-os-image-blob-name}.vhd"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Windows",
        "vhd": {
          "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk.vhd"
        }
      },
      "dataDisks": []
    },
    "vmId": "926cd555-a07c-4ff5-b214-4aa4dd09d79b",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a Linux vm with a patch setting assessment Mode of ImageDefault.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
```

```
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "assessmentMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "assessmentMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
```

```
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "assessmentMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
```

```
  "name": "myVM",
  "location": "westus"
}
```

Create a Linux vm with a patch setting patch Mode of AutomaticByPlatform and Automatic ByPlatformSettings.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform",
          "automaticByPlatformSettings": {
            "rebootSetting": "Never",
            "bypassPlatformSafetyChecksOnUserSchedule": true
          }
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform",
          "automaticByPlatformSettings": {
            "rebootSetting": "Never",
            "bypassPlatformSafetyChecksOnUserSchedule": true
          }
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
```

```
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform",
          "automaticByPlatformSettings": {
            "rebootSetting": "Never",
            "bypassPlatformSafetyChecksOnUserSchedule": true
          }
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a Linux vm with a patch setting patchMode of ImageDefault.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "storageProfile": {
```

```
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
```

```
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
```

```
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a Linux vm with a patch settings patchMode and assessmentMode set to AutomaticByPlatform.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform"
```

```
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a platform-image vm with unmanaged os and data disks.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{v
m-name}?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "vhd": {
          "uri": "http://{existing-storage-account-
```

```
name}.blob.core.windows.net/{existing-container-name}/myDisk.vhd"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": [
        {
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 0,
          "vhd": {
            "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/{existing-container-name}/myDisk0.vhd"
          }
        },
        {
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 1,
          "vhd": {
            "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/{existing-container-name}/myDisk1.vhd"
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
```

```
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "vhd": {
          "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk.vhd"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "caching": "ReadWrite"
      },
      "dataDisks": [
        {
          "name": "dataDisk0",
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "caching": "None",
          "vhd": {
            "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk0.vhd"
          },
          "lun": 0
        },
        {
          "name": "dataDisk1",
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "caching": "None",
          "vhd": {
            "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk1.vhd"
          },
          "lun": 1
        }
      ]
    },
    "vmId": "5230a749-2f68-4830-900b-702182d32e63",
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
```

```
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "vhd": {
          "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk.vhd"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "caching": "ReadWrite"
      },
      "dataDisks": [
        {
          "name": "dataDisk0",
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "caching": "None",
          "vhd": {
            "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk0.vhd"
          },
          "lun": 0
        },
        {
          "name": "dataDisk1",
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "caching": "None",
          "vhd": {
            "uri": "http://{existing-storage-account-
name}.blob.core.windows.net/vhds/myDisk1.vhd"
          },
          "lun": 1
        }
      ]
    },
    "vmId": "5230a749-2f68-4830-900b-702182d32e63",
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a VM from a community gallery image

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
```

```
      "imageReference": {
        "communityGalleryImageId":
"/CommunityGalleries/galleryPublicName/Images/communityGalleryImageName/Versions/
communityGalleryImageVersionName"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "communityGalleryImageId":
"/CommunityGalleries/galleryPublicName/Images/communityGalleryImageName/Versions/commu
nityGalleryImageVersionName"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
```

```
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "communityGalleryImageId":
"/CommunityGalleries/galleryPublicName/Images/communityGalleryImageName/Versions/commu
nityGalleryImageVersionName"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm from a custom image.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/{existing-custom-image-name}"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgExistingNic",
```

```
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/nsgcustom"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/nsgcustom"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
```

```
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a vm from a generalized shared image.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/galleries/myShared
Gallery/images/mySharedImage"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/galleries/mySharedGalle
ry/images/mySharedImage"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
```

```
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/galleries/mySharedGalle
ry/images/mySharedImage"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a VM from a shared gallery image

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sharedGalleryImageId":
"/SharedGalleries/sharedGalleryName/Images/sharedGalleryImageName/Versions/shared
GalleryImageVersionName"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
```

```
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sharedGalleryImageId":
"/SharedGalleries/sharedGalleryName/Images/sharedGalleryImageName/Versions/sharedGalle
ryImageVersionName"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sharedGalleryImageId":
"/SharedGalleries/sharedGalleryName/Images/sharedGalleryImageName/Versions/sharedGalle
ryImageVersionName"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm from a specialized shared image.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
```

```
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/galleries/myShared
Gallery/images/mySharedImage"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/galleries/mySharedGalle
ry/images/mySharedImage"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
```

```
  "location": "westus"
}
```

Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/galleries/mySharedGalle
ry/images/mySharedImage"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": []
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm in a Virtual Machine Scale Set with customer assigned platformFaultDomain.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
```

```
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "virtualMachineScaleSet": {
      "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachineScal
eSets/{existing-flex-vmss-name-with-platformFaultDomainCount-greater-than-1}"
    },
    "platformFaultDomain": 1
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
```

```
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "7cce54f2-ecd3-4ddd-a8d9-50984faa3918",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "virtualMachineScaleSet": {
      "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachineScaleSets
/myExistingFlexVmss"
    },
    "platformFaultDomain": 1,
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "7cce54f2-ecd3-4ddd-a8d9-50984faa3918",
    "hardwareProfile": {
```

```
      "vmSize": "Standard_D1_v2"
    },
    "virtualMachineScaleSet": {
      "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachineScaleSets
/myExistingFlexVmss"
    },
    "platformFaultDomain": 1,
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a vm in an availability set.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "availabilitySet": {
      "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/availabilitySets/{
existing-availability-set-name}"
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b7a098cc-b0b8-46e8-a205-62f301a62a8f",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "availabilitySet": {
      "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/availabilitySets/NSGEXI
STINGAS"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
```

```
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b7a098cc-b0b8-46e8-a205-62f301a62a8f",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "availabilitySet": {
      "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/availabilitySets/NSGEXI
STINGAS"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm with a marketplace image plan.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
```

```
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
```

```
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
```

```
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Create a vm with an extensions time budget.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
    },
    "extensionsTimeBudget": "PT30M"
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "extensionsTimeBudget": "PT30M",
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
```

```
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "extensionsTimeBudget": "PT30M",
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a vm with Application Profile.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "{image_sku}",
        "publisher": "{image_publisher}",
        "version": "latest",
```

```
        "offer": "{image_offer}"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "applicationProfile": {
      "galleryApplications": [
        {
          "tags": "myTag1",
          "order": 1,
          "packageReferenceId": "/subscriptions/32c17a9e-aa7b-4ba5-a45b-
e324116b6fdb/resourceGroups/myresourceGroupName2/providers/Microsoft.Compute/gall
eries/myGallery1/applications/MyApplication1/versions/1.0",
          "configurationReference":
"https://mystorageaccount.blob.core.windows.net/configurations/settings.config",
          "treatFailureAsDeploymentFailure": false,
          "enableAutomaticUpgrade": false
        },
        {
          "packageReferenceId": "/subscriptions/32c17a9e-aa7b-4ba5-a45b-
e324116b6fdg/resourceGroups/myresourceGroupName3/providers/Microsoft.Compute/gall
eries/myGallery2/applications/MyApplication2/versions/1.1"
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "ssh": {
          "publicKeys": [
            {
              "path": "/home/{your-username}/.ssh/authorized_keys",
              "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCeClRAk2ipUs/l5voIsDC5q9RI+YSRd1Bvd/O+axgY4WiBzG+4FwJWZm
/mLLe5DoOdHQwmU2FrKXZSW4w2sYE70KeWnrFViCOX5MTVvJgPE8ClugNl8RWth/tU849DvM9sT7vFgfVSHcAS
2yDRyDlueii+8nF2ym8XWAPltFVCyLHRsyBp5YPqK8JFYIa1eybKsY3hEAxRCA+/7bq8et+Gj3coOsuRmrehav
7rE6N12Pb80I6ofa6SM5XNYq4Xk0iYNx7R3kdz0Jj9XgZYWjAHjJmT0gTRoOnt6upOuxK7xI/ykWrllgpXrCPu
3Ymz+c+ujaqcxDopnAl2lmf69/J1"
            }
          ]
```

```
        },
        "disablePasswordAuthentication": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "applicationProfile": {
      "galleryApplications": [
        {
          "tags": "myTag1",
          "order": 1,
          "packageReferenceId": "/subscriptions/32c17a9e-aa7b-4ba5-a45b-
e324116b6fdb/resourceGroups/myresourceGroupName2/providers/Microsoft.Compute/galleries
/myGallery1/applications/MyApplication1/versions/1.0",
          "configurationReference":
"https://mystorageaccount.blob.core.windows.net/configurations/settings.config"
        },
        {
          "packageReferenceId": "/subscriptions/32c17a9e-aa7b-4ba5-a45b-
e324116b6fdg/resourceGroups/myresourceGroupName3/providers/Microsoft.Compute/galleries
/myGallery2/applications/MyApplication2/versions/1.1"
        }
      ]
    },
    "vmId": "e0de9b84-a506-4b95-9623-00a425d05c90",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "ssh": {
          "publicKeys": [
```

```
            {
              "path": "/home/{your-username}/.ssh/authorized_keys",
              "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCeClRAk2ipUs/l5voIsDC5q9RI+YSRd1Bvd/O+axgY4WiBzG+4FwJWZm
/mLLe5DoOdHQwmU2FrKXZSW4w2sYE70KeWnrFViCOX5MTVvJgPE8ClugNl8RWth/tU849DvM9sT7vFgfVSHcAS
2yDRyDlueii+8nF2ym8XWAPltFVCyLHRsyBp5YPqK8JFYIa1eybKsY3hEAxRCA+/7bq8et+Gj3coOsuRmrehav
7rE6N12Pb80I6ofa6SM5XNYq4Xk0iYNx7R3kdz0Jj9XgZYWjAHjJmT0gTRoOnt6upOuxK7xI/ykWrllgpXrCPu
3Ymz+c+ujaqcxDopnAl2lmf69/J1"
            }
          ]
        },
        "disablePasswordAuthentication": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "applicationProfile": {
      "galleryApplications": [
        {
          "tags": "myTag1",
          "order": 1,
          "packageReferenceId": "/subscriptions/32c17a9e-aa7b-4ba5-a45b-
e324116b6fdb/resourceGroups/myresourceGroupName2/providers/Microsoft.Compute/galleries
/myGallery1/applications/MyApplication1/versions/1.0",
          "configurationReference":
"https://mystorageaccount.blob.core.windows.net/configurations/settings.config"
        },
        {
          "packageReferenceId": "/subscriptions/32c17a9e-aa7b-4ba5-a45b-
e324116b6fdg/resourceGroups/myresourceGroupName3/providers/Microsoft.Compute/galleries
/myGallery2/applications/MyApplication2/versions/1.1"
        }
      ]
    },
    "vmId": "e0de9b84-a506-4b95-9623-00a425d05c90",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a VM with automatic zone placement

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus2",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  },
  "placement": {
    "zonePlacementPolicy": "Any",
    "includeZones": [
      "1",
      "3"
    ]
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
```

```
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "zones": [
    "3"
  ],
  "placement": {
    "zonePlacementPolicy": "Any",
    "includeZones": [
      "1",
      "3"
    ]
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus2"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
```

```
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "zones": [
    "3"
  ],
  "placement": {
    "zonePlacementPolicy": "Any",
    "includeZones": [
      "1",
      "3"
    ]
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus2"
}
```


Create a vm with boot diagnostics.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
```

```
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
```

```
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
```

```
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm with data disks using 'Copy' and 'Restore' options.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      },
      "dataDisks": [
        {
          "diskSizeGB": 1023,
          "createOption": "Copy",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/snapshots/{existin
g-snapshot-name}"
          },
          "lun": 0
        },
        {
          "diskSizeGB": 1023,
          "createOption": "Copy",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/{existing-
disk-name}"
          },
          "lun": 1
        },
        {
          "diskSizeGB": 1023,
          "createOption": "Restore",
```

```
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/restorePointCollec
tions/{existing-rpc-name}/restorePoints/{existing-rp-
name}/diskRestorePoints/{existing-disk-restore-point-name}"
          },
          "lun": 2
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": [
```

```
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Copy",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/snapshots/{existing-
snapshot-name}"
          },
          "lun": 0,
          "diskSizeGB": 1023
        },
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Copy",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/{existing-disk-
name}"
          },
          "lun": 1,
          "diskSizeGB": 1023
        },
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Restore",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/restorePointCollections
/{existing-rpc-name}/restorePoints/{existing-rp-name}/diskRestorePoints/{existing-
disk-restore-point-name}"
          },
          "lun": 2,
          "diskSizeGB": 1023
        }
      ]
    },
    "vmId": "3906fef9-a1e5-4b83-a8a8-540858b41df0",
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
```

```
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": [
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Copy",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/snapshots/{existing-
snapshot-name}"
          },
          "lun": 0,
          "diskSizeGB": 1023,
          "toBeDetached": false
        },
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Copy",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/{existing-disk-
name}"
          },
          "lun": 1,
          "diskSizeGB": 1023,
          "toBeDetached": false
        },
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Restore",
          "sourceResource": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/restorePointCollections
/{existing-rpc-name}/restorePoints/{existing-rp-name}/diskRestorePoints/{existing-
disk-restore-point-name}"
          },
          "lun": 2,
          "diskSizeGB": 1023,
          "toBeDetached": false
        }
      ]
    },
    "vmId": "3906fef9-a1e5-4b83-a8a8-540858b41df0",
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
```

```
  "location": "westus"
}
```

Create a VM with Disk Controller Type

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D4_v3"
    },
    "scheduledEventsPolicy": {
      "scheduledEventsAdditionalPublishingTargets": {
        "eventGridAndResourceGraph": {
          "enable": true,
          "scheduledEventsApiVersion": "2020-07-01"
        }
      },
      "userInitiatedRedeploy": {
        "automaticallyApprove": true
      },
      "userInitiatedReboot": {
        "automaticallyApprove": true
      },
      "allInstancesDown": {
        "automaticallyApprove": true
      }
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      },
      "diskControllerType": "NVMe"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
```

```
      }
    },
    "userData": "U29tZSBDdXN0b20gRGF0YQ=="
  }
}
```


Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": [],
      "diskControllerType": "NVMe"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D4_v3"
    },
    "scheduledEventsPolicy": {
      "scheduledEventsAdditionalPublishingTargets": {
        "eventGridAndResourceGraph": {
          "enable": true,
          "scheduledEventsApiVersion": "2020-07-01"
        }
      },
      "userInitiatedRedeploy": {
        "automaticallyApprove": true
      },
      "userInitiatedReboot": {
        "automaticallyApprove": true
```

```
      },
      "allInstancesDown": {
        "automaticallyApprove": true
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Updating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": [],
      "diskControllerType": "NVMe"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D4_v3"
    },
    "scheduledEventsPolicy": {
      "scheduledEventsAdditionalPublishingTargets": {
        "eventGridAndResourceGraph": {
          "enable": true,
          "scheduledEventsApiVersion": "2020-07-01"
        }
      },
      "userInitiatedRedeploy": {
```

```
        "automaticallyApprove": true
      },
      "userInitiatedReboot": {
        "automaticallyApprove": true
      },
      "allInstancesDown": {
        "automaticallyApprove": true
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a vm with DiskEncryptionSet resource id in the os disk and data disk.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/{existing-
custom-image-name}"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS",
          "diskEncryptionSet": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets
/{existing-diskEncryptionSet-name}"
          }
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      },
      "dataDisks": [
        {
          "caching": "ReadWrite",
          "managedDisk": {
            "storageAccountType": "Standard_LRS",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets
/{existing-diskEncryptionSet-name}"
            }
          },
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 0
        },
        {
          "caching": "ReadWrite",
          "managedDisk": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/{existing-
```

```
managed-disk-name}",
            "storageAccountType": "Standard_LRS",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets
/{existing-diskEncryptionSet-name}"
            }
          },
          "diskSizeGB": 1023,
          "createOption": "Attach",
          "lun": 1
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/nsgcustom"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS",
          "diskEncryptionSet": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskencryptionset-name}"
```

```
          }
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": [
        {
          "caching": "ReadWrite",
          "managedDisk": {
            "storageAccountType": "Standard_LRS",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskEncryptionSet-name}"
            }
          },
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 0
        },
        {
          "caching": "ReadWrite",
          "managedDisk": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/{existing-
managed-disk-name}",
            "storageAccountType": "Standard_LRS",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskEncryptionSet-name}"
            }
          },
          "diskSizeGB": 1023,
          "createOption": "Attach",
          "lun": 1
        }
      ]
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
```

```
    },
    "storageProfile": {
      "imageReference": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/images/nsgcustom"
      },
      "osDisk": {
        "name": "myVMosdisk",
        "diskSizeGB": 30,
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "osType": "Linux"
      },
      "dataDisks": [
        {
          "caching": "ReadWrite",
          "managedDisk": {
            "storageAccountType": "Standard_LRS",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskEncryptionSet-name}"
            }
          },
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 0
        },
        {
          "caching": "ReadWrite",
          "managedDisk": {
            "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/{existing-
managed-disk-name}",
            "storageAccountType": "Standard_LRS",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskEncryptionSet-name}"
            }
          },
          "diskSizeGB": 1023,
          "createOption": "Attach",
          "lun": 1
        }
      ]
    },
    "vmId": "71aa3d5a-d73d-4970-9182-8580433b2865",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm with empty data disks.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
```

```
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      },
      "dataDisks": [
        {
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 0
        },
        {
          "diskSizeGB": 1023,
          "createOption": "Empty",
          "lun": 1
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
```

```
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": [
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Empty",
          "lun": 0,
          "diskSizeGB": 1023
        },
        {
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Standard_LRS"
          },
          "createOption": "Empty",
          "lun": 1,
          "diskSizeGB": 1023
        }
      ]
    },
    "vmId": "3906fef9-a1e5-4b83-a8a8-540858b41df0",
    "hardwareProfile": {
      "vmSize": "Standard_D2_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
```

```
        }
      }
    ]
  },
  "storageProfile": {
    "imageReference": {
      "sku": "2016-Datacenter",
      "publisher": "MicrosoftWindowsServer",
      "version": "latest",
      "offer": "WindowsServer"
    },
    "osDisk": {
      "osType": "Windows",
      "caching": "ReadWrite",
      "createOption": "FromImage",
      "name": "myVMosdisk",
      "managedDisk": {
        "storageAccountType": "Standard_LRS"
      }
    },
    "dataDisks": [
      {
        "caching": "None",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "Empty",
        "lun": 0,
        "diskSizeGB": 1023,
        "toBeDetached": false
      },
      {
        "caching": "None",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "Empty",
        "lun": 1,
        "diskSizeGB": 1023,
        "toBeDetached": false
      }
    ]
  },
  "vmId": "3906fef9-a1e5-4b83-a8a8-540858b41df0",
  "hardwareProfile": {
    "vmSize": "Standard_D2_v2"
  },
  "securityProfile": {
    "securityType": "Standard"
  },
  "provisioningState": "Creating"
},
"name": "myVM",
"location": "westus"
}
```


Create a VM with encryption identity

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "identity": {
    "type": "UserAssigned",
    "userAssignedIdentities": {
      "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.ManagedIdentity/userAssign
edIdentities/myIdentity": {}
    }
```

```
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "encryptionIdentity": {
        "userAssignedIdentityResourceId": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.ManagedIdentity/userAssign
edIdentities/myIdentity"
      }
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "identity": {
    "type": "UserAssigned",
    "userAssignedIdentities": {
      "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.ManagedIdentity/userAssignedIde
ntities/myIdentity": {}
    }
  },
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
```

```
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "encryptionIdentity": {
        "userAssignedIdentityResourceId": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.ManagedIdentity/userAssignedIde
ntities/myIdentity"
      },
      "securityType": "Standard"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "identity": {
    "type": "UserAssigned",
    "userAssignedIdentities": {
      "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.ManagedIdentity/userAssignedIde
ntities/myIdentity": {}
    }
  },
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
```

```
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "encryptionIdentity": {
        "userAssignedIdentityResourceId": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.ManagedIdentity/userAssignedIde
ntities/myIdentity"
      },
      "securityType": "Standard"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a vm with ephemeral os disk and enableFull Caching set to true.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "TempDisk",
          "enableFullCaching": true
        },
```

```
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
```

```
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
```

```
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Create a vm with ephemeral os disk provisioning in Cache disk using placement property.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "CacheDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "CacheDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
```

```
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "CacheDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a vm with ephemeral os disk provisioning in Nvme disk using placement property.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
```

```
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "NvmeDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
```

```
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "NvmeDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "NvmeDisk"
        },
```

```
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a vm with ephemeral os disk provisioning in Resource disk using placement property.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "ResourceDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
```

```
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "ResourceDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
```

```
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local",
          "placement": "ResourceDisk"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Create a vm with ephemeral os disk.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
```

```
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
```

```
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "diffDiskSettings": {
          "option": "Local"
        },
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a VM with FIPS 140-3 Enabled

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{v
m-name}?api-version=2025-11-01

{
  "location": "eastus2euap",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "additionalCapabilities": {
      "enableFips1403Encryption": true
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
```

```
          "storageAccountType": "Standard_LRS"
        },
        "name": "vmOSdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "{vm-name}",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-
name}",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "{vm-name}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "vmOSdisk",
        "managedDisk": {
```

```
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "additionalCapabilities": {
      "enableFips1403Encryption": true
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Updating"
  },
  "name": "{vm-name}",
  "location": "eastus2euap"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-
name}",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "{vm-name}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "vmOSdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
```

```
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "additionalCapabilities": {
      "enableFips1403Encryption": true
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "{vm-name}",
  "location": "eastus2euap"
}
```

Create a VM with HibernationEnabled

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{v
m-name}?api-version=2025-11-01

{
  "location": "eastus2euap",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "additionalCapabilities": {
      "hibernationEnabled": true
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "vmOSdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "{vm-name}",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
```

```
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-name}",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "{vm-name}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgExistingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "vmOSdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "additionalCapabilities": {
      "hibernationEnabled": true
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Updating"
  },
  "name": "{vm-name}",
  "location": "eastus2euap"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-
name}",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "{vm-name}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "vmOSdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "additionalCapabilities": {
      "hibernationEnabled": true
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "{vm-name}",
  "location": "eastus2euap"
}
```


Create a vm with Host Encryption using encryption AtHost property.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "encryptionAtHost": true
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
```

```
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "encryptionAtHost": true,
      "securityType": "Standard"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
```

```
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "encryptionAtHost": true,
      "securityType": "Standard"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a vm with managed boot diagnostics.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
```

```
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "enabled": true
      }
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
```

```
  "location": "westus"
}
```

Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a VM with network interface configuration

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkApiVersion": "2020-11-01",
      "networkInterfaceConfigurations": [
        {
          "name": "{nic-config-name}",
          "tags": {
            "nicTag": "tag"
          },
          "properties": {
            "primary": true,
            "deleteOption": "Delete",
            "ipConfigurations": [
              {
                "name": "{ip-config-name}",
                "properties": {
                  "primary": true,
                  "publicIPAddressConfiguration": {
                    "name": "{publicIP-config-name}",
                    "tags": {
                      "pipTag": "tag"
                    },
                    "sku": {
                      "name": "Basic",
                      "tier": "Global"
                    },
                    "properties": {
                      "deleteOption": "Detach",
                      "publicIPAllocationMethod": "Static"
                    }
                  }
                }
              }
            ]
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/toBeC
reatedNetworkInterface",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b7a098cc-b0b8-46e8-a205-62f301a62a8f",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/toBeC
```

```
reatedNetworkInterface",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b7a098cc-b0b8-46e8-a205-62f301a62a8f",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a VM with network interface configuration with public ip address dns settings

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkApiVersion": "2020-11-01",
      "networkInterfaceConfigurations": [
```

```
        {
          "name": "{nic-config-name}",
          "properties": {
            "primary": true,
            "deleteOption": "Delete",
            "ipConfigurations": [
              {
                "name": "{ip-config-name}",
                "properties": {
                  "primary": true,
                  "publicIPAddressConfiguration": {
                    "name": "{publicIP-config-name}",
                    "sku": {
                      "name": "Basic",
                      "tier": "Global"
                    },
                    "properties": {
                      "deleteOption": "Detach",
                      "publicIPAllocationMethod": "Static",
                      "dnsSettings": {
                        "domainNameLabel": "aaaaa",
                        "domainNameLabelScope": "TenantReuse"
                      }
                    }
                  }
                }
              }
            ]
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/toBeC
reatedNetworkInterface",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
```

```
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b7a098cc-b0b8-46e8-a205-62f301a62a8f",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/toBeC
reatedNetworkInterface",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b7a098cc-b0b8-46e8-a205-62f301a62a8f",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
```

```
  "name": "myVM",
  "location": "westus"
}
```

Create a vm with password authentication.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
```

```
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b248db33-62ba-4d2d-b791-811e075ee0f5",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
```

```
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "b248db33-62ba-4d2d-b791-811e075ee0f5",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a vm with premium storage.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
```

```
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
```

```
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a VM with ProxyAgent Settings of enabled and mode.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": true,
        "wireServer": {
          "mode": "Audit"
        },
```

```
        "imds": {
          "mode": "Audit"
        }
      }
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
```

```
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": true,
        "wireServer": {
          "mode": "Audit"
        },
        "imds": {
          "mode": "Audit"
        }
      },
      "securityType": "Standard"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "MicrosoftWindowsServer",
        "offer": "WindowsServer",
        "sku": "2019-Datacenter",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
```

```
      "proxyAgentSettings": {
        "enabled": true,
        "wireServer": {
          "mode": "Audit"
        },
        "imds": {
          "mode": "Audit"
        }
      },
      "securityType": "Standard"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a vm with resiliency profile and zone movement enabled.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_ZRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      },
      "dataDisks": [
        {
          "diskSizeGB": 128,
          "lun": 0,
          "createOption": "Empty",
          "managedDisk": {
            "storageAccountType": "Premium_ZRS"
          },
          "name": "myVMdatadisk1",
          "caching": "ReadWrite"
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
```

```
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "resiliencyProfile": {
      "zoneMovement": {
        "isEnabled": true
      }
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_ZRS"
        }
      },
      "dataDisks": [
        {
          "diskSizeGB": 128,
          "lun": 0,
          "createOption": "Empty",
          "managedDisk": {
            "storageAccountType": "Premium_ZRS"
          },
          "name": "myVMdatadisk1",
          "caching": "ReadWrite"
        }
      ]
```

```
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "resiliencyProfile": {
      "zoneMovement": {
        "isEnabled": true
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_ZRS"
        }
      },
      "dataDisks": [
        {
          "diskSizeGB": 128,
          "lun": 0,
          "createOption": "Empty",
          "managedDisk": {
            "storageAccountType": "Premium_ZRS"
          },
          "name": "myVMdatadisk1",
          "caching": "ReadWrite"
        }
      ]
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
```

```
    },
    "resiliencyProfile": {
      "zoneMovement": {
        "isEnabled": true
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a vm with Scheduled Events Profile

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "scheduledEventsPolicy": {
      "scheduledEventsAdditionalPublishingTargets": {
        "eventGridAndResourceGraph": {
          "enable": true,
          "scheduledEventsApiVersion": "2020-07-01"
        }
      },
      "userInitiatedRedeploy": {
        "automaticallyApprove": true
      },
      "userInitiatedReboot": {
        "automaticallyApprove": true
      },
      "allInstancesDown": {
        "automaticallyApprove": true
      }
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
```

```
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
    },
    "scheduledEventsProfile": {
      "terminateNotificationProfile": {
        "notBeforeTimeout": "PT10M",
        "enable": true
      },
      "osImageNotificationProfile": {
        "notBeforeTimeout": "PT15M",
        "enable": true
      }
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
```

```
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "scheduledEventsPolicy": {
      "scheduledEventsAdditionalPublishingTargets": {
        "eventGridAndResourceGraph": {
          "enable": true,
          "scheduledEventsApiVersion": "2020-07-01"
        }
      },
      "userInitiatedRedeploy": {
        "automaticallyApprove": true
      },
      "userInitiatedReboot": {
        "automaticallyApprove": true
      },
      "allInstancesDown": {
        "automaticallyApprove": true
      }
    },
    "scheduledEventsProfile": {
      "terminateNotificationProfile": {
        "notBeforeTimeout": "PT10M",
        "enable": true
      },
      "osImageNotificationProfile": {
        "notBeforeTimeout": "PT15M",
        "enable": true
      }
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
```

```
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "scheduledEventsPolicy": {
      "scheduledEventsAdditionalPublishingTargets": {
        "eventGridAndResourceGraph": {
          "enable": true,
          "scheduledEventsApiVersion": "2020-07-01"
        }
      },
      "userInitiatedRedeploy": {
        "automaticallyApprove": true
      },
      "userInitiatedReboot": {
        "automaticallyApprove": true
      },
      "allInstancesDown": {
        "automaticallyApprove": true
      }
    },
    "scheduledEventsProfile": {
      "terminateNotificationProfile": {
        "notBeforeTimeout": "PT10M",
        "enable": true
      },
      "osImageNotificationProfile": {
        "notBeforeTimeout": "PT15M",
        "enable": true
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a VM with securityType ConfidentialVM with Customer Managed Keys

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DC2as_v5"
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
```

```
      },
      "securityType": "ConfidentialVM"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows-cvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "17763.2183.2109130127",
        "offer": "2019-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "DiskWithVMGuestState",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets
/{existing-diskEncryptionSet-name}"
            }
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
```

```
        "sku": "windows-cvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "17763.2183.2109130127",
        "offer": "2019-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "DiskWithVMGuestState",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskEncryptionSet-name}"
            }
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DC2as_v5"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows-cvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "17763.2183.2109130127",
        "offer": "2019-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
```

```
          "securityProfile": {
            "securityEncryptionType": "DiskWithVMGuestState",
            "diskEncryptionSet": {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/diskEncryptionSets/{exi
sting-diskEncryptionSet-name}"
            }
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DC2as_v5"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Create a VM with securityType ConfidentialVM with NonPersistedTPM securityEncryptionType

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DC2es_v5"
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": false,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "linux-cvm",
        "publisher": "UbuntuServer",
        "version": "17763.2183.2109130127",
        "offer": "2022-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "NonPersistedTPM"
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
```

```
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "linux-cvm",
        "publisher": "UbuntuServer",
        "version": "17763.2183.2109130127",
        "offer": "2022-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "NonPersistedTPM"
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": false,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
```

```
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DC2es_v5"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "linux-cvm",
        "publisher": "UbuntuServer",
        "version": "17763.2183.2109130127",
        "offer": "2022-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "NonPersistedTPM"
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": false,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DC2es_v5"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Create a VM with securityType ConfidentialVM with Platform Managed Keys

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DC2as_v5"
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows-cvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "17763.2183.2109130127",
        "offer": "2019-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "DiskWithVMGuestState"
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "name": "myVM",
  "properties": {
```

```
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows-cvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "17763.2183.2109130127",
        "offer": "2019-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "DiskWithVMGuestState"
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DC2as_v5"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
```

```
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows-cvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "17763.2183.2109130127",
        "offer": "2019-datacenter-cvm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS",
          "securityProfile": {
            "securityEncryptionType": "DiskWithVMGuestState"
          }
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "ConfidentialVM"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DC2as_v5"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a vm with ssh authentication.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "{image_sku}",
        "publisher": "{image_publisher}",
        "version": "latest",
        "offer": "{image_offer}"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
```

```
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "linuxConfiguration": {
        "ssh": {
          "publicKeys": [
            {
              "path": "/home/{your-username}/.ssh/authorized_keys",
              "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCeClRAk2ipUs/l5voIsDC5q9RI+YSRd1Bvd/O+axgY4WiBzG+4F
wJWZm/mLLe5DoOdHQwmU2FrKXZSW4w2sYE70KeWnrFViCOX5MTVvJgPE8ClugNl8RWth/tU849DvM9sT7
vFgfVSHcAS2yDRyDlueii+8nF2ym8XWAPltFVCyLHRsyBp5YPqK8JFYIa1eybKsY3hEAxRCA+/7bq8et+
Gj3coOsuRmrehav7rE6N12Pb80I6ofa6SM5XNYq4Xk0iYNx7R3kdz0Jj9XgZYWjAHjJmT0gTRoOnt6upO
uxK7xI/ykWrllgpXrCPu3Ymz+c+ujaqcxDopnAl2lmf69/J1"
            }
          ]
        },
        "disablePasswordAuthentication": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "ssh": {
          "publicKeys": [
            {
              "path": "/home/{your-username}/.ssh/authorized_keys",
              "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCeClRAk2ipUs/l5voIsDC5q9RI+YSRd1Bvd/O+axgY4WiBzG+4FwJWZm
/mLLe5DoOdHQwmU2FrKXZSW4w2sYE70KeWnrFViCOX5MTVvJgPE8ClugNl8RWth/tU849DvM9sT7vFgfVSHcAS
2yDRyDlueii+8nF2ym8XWAPltFVCyLHRsyBp5YPqK8JFYIa1eybKsY3hEAxRCA+/7bq8et+Gj3coOsuRmrehav
7rE6N12Pb80I6ofa6SM5XNYq4Xk0iYNx7R3kdz0Jj9XgZYWjAHjJmT0gTRoOnt6upOuxK7xI/ykWrllgpXrCPu
3Ymz+c+ujaqcxDopnAl2lmf69/J1"
            }
          ]
        },
        "disablePasswordAuthentication": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
```

```
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "e0de9b84-a506-4b95-9623-00a425d05c90",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "linuxConfiguration": {
        "ssh": {
          "publicKeys": [
            {
              "path": "/home/{your-username}/.ssh/authorized_keys",
              "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCeClRAk2ipUs/l5voIsDC5q9RI+YSRd1Bvd/O+axgY4WiBzG+4FwJWZm
/mLLe5DoOdHQwmU2FrKXZSW4w2sYE70KeWnrFViCOX5MTVvJgPE8ClugNl8RWth/tU849DvM9sT7vFgfVSHcAS
2yDRyDlueii+8nF2ym8XWAPltFVCyLHRsyBp5YPqK8JFYIa1eybKsY3hEAxRCA+/7bq8et+Gj3coOsuRmrehav
7rE6N12Pb80I6ofa6SM5XNYq4Xk0iYNx7R3kdz0Jj9XgZYWjAHjJmT0gTRoOnt6upOuxK7xI/ykWrllgpXrCPu
3Ymz+c+ujaqcxDopnAl2lmf69/J1"
            }
          ]
        },
        "disablePasswordAuthentication": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "16.04-LTS",
        "publisher": "Canonical",
        "version": "latest",
```

```
        "offer": "UbuntuServer"
      },
      "osDisk": {
        "osType": "Linux",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "e0de9b84-a506-4b95-9623-00a425d05c90",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a VM with Uefi Settings of secureBoot and vTPM.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "TrustedLaunch"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows10-tvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "18363.592.2001092016",
        "offer": "windowsserver-gen2preview-preview"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
```

```
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows10-tvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "18363.592.2001092016",
        "offer": "windowsserver-gen2preview-preview"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "TrustedLaunch"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows10-tvm",
        "publisher": "MicrosoftWindowsServer",
        "version": "18363.592.2001092016",
        "offer": "windowsserver-gen2preview-preview"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "StandardSSD_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "securityProfile": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "TrustedLaunch"
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_D2s_v3"
    },
    "provisioningState": "Creating"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Create a VM with UserData

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{v
m-name}?api-version=2025-11-01
```

```
{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "vmOSdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "{vm-name}",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
    },
    "userData": "RXhhbXBsZSBVc2VyRGF0YQ=="
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-
name}",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "{vm-name}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
```

```
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "vmOSdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "{vm-name}",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/{vm-
name}",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "{vm-name}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
```

```
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "vmOSdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "{vm-name}",
  "location": "westus"
}
```

Create a VM with VM Size Properties

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D4_v3",
      "vmSizeProperties": {
        "vCPUsAvailable": 1,
        "vCPUsPerCore": 1
      }
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
```

```
      ]
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
        "enabled": true
      }
    },
    "userData": "U29tZSBDdXN0b20gRGF0YQ=="
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D4_v3",
      "vmSizeProperties": {
        "vCPUsAvailable": 1,
```

```
        "vCPUsPerCore": 1
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Updating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        }
      },
      "dataDisks": []
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "storageUri": "http://nsgdiagnostic.blob.core.windows.net",
        "enabled": true
      }
    },
    "vmId": "676420ba-7a24-4bfe-80bd-9c841ee184fa",
    "hardwareProfile": {
      "vmSize": "Standard_D4_v3",
      "vmSizeProperties": {
        "vCPUsAvailable": 1,
        "vCPUsPerCore": 1
      }
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
```

```
  "location": "westus"
}
```

Create a Windows vm with a patch setting assessment Mode of ImageDefault.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "assessmentMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
```

```
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "assessmentMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": false,
        "patchSettings": {
          "assessmentMode": "ImageDefault"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
```

```
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a Windows vm with a patch setting patchMode of AutomaticByOS.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
```

```
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByOS"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
nsgExistingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByOS"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
```

```
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByOS"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create a Windows vm with a patch setting patch Mode of AutomaticByPlatform and Automatic ByPlatformSettings.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform",
          "automaticByPlatformSettings": {
            "rebootSetting": "Never",
            "bypassPlatformSafetyChecksOnUserSchedule": false
          }
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform",
          "automaticByPlatformSettings": {
            "rebootSetting": "Never",
            "bypassPlatformSafetyChecksOnUserSchedule": false
          }
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
```

```
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform",
          "automaticByPlatformSettings": {
            "rebootSetting": "Never",
            "bypassPlatformSafetyChecksOnUserSchedule": false
          }
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a Windows vm with a patch setting patchMode of AutomaticByPlatform and enableHotpatching set to true.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
```

```
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "enableHotpatching": true
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "enableHotpatching": true
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
```

```
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "enableHotpatching": true
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
```

```
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a Windows vm with a patch setting patchMode of Manual.

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "Manual"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
```

```
}
```

Sample response

Status code: 200

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "Manual"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
```

```
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": false,
        "patchSettings": {
          "patchMode": "Manual"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Create a Windows vm with patch settings patchMode and assessmentMode set to AutomaticByPlatform.

Sample request

HTTP

HTTP


```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_D1_v2"
    },
    "storageProfile": {
      "imageReference": {
```

```
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        },
        "name": "myVMosdisk",
        "createOption": "FromImage"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
  }
}
```

Sample response

Status code: 200

JSON

```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
```

```
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```


Status code: 201

JSON


```
{
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "type": "Microsoft.Compute/virtualMachines",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform",
          "assessmentMode": "AutomaticByPlatform"
        }
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "2016-Datacenter",
        "publisher": "MicrosoftWindowsServer",
        "version": "latest",
        "offer": "WindowsServer"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadWrite",
        "createOption": "FromImage",
        "name": "myVMosdisk",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },
      "dataDisks": []
```

```
    },
    "vmId": "a149cd25-409f-41af-8088-275f5486bc93",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "name": "myVM",
  "location": "westus"
}
```

Create or update a VM with capacity reservation

Sample request

HTTP

HTTP

```
PUT https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/my
VM?api-version=2025-11-01

{
  "location": "westus",
  "plan": {
    "publisher": "microsoft-ads",
    "product": "windows-data-science-vm",
    "name": "windows2016"
  },
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "storageProfile": {
      "imageReference": {
        "sku": "windows2016",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "windows-data-science-vm"
      },
      "osDisk": {
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      }
    },
    "capacityReservation": {
      "capacityReservationGroup": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/CapacityReservatio
nGroups/{crgName}"
      }
    },
    "osProfile": {
      "adminUsername": "{your-username}",
      "computerName": "myVM",
      "adminPassword": "{your-password}"
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/
{existing-nic-name}",
          "properties": {
            "primary": true
          }
        }
      ]
    }
```

```
    }
}
```

Sample response

Status code: 200

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "capacityReservation": {
      "capacityReservationGroup": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/CapacityReservationGrou
ps/{crgName}"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```

Status code: 201

JSON


```
{
  "name": "myVM",
  "properties": {
    "osProfile": {
      "adminUsername": "{your-username}",
      "secrets": [],
      "computerName": "myVM",
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true
      }
    },
    "capacityReservation": {
      "capacityReservationGroup": {
        "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/CapacityReservationGrou
ps/{crgName}"
      }
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterfaces/nsgEx
istingNic",
          "properties": {
            "primary": true
          }
        }
      ]
    },
    "storageProfile": {
      "imageReference": {
        "sku": "standard-data-science-vm",
        "publisher": "microsoft-ads",
        "version": "latest",
        "offer": "standard-data-science-vm"
      },
      "osDisk": {
        "osType": "Windows",
        "caching": "ReadOnly",
        "managedDisk": {
          "storageAccountType": "Standard_LRS"
        },
        "createOption": "FromImage",
        "name": "myVMosdisk"
      },
      "dataDisks": []
    },
    "vmId": "5c0d55a7-c407-4ed6-bf7d-ddb810267c85",
    "hardwareProfile": {
      "vmSize": "Standard_DS1_v2"
    },
    "securityProfile": {
      "securityType": "Standard"
    },
    "provisioningState": "Creating"
  },
  "plan": {
    "publisher": "microsoft-ads",
    "product": "standard-data-science-vm",
    "name": "standard-data-science-vm"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "westus"
}
```


Definitions

Expand table

Name	Description

Additional Capabilities	Enables or disables a capability on the virtual machine or virtual machine scale set.
Additional UnattendContent	Specifies additional XML formatted information that can be included in the Unattend.xml file, which is used by Windows Setup. Contents are defined by setting name, component name, and the pass in which the content is applied.
AllInstancesDown	Specifies if Scheduled Events should be auto-approved when all instances are down.
ApiEntity Reference	The API entity reference.
ApiError	Api error.
ApiErrorBase	Api error base.
ApplicationProfile	Contains the list of gallery applications that should be made available to the VM/VMSS
AvailablePatch Summary	Describes the properties of an virtual machine instance view for available patch summary.
BillingProfile	Specifies the billing related details of a Azure Spot VM or VMSS. Minimum api-version: 2019-03-01.
BootDiagnostics	Boot Diagnostics is a debugging feature which allows you to view Console Output and Screenshot to diagnose VM status. You can easily view the output of your console log. Azure also enables you to see a screenshot of the VM from the hypervisor.
BootDiagnostics InstanceView	The instance view of a virtual machine boot diagnostics.
CachingTypes	Specifies the caching requirements. Possible values are: **None, ReadOnly, ReadWrite.** The default values are: **None for Standard storage. ReadOnly for Premium storage**
Capacity ReservationProfile	The parameters of a capacity reservation Profile.
CloudError	An error response from the Compute service.
Common.User AssignedIdentities Value	
Component Names	
createdByType	The type of identity that created the resource.
DataDisk	Describes a data disk.
DeleteOptions	Specify what happens to the network interface when the VM is deleted
DiagnosticsProfile	Specifies the boot diagnostic settings state. Minimum api-version: 2015-06-15.
DiffDiskOptions	Specifies the ephemeral disk option for operating system disk.
DiffDiskPlacement	Specifies the ephemeral disk placement for operating system disk. This property can be used by user in the request to choose the location i.e, cache disk, resource disk or nvme disk space for Ephemeral OS disk provisioning. For more information on Ephemeral OS disk size requirements, please refer Ephemeral OS disk size requirements for Windows VM at https://docs.microsoft.com/azure/virtual-machines/windows/ephemeral-os-disks#size-requirements and Linux VM at https://docs.microsoft.com/azure/virtual-machines/linux/ephemeral-os-disks#size-requirements. Minimum api-version for NvmeDisk: 2024-03-01.
DiffDiskSettings	Describes the parameters of ephemeral disk settings that can be specified for operating system disk. **Note:** The ephemeral disk settings can only be specified for managed disk.
DiskController Types	Specifies the disk controller type configured for the VM and VirtualMachineScaleSet. This property is only supported for virtual machines whose operating system disk and VM sku supports Generation 2 (https://docs.microsoft.com/en-us/azure/virtual-machines/generation-2), please check the HyperVGenerations capability returned as part of VM sku capabilities in the response of Microsoft.Compute SKUs api for the region contains V2 (https://docs.microsoft.com/rest/api/compute/resourceskus/list). For more information about Disk Controller Types supported please refer to https://aka.ms/azure-diskcontrollertypes .

DiskCreateOption Types	Specifies how the virtual machine disk should be created. Possible values are **Attach:** This value is used when you are using a specialized disk to create the virtual machine. **FromImage:** This value is used when you are using an image to create the virtual machine. If you are using a platform image, you should also use the imageReference element described above. If you are using a marketplace image, you should also use the plan element previously described. **Empty:** This value is used when creating an empty data disk. **Copy:** This value is used to create a data disk from a snapshot or another disk. **Restore:** This value is used to create a data disk from a disk restore point.
DiskDeleteOption Types	Specifies the behavior of the managed disk when the VM gets deleted, for example whether the managed disk is deleted or detached. Supported values are: **Delete.** If this value is used, the managed disk is deleted when VM gets deleted. **Detach.** If this value is used, the managed disk is retained after VM gets deleted. Minimum api-version: 2021-03-01.
DiskDetachOption Types	Specifies the detach behavior to be used while detaching a disk or which is already in the process of detachment from the virtual machine. Supported values are: **ForceDetach.** detachOption: **ForceDetach** is applicable only for managed data disks. If a previous detachment attempt of the data disk did not complete due to an unexpected failure from the virtual machine and the disk is still not released then use force-detach as a last resort option to detach the disk forcibly from the VM. All writes might not have been flushed when using this detach behavior. **This feature is still in preview**. To force-detach a data disk update toBeDetached to 'true' along with setting detachOption: 'ForceDetach'.
DiskEncryptionSet Parameters	Describes the parameter of customer managed disk encryption set resource id that can be specified for disk. **Note:** The disk encryption set resource id can only be specified for managed disk. Please refer https://aka.ms/mdssewithcmkoverview for more details.
DiskEncryption Settings	Describes a Encryption Settings for a Disk
DiskInstanceView	The instance view of the disk.
DomainName LabelScopeTypes	The Domain name label scope.The concatenation of the hashed domain name label that generated according to the policy from domain name label scope and vm index will be the domain name labels of the PublicIPAddress resources that will be created
EncryptionIdentity	Specifies the Managed Identity used by ADE to get access token for keyvault operations.
EventGridAnd ResourceGraph	Specifies eventGridAndResourceGraph related Scheduled Event related configurations.
ExtendedLocation	The complex type of the extended location.
ExtendedLocation Types	The type of the extended location.
HardwareProfile	Specifies the hardware settings for the virtual machine.
HostEndpoint Settings	Specifies particular host endpoint settings.
HyperVGeneration Type	Specifies the HyperVGeneration Type associated with a resource
ImageReference	Specifies information about the image to use. You can specify information about platform images, marketplace images, or virtual machine images. This element is required when you want to use a platform image, marketplace image, or virtual machine image, but is not used in other creation operations. NOTE: Image reference publisher and offer can only be set when you create the scale set.
InnerError	Inner error details.
InstanceView Status	Instance view status.
IPVersions	Available from Api-Version 2017-03-30 onwards, it represents whether the specific ipconfiguration is IPv4 or IPv6. Default is taken as IPv4. Possible values are: 'IPv4' and 'IPv6'.
KeyVaultKey Reference	Describes a reference to Key Vault Key
KeyVaultSecret Reference	Describes a reference to Key Vault Secret

LastPatch Installation Summary	Describes the properties of the last installed patch summary.
Linux Configuration	Specifies the Linux operating system settings on the virtual machine. For a list of supported Linux distributions, see Linux on Azure-Endorsed Distributions.
LinuxPatch AssessmentMode	Specifies the mode of VM Guest Patch Assessment for the IaaS virtual machine. Possible values are: **ImageDefault** - You control the timing of patch assessments on a virtual machine. **AutomaticByPlatform** - The platform will trigger periodic patch assessments. The property provisionVMAgent must be true.
LinuxPatch Settings	Specifies settings related to VM Guest Patching on Linux.
LinuxVMGuest PatchAutomatic ByPlatformReboot Setting	Specifies the reboot setting for all AutomaticByPlatform patch installation operations.
LinuxVMGuest PatchAutomatic ByPlatform Settings	Specifies additional settings to be applied when patch mode AutomaticByPlatform is selected in Linux patch settings.
LinuxVMGuest PatchMode	Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible. Possible values are: **ImageDefault** - The virtual machine's default patching configuration is used. **AutomaticByPlatform** - The virtual machine will be automatically updated by the platform. The property provisionVMAgent must be true
Maintenance OperationResult CodeTypes	The Last Maintenance Operation Result Code.
Maintenance RedeployStatus	Maintenance Operation Status.
ManagedDisk Parameters	The parameters of a managed disk.
Mode	Specifies the mode that ProxyAgent will execute on if the feature is enabled. ProxyAgent will start to audit or monitor but not enforce access control over requests to host endpoints in Audit mode, while in Enforce mode it will enforce access control. The default value is Enforce mode.
Modes	Specifies the execution mode. In Audit mode, the system acts as if it is enforcing the access control policy, including emitting access denial entries in the logs but it does not actually deny any requests to host endpoints. In Enforce mode, the system will enforce the access control and it is the recommended mode of operation.
NetworkApi Version	specifies the Microsoft.Network API version used when creating networking resources in the Network Interface Configurations
NetworkInterface AuxiliaryMode	Specifies whether the Auxiliary mode is enabled for the Network Interface resource.
NetworkInterface AuxiliarySku	Specifies whether the Auxiliary sku is enabled for the Network Interface resource.
NetworkInterface Reference	Describes a network interface reference.
NetworkProfile	Specifies the network interfaces or the networking configuration of the virtual machine.
OperatingSystem Types	This property allows you to specify the type of the OS that is included in the disk if creating a VM from a custom image. Possible values are: **Windows, Linux.**

OSDisk	Specifies information about the operating system disk used by the virtual machine. For more information about disks, see About disks and VHDs for Azure virtual machines.
OSImage NotificationProfile	
OSProfile	Specifies the operating system settings for the virtual machine. Some of the settings cannot be changed once VM is provisioned.
PassNames	
PatchOperation Status	The overall success or failure status of the operation. It remains "InProgress" until the operation completes. At that point it will become "Unknown", "Failed", "Succeeded", or "CompletedWithWarnings."
PatchSettings	Specifies settings related to VM Guest Patching on Windows.
Placement	Describes the user-defined constraints for resource hardware placement.
Plan	Specifies information about the marketplace image used to create the virtual machine. This element is only used for marketplace images. Before you can use a marketplace image from an API, you must enable the image for programmatic use. In the Azure portal, find the marketplace image that you want to use and then click **Want to deploy programmatically, Get Started ->**. Enter any required information and then click **Save**.
ProtocolTypes	Specifies the protocol of WinRM listener. Possible values are: **http, https.**
ProxyAgent Settings	Specifies ProxyAgent settings for the virtual machine or virtual machine scale set. Minimum api-version: 2023-09-01.
PublicIPAddress Sku	Describes the public IP Sku. It can only be set with OrchestrationMode as Flexible.
PublicIPAddress SkuName	Specify public IP sku name
PublicIPAddress SkuTier	Specify public IP sku tier
PublicIPAllocation Method	Specify the public IP allocation type
ResiliencyProfile	Gets resiliency solutions enabled on the VM. This includes backup or disaster recovery solutions.
ResourceIdentity Type	The type of identity used for the virtual machine. The type 'SystemAssigned, UserAssigned' includes both an implicitly created identity and a set of user assigned identities. The type 'None' will remove any identities from the virtual machine.
ScheduledEvents Additional PublishingTargets	
ScheduledEvents Policy	Specifies Redeploy, Reboot and ScheduledEventsAdditionalPublishingTargets Scheduled Event related configurations.
ScheduledEvents Profile	
SecurityEncryption Types	Specifies the EncryptionType of the managed disk. It is set to DiskWithVMGuestState for encryption of the managed disk along with VMGuestState blob, VMGuestStateOnly for encryption of just the VMGuestState blob, and NonPersistedTPM for not persisting firmware state in the VMGuestState blob.. **Note:** It can be set for only Confidential VMs.
SecurityProfile	Specifies the Security profile settings for the virtual machine or virtual machine scale set.
SecurityTypes	Specifies the VM securityType; UefiSettings are enabled only when set to TrustedLaunch or ConfidentialVM, and returns a Standard value starting API version 2025-11-01.
SettingNames	Specifies the name of the setting to which the content applies. Possible values are: FirstLogonCommands and AutoLogon.
SshConfiguration	SSH configuration for Linux based VMs running on Azure
SshPublicKey	Contains information about SSH certificate public key and the path on the Linux VM where the public key is placed.

StatusLevelTypes	The level code.
StorageAccountTypes	Specifies the storage account type for the managed disk. Managed OS disk storage account type can only be set when you create the scale set. NOTE: UltraSSD_LRS can only be used with data disks. It cannot be used with OS Disk. Standard_LRS uses Standard HDD. StandardSSD_LRS uses Standard SSD. Premium_LRS uses Premium SSD. UltraSSD_LRS uses Ultra disk. Premium_ZRS uses Premium SSD zone redundant storage. StandardSSD_ZRS uses Standard SSD zone redundant storage. For more information regarding disks supported for Windows Virtual Machines, refer to https://docs.microsoft.com/azure/virtual-machines/windows/disks-types and, for Linux Virtual Machines, refer to https://docs.microsoft.com/azure/virtual-machines/linux/disks-types
StorageAlignmentStatus	Specifies the storage alignment status for the disk.
StorageFaultDomainAlignmentType	Specifies the storage fault domain alignment type for the disk.
StorageProfile	Specifies the storage settings for the virtual machine disks.
SubResource	
systemData	Metadata pertaining to creation and last modification of the resource.
TerminateNotificationProfile	
UefiSettings	Specifies the security settings like secure boot and vTPM used while creating the virtual machine. Minimum api-version: 2020-12-01.
UserInitiatedReboot	Specifies Reboot related Scheduled Event related configurations.
UserInitiatedRedeploy	Specifies Redeploy related Scheduled Event related configurations.
VaultCertificate	Describes a single certificate reference in a Key Vault, and where the certificate should reside on the VM.
VaultSecretGroup	Describes a set of certificates which are all in the same Key Vault.
VirtualHardDisk	Describes the uri of a disk.
VirtualMachine	Describes a Virtual Machine.
VirtualMachineAgentInstanceView	The instance view of the VM Agent running on the virtual machine.
VirtualMachineEvictionPolicyTypes	Specifies the eviction policy for the Azure Spot VM/VMSS
VirtualMachineExtension	Describes a Virtual Machine Extension.
VirtualMachineExtensionHandlerInstanceView	The instance view of a virtual machine extension handler.
VirtualMachineExtensionInstanceView	The instance view of a virtual machine extension.
VirtualMachineHealthStatus	The health status of the VM.
VirtualMachineIdentity	Identity for the virtual machine.
VirtualMachineInstanceView	The instance view of a virtual machine.
VirtualMachineIpTag	Contains the IP tag associated with the public IP address.

VirtualMachine NetworkInterface Configuration	Describes a virtual machine network interface configurations.
VirtualMachine NetworkInterface DnsSettings Configuration	Describes a virtual machines network configuration's DNS settings.
VirtualMachine NetworkInterface IPConfiguration	Describes a virtual machine network profile's IP configuration.
VirtualMachine PatchStatus	The status of virtual machine patch operations.
VirtualMachine PriorityTypes	Specifies the priority for a standalone virtual machine or the virtual machines in the scale set. 'Low' enum will be deprecated in the future, please use 'Spot' as the enum to deploy Azure Spot VM/VMSS.
VirtualMachine PublicIPAddress Configuration	Describes a virtual machines IP Configuration's PublicIPAddress configuration
VirtualMachine PublicIPAddress DnsSettings Configuration	Describes a virtual machines network configuration's DNS settings.
VirtualMachine SizeTypes	Specifies the size of the virtual machine. The enum data type is currently deprecated and will be removed by December 23rd 2023. The recommended way to get the list of available sizes is using these APIs: List all available virtual machine sizes in an availability set, List all available virtual machine sizes in a region, List all available virtual machine sizes for resizing. For more information about virtual machine sizes, see Sizes for virtual machines. The available VM sizes depend on region and availability set.
VMDiskSecurity Profile	Specifies the security profile settings for the managed disk. **Note:** It can only be set for Confidential VMs.
VMGallery Application	Specifies the required information to reference a compute gallery application version
VMSizeProperties	Specifies VM Size Property settings on the virtual machine.
Windows Configuration	Specifies Windows operating system settings on the virtual machine.
WindowsPatch AssessmentMode	Specifies the mode of VM Guest patch assessment for the IaaS virtual machine. Possible values are: **ImageDefault** - You control the timing of patch assessments on a virtual machine. **AutomaticByPlatform** - The platform will trigger periodic patch assessments. The property provisionVMAgent must be true.
WindowsVMGuest PatchAutomatic ByPlatformReboot Setting	Specifies the reboot setting for all AutomaticByPlatform patch installation operations.
WindowsVMGuest PatchAutomatic ByPlatform Settings	Specifies additional settings to be applied when patch mode AutomaticByPlatform is selected in Windows patch settings.
WindowsVMGuest PatchMode	Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible. Possible values are: **Manual** - You control the application of patches to a virtual machine. You do this by applying patches manually inside the VM. In this mode, automatic updates are disabled; the property WindowsConfiguration.enableAutomaticUpdates must be false **AutomaticByOS** - The virtual machine will automatically be updated by the OS. The property WindowsConfiguration.enableAutomaticUpdates must be true.

	AutomaticByPlatform - the virtual machine will automatically updated by the platform. The properties provisionVMAgent and WindowsConfiguration.enableAutomaticUpdates must be true
Win RMConfiguration	Describes Windows Remote Management configuration of the VM
WinRMListener	Describes Protocol and thumbprint of Windows Remote Management listener
ZoneMovement	Describes zone movement configuration. This allows VM to be moved across availability zones during an outage.
ZonePlacement PolicyType	Specifies the policy for resource's placement in availability zone. Possible values are: **Any** (used for Virtual Machines), **Auto** (used for Virtual Machine Scale Sets) - An availability zone will be automatically picked by system as part of resource creation.

AdditionalCapabilities

Object

Enables or disables a capability on the virtual machine or virtual machine scale set.

Expand table

Name	Type	Description
enableFips1403Encryption	boolean	The flag enables the usage of FIPS 140-3 compliant cryptography on the protectedSettings of an extension. Learn more at: https://aka.ms/linuxagentfipssupport.
hibernationEnabled	boolean	The flag that enables or disables hibernation capability on the VM.
ultraSSDEnabled	boolean	The flag that enables or disables a capability to have one or more managed data disks with UltraSSD_LRS storage account type on the VM or VMSS. Managed disks with storage account type UltraSSD_LRS can be added to a virtual machine or virtual machine scale set only if this property is enabled.

AdditionalUnattendContent

Object

Specifies additional XML formatted information that can be included in the Unattend.xml file, which is used by Windows Setup. Contents are defined by setting name, component name, and the pass in which the content is applied.

Expand table

Name	Type	Description
componentName	Component Names	The component name. Currently, the only allowable value is Microsoft-Windows-Shell-Setup.
content	string	Specifies the XML formatted content that is added to the unattend.xml file for the specified path and component. The XML must be less than 4KB and must include the root element for the setting or feature that is being inserted.
passName	PassNames	The pass name. Currently, the only allowable value is OobeSystem.
settingName	SettingNames	Specifies the name of the setting to which the content applies. Possible values are: FirstLogonCommands and AutoLogon.

AllInstancesDown

Object

Specifies if Scheduled Events should be auto-approved when all instances are down.

Expand table

Name	Type	Description
automaticallyApprove	boolean	Specifies if Scheduled Events should be auto-approved when all instances are down. its default value is true

ApiEntityReference

Object

The API entity reference.

Expand table

Name	Type	Description
id	string	The ARM resource id in the form of /subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/...

ApiError

Object

Api error.

Expand table

Name	Type	Description
code	string	The error code.
details	ApiError Base[]	The Api error details
innererror	InnerError	The Api inner error
message	string	The error message.
target	string	The target of the particular error.

ApiErrorBase

Object

Api error base.

Expand table

Name	Type	Description
code	string	The error code.
message	string	The error message.
target	string	The target of the particular error.

ApplicationProfile

Object

Contains the list of gallery applications that should be made available to the VM/VMSS

Expand table

Name	Type	Description
galleryApplications	VMGallery Application[]	Specifies the gallery applications that should be made available to the VM/VMSS

AvailablePatchSummary

Object

Describes the properties of an virtual machine instance view for available patch summary.

Expand table

Name	Type	Description
assessmentActivityId	string	The activity ID of the operation that produced this result. It is used to correlate across CRP and extension logs.
criticalAndSecurityPatchCount	integer (int32)	The number of critical or security patches that have been detected as available and not yet installed.
error	ApiError	The errors that were encountered during execution of the operation. The details array contains the list of them.
lastModifiedTime	string (date-time)	The UTC timestamp when the operation began.
otherPatchCount	integer (int32)	The number of all available patches excluding critical and security.
rebootPending	boolean	The overall reboot status of the VM. It will be true when partially installed patches require a reboot to complete installation but the reboot has not yet occurred.
startTime	string (date-time)	The UTC timestamp when the operation began.
status	Patch Operation Status	The overall success or failure status of the operation. It remains "InProgress" until the operation completes. At that point it will become "Unknown", "Failed", "Succeeded", or "CompletedWithWarnings."

BillingProfile

Object

Specifies the billing related details of a Azure Spot VM or VMSS. Minimum api-version: 2019-03-01.

Expand table

Name	Type	Description
maxPrice	number (double)	Specifies the maximum price you are willing to pay for a Azure Spot VM/VMSS. This price is in US Dollars. This price will be compared with the current Azure Spot price for the VM size. Also, the prices are compared at the time of create/update of Azure Spot VM/VMSS and the operation will only succeed if the maxPrice is greater than the current Azure Spot price. The maxPrice will also be used for evicting a Azure Spot VM/VMSS if the current Azure Spot price goes beyond the maxPrice after creation of VM/VMSS. Possible values are: - Any decimal value greater than zero. Example: 0.01538 -1 – indicates default price to be up-to on-demand. You can set the maxPrice to -1 to indicate that the Azure Spot VM/VMSS should not be evicted for price reasons. Also, the default max price is -1 if it is not provided by you. Minimum api-version: 2019-03-01.

BootDiagnostics

Object

Boot Diagnostics is a debugging feature which allows you to view Console Output and Screenshot to diagnose VM status. You can easily view the output of your console log. Azure also enables you to see a screenshot of the VM from the hypervisor.

Expand table

Name	Type	Description
enabled	boolean	Whether boot diagnostics should be enabled on the Virtual Machine.
storageUri	string	Uri of the storage account to use for placing the console output and screenshot. If storageUri is not specified while enabling boot diagnostics, managed storage will be used.

BootDiagnosticsInstanceView

Object

The instance view of a virtual machine boot diagnostics.

Expand table

Name	Type	Description
consoleScreenshotBlobUri	string	The console screenshot blob URI. **Note:** This will **not** be set if boot diagnostics is currently enabled with managed storage.
serialConsoleLogBlobUri	string	The serial console log blob Uri. **Note:** This will **not** be set if boot diagnostics is currently enabled with managed storage.
status	InstanceView Status	The boot diagnostics status information for the VM. **Note:** It will be set only if there are errors encountered in enabling boot diagnostics.

CachingTypes

Enumeration

Specifies the caching requirements. Possible values are: **None, ReadOnly, ReadWrite.** The default values are: **None for Standard storage. ReadOnly for Premium storage**

Expand table

Value	Description
None	
ReadOnly	
ReadWrite	

CapacityReservationProfile

Object

The parameters of a capacity reservation Profile.

Expand table

Name	Type	Description
capacityReservationGroup	SubResource	Specifies the capacity reservation group resource id that should be used for allocating the virtual machine or scaleset vm instances provided enough capacity has been reserved. Please refer to https://aka.ms/CapacityReservation for more details.

CloudError

Object

An error response from the Compute service.

Expand table

Name	Type	Description
error	ApiError	Api error.

Common.UserAssignedIdentitiesValue

Object

Expand table

Name	Type	Description
clientId	string	The client id of user assigned identity.
principalId	string	The principal id of user assigned identity.

ComponentNames

Enumeration

Expand table

Value	Description
Microsoft-Windows-Shell-Setup	

createdByType

Enumeration

The type of identity that created the resource.

Expand table

Value	Description
User	
Application	
ManagedIdentity	
Key	

DataDisk

Object

Describes a data disk.

Expand table

Name	Type	Description
caching	CachingTypes	Specifies the caching requirements. Possible values are: **None, ReadOnly, ReadWrite.** The defaulting behavior is: **None for Standard storage. ReadOnly for Premium storage.**
createOption	DiskCreateOptionTypes	Specifies how the virtual machine disk should be created. Possible values are **Attach:** This value is used when you are using a specialized disk to create the virtual machine. **FromImage:** This value is used when you are using an image to create the virtual machine data disk. If you are using a platform image, you should also use the imageReference element described above. If you are using

		a marketplace image, you should also use the plan element previously described. **Empty:** This value is used when creating an empty data disk. **Copy:** This value is used to create a data disk from a snapshot or another disk. **Restore:** This value is used to create a data disk from a disk restore point.
deleteOption	DiskDeleteOptionTypes	Specifies whether data disk should be deleted or detached upon VM deletion. Possible values are: **Delete.** If this value is used, the data disk is deleted when VM is deleted. **Detach.** If this value is used, the data disk is retained after VM is deleted. The default value is set to **Detach**.
detachOption	DiskDetachOptionTypes	Specifies the detach behavior to be used while detaching a disk or which is already in the process of detachment from the virtual machine. Supported values: **ForceDetach.** detachOption: **ForceDetach** is applicable only for managed data disks. If a previous detachment attempt of the data disk did not complete due to an unexpected failure from the virtual machine and the disk is still not released then use force-detach as a last resort option to detach the disk forcibly from the VM. All writes might not have been flushed when using this detach behavior. **This feature is still in preview**. To force-detach a data disk update toBeDetached to 'true' along with setting detachOption: 'ForceDetach'.
diskIOPSReadWrite	integer (int64)	Specifies the Read-Write IOPS for the managed disk when StorageAccountType is UltraSSD_LRS.
diskMBpsReadWrite	integer (int64)	Specifies the bandwidth in MB per second for the managed disk when StorageAccountType is UltraSSD_LRS.
diskSizeGB	integer (int32)	Specifies the size of an empty data disk in gigabytes. This element can be used to overwrite the size of the disk in a virtual machine image. The property 'diskSizeGB' is the number of bytes x 1024^3 for the disk and the value cannot be larger than 1023.
image	VirtualHardDisk	The source user image virtual hard disk. The virtual hard disk will be copied before being attached to the virtual machine. If SourceImage is provided, the destination virtual hard drive must not exist.
lun	integer (int32)	Specifies the logical unit number of the data disk. This value is used to identify data disks within the VM and therefore must be unique for each data disk attached to a VM.
managedDisk	ManagedDiskParameters	The managed disk parameters.
name	string	The disk name.
sourceResource	ApiEntityReference	The source resource identifier. It can be a snapshot, or disk restore point from which to create a disk.
storageFaultDomainAlignment	StorageFaultDomainAlignmentType	Specifies the storage fault domain alignment type for the disk.
toBeDetached	boolean	Specifies whether the data disk is in process of detachment from the VirtualMachine/VirtualMachineScaleset
vhd	VirtualHardDisk	The virtual hard disk.
writeAcceleratorEnabled	boolean	Specifies whether writeAccelerator should be enabled or disabled on the disk.

DeleteOptions

Enumeration

Specify what happens to the network interface when the VM is deleted

Expand table

Value	Description
Delete	
Detach	

DiagnosticsProfile

Object

Specifies the boot diagnostic settings state. Minimum api-version: 2015-06-15.

Expand table

Name	Type	Description
bootDiagnostics	Boot Diagnostics	Boot Diagnostics is a debugging feature which allows you to view Console Output and Screenshot to diagnose VM status. **NOTE**: If storageUri is being specified then ensure that the storage account is in the same region and subscription as the VM. You can easily view the output of your console log. Azure also enables you to see a screenshot of the VM from the hypervisor.

DiffDiskOptions

Enumeration

Specifies the ephemeral disk option for operating system disk.

Expand table

Value	Description
Local	

DiffDiskPlacement

Enumeration

Specifies the ephemeral disk placement for operating system disk. This property can be used by user in the request to choose the location i.e, cache disk, resource disk or nvme disk space for Ephemeral OS disk provisioning. For more information on Ephemeral OS disk size requirements, please refer Ephemeral OS disk size requirements for Windows VM at https://docs.microsoft.com/azure/virtual-machines/windows/ephemeral-os-disks#size-requirements and Linux VM at https://docs.microsoft.com/azure/virtual-machines/linux/ephemeral-os-disks#size-requirements. Minimum api-version for NvmeDisk: 2024-03-01.

Expand table

Value	Description
CacheDisk	
ResourceDisk	
NvmeDisk	

DiffDiskSettings

Object

Describes the parameters of ephemeral disk settings that can be specified for operating system disk. **Note:** The ephemeral disk settings can only be specified for managed disk.

Expand table

Name	Type	Description
enableFullCaching	boolean	Specifies whether or not to enable full caching for this VM which will cache the OS disk locally on the host and make this VM more resilient to storage outages
option	DiffDisk Options	Specifies the ephemeral disk settings for operating system disk.
placement	DiffDisk Placement	Specifies the ephemeral disk placement for operating system disk. Possible values are: **CacheDisk, ResourceDisk, NvmeDisk.** The defaulting behavior is: **CacheDisk** if one is configured for the VM size otherwise **ResourceDisk** or **NvmeDisk** is used. Refer to the VM size documentation for Windows VM at https://docs.microsoft.com/azure/virtual-machines/windows/sizes and Linux VM at https://docs.microsoft.com/azure/virtual-machines/linux/sizes to check which VM sizes exposes a cache disk. Minimum api-version for NvmeDisk: 2024-03-01.

DiskControllerTypes

Enumeration

Specifies the disk controller type configured for the VM and VirtualMachineScaleSet. This property is only supported for virtual machines whose operating system disk and VM sku supports Generation 2 (https://docs.microsoft.com/en-us/azure/virtual-machines/generation-2), please check the HyperVGenerations capability returned as part of VM sku capabilities in the response of Microsoft.Compute SKUs api for the region contains V2 (https://docs.microsoft.com/rest/api/compute/resourceskus/list). For more information about Disk Controller Types supported please refer to https://aka.ms/azure-diskcontrollertypes.

Expand table

Value	Description
SCSI	
NVMe	

DiskCreateOptionTypes

Enumeration

Specifies how the virtual machine disk should be created. Possible values are **Attach:** This value is used when you are using a specialized disk to create the virtual machine. **FromImage:** This value is used when you are using an image to create the virtual machine. If you are using a platform image, you should also use the imageReference element described above. If you are using a marketplace image, you should also use the plan element previously described. **Empty:** This value is used when creating an empty data disk. **Copy:** This value is used to create a data disk from a snapshot or another disk. **Restore:** This value is used to create a data disk from a disk restore point.

Expand table

Value	Description
FromImage	
Empty	
Attach	
Copy	
Restore	

DiskDeleteOptionTypes

Enumeration

Specifies the behavior of the managed disk when the VM gets deleted, for example whether the managed disk is deleted or detached. Supported values are: **Delete.** If this value is used, the managed disk is deleted when VM gets deleted. **Detach.** If this value is used, the managed disk is retained after VM gets deleted. Minimum api-version: 2021-03-01.

Expand table

Value	Description
Delete	
Detach	

DiskDetachOptionTypes

Enumeration

Specifies the detach behavior to be used while detaching a disk or which is already in the process of detachment from the virtual machine. Supported values are: **ForceDetach.** detachOption: **ForceDetach** is applicable only for managed data disks. If a previous detachment attempt of the data disk did not complete due to an unexpected failure from the virtual machine and the disk is still not released then use force-detach as a last resort option to detach the disk forcibly from the VM. All writes might not have been flushed when using this detach behavior. **This feature is still in preview**. To force-detach a data disk update toBeDetached to 'true' along with setting detachOption: 'ForceDetach'.

Expand table

Value	Description
ForceDetach	

DiskEncryptionSetParameters

Object

Describes the parameter of customer managed disk encryption set resource id that can be specified for disk. **Note:** The disk encryption set resource id can only be specified for managed disk. Please refer https://aka.ms/mdssewithcmkoverview for more details.

Expand table

Name	Type	Description
id	string	Resource Id

DiskEncryptionSettings

Object

Describes a Encryption Settings for a Disk

Expand table

Name	Type	Description
diskEncryptionKey	KeyVault Secret Reference	Specifies the location of the disk encryption key, which is a Key Vault Secret.
enabled	boolean	Specifies whether disk encryption should be enabled on the virtual machine.
keyEncryptionKey	KeyVaultKey Reference	Specifies the location of the key encryption key in Key Vault.

DiskInstanceView

Object

The instance view of the disk.

Expand table

Name	Type	Description
encryptionSettings	Disk Encryption Settings[]	Specifies the encryption settings for the OS Disk. Minimum api-version: 2015-06-15
name	string	The disk name.
statuses	InstanceView Status[]	The resource status information.
storageAlignmentStatus	Storage Alignment Status	Specifies the storage alignment status for the disk.

DomainNameLabelScopeTypes

Enumeration

The Domain name label scope.The concatenation of the hashed domain name label that generated according to the policy from domain name label scope and vm index will be the domain name labels of the PublicIPAddress resources that will be created

Expand table

Value	Description
TenantReuse	
SubscriptionReuse	
ResourceGroupReuse	
NoReuse	

EncryptionIdentity

Object

Specifies the Managed Identity used by ADE to get access token for keyvault operations.

Expand table

Name	Type	Description
userAssignedIdentityResourceId	string	Specifies ARM Resource ID of one of the user identities associated with the VM.

EventGridAndResourceGraph

Object

Specifies eventGridAndResourceGraph related Scheduled Event related configurations.

Expand table

Name	Type	Description
enable	boolean	Specifies if event grid and resource graph is enabled for Scheduled event related configurations.
scheduledEventsApiVersion	string	Specifies the api-version to determine which Scheduled Events configuration schema version will be delivered.

ExtendedLocation

Object

The complex type of the extended location.

Expand table

Name	Type	Description
name	string	The name of the extended location.
type	Extended Location Types	The type of the extended location.

ExtendedLocationTypes

Enumeration

The type of the extended location.

Expand table

Value	Description
EdgeZone	

HardwareProfile

Object

Specifies the hardware settings for the virtual machine.

Expand table

Name	Type	Description
vmSize	Virtual MachineSize Types	Specifies the size of the virtual machine. The enum data type is currently deprecated and will be removed by December 23rd 2023. The recommended way to get the list of available sizes is using these APIs: List all available virtual machine sizes in an availability set, List all available virtual machine sizes in a region, List all available virtual machine sizes for resizing. For more information about virtual machine sizes, see Sizes for virtual machines. The available VM sizes depend on region and availability set.
vmSizeProperties	VMSize Properties	Specifies the properties for customizing the size of the virtual machine. Minimum api-version: 2021-07-01. This feature is still in preview mode and is not supported for VirtualMachineScaleSet. Please follow the instructions in VM Customization for more details.

HostEndpointSettings

Object

Specifies particular host endpoint settings.

Expand table

Name	Type	Description
inVMAccessControlProfileReferenceId	string	Specifies the InVMAccessControlProfileVersion resource id in the format of /subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Microsoft.
mode	Modes	Specifies the execution mode. In Audit mode, the system acts as if it is enforcing the access c deny any requests to host endpoints. In Enforce mode, the system will enforce the access cor

HyperVGenerationType

Enumeration

Specifies the HyperVGeneration Type associated with a resource

Expand table

Value	Description
V1	
V2	

ImageReference

Object

Specifies information about the image to use. You can specify information about platform images, marketplace images, or virtual machine images. This element is required when you want to use a platform image, marketplace image, or virtual machine image, but is not used in other creation operations. NOTE: Image reference publisher and offer can only be set when you create the scale set.

Expand table

Name	Type	Description
communityGalleryImageId	string	Specified the community gallery image unique id for vm deployment. This can be fetched from communi
exactVersion	string	Specifies in decimal numbers, the version of platform image or marketplace image used to create the virt from 'version', only if the value specified in 'version' field is 'latest'.
id	string	Resource Id
offer	string	Specifies the offer of the platform image or marketplace image used to create the virtual machine.
publisher	string	The image publisher.
sharedGalleryImageId	string	Specified the shared gallery image unique id for vm deployment. This can be fetched from shared gallery
sku	string	The image SKU.
version	string	Specifies the version of the platform image or marketplace image used to create the virtual machine. The 'latest'. Major, Minor, and Build are decimal numbers. Specify 'latest' to use the latest version of an image 'latest', the VM image will not automatically update after deploy time even if a new version becomes avai gallery image deployment, gallery image should always use 'id' field for deployment, to use 'latest' versio '/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/ga in the 'id' field without version input.

InnerError

Object

Inner error details.

Expand table

Name	Type	Description
errordetail	string	The internal error message or exception dump.
exceptiontype	string	The exception type.

InstanceViewStatus

Object

Instance view status.

Expand table

Name	Type	Description
code	string	The status code.
displayStatus	string	The short localizable label for the status.

level	StatusLevel Types	The level code.
message	string	The detailed status message, including for alerts and error messages.
time	string (date-time)	The time of the status.

IPVersions

Enumeration

Available from Api-Version 2017-03-30 onwards, it represents whether the specific ipconfiguration is IPv4 or IPv6. Default is taken as IPv4. Possible values are: 'IPv4' and 'IPv6'.

Expand table

Value	Description
IPv4	
IPv6	

KeyVaultKeyReference

Object

Describes a reference to Key Vault Key

Expand table

Name	Type	Description
keyUrl	string	The URL referencing a key encryption key in Key Vault.
sourceVault	SubResource	The relative URL of the Key Vault containing the key.

KeyVaultSecretReference

Object

Describes a reference to Key Vault Secret

Expand table

Name	Type	Description
secretUrl	string	The URL referencing a secret in a Key Vault.
sourceVault	SubResource	The relative URL of the Key Vault containing the secret.

LastPatchInstallationSummary

Object

Describes the properties of the last installed patch summary.

Expand table

Name	Type	Description
error	ApiError	The errors that were encountered during execution of the operation. The details array contains the list of them.
excludedPatchCount	integer (int32)	The number of all available patches but excluded explicitly by a customer-specified exclusion list match.
failedPatchCount	integer (int32)	The count of patches that failed installation.
installationActivityId	string	The activity ID of the operation that produced this result. It is used to correlate across CRP and extension logs.

installedPatchCount	integer (int32)	The count of patches that successfully installed.
lastModifiedTime	string (date-time)	The UTC timestamp when the operation began.
maintenanceWindowExceeded	boolean	Describes whether the operation ran out of time before it completed all its intended actions
notSelectedPatchCount	integer (int32)	The number of all available patches but not going to be installed because it didn't match a classification or inclusion list entry.
pendingPatchCount	integer (int32)	The number of all available patches expected to be installed over the course of the patch installation operation.
startTime	string (date-time)	The UTC timestamp when the operation began.
status	Patch Operation Status	The overall success or failure status of the operation. It remains "InProgress" until the operation completes. At that point it will become "Unknown", "Failed", "Succeeded", or "CompletedWithWarnings."

LinuxConfiguration

Object

Specifies the Linux operating system settings on the virtual machine. For a list of supported Linux distributions, see Linux on Azure-Endorsed Distributions.

Expand table

Name	Type	Description
disablePasswordAuthentication	boolean	Specifies whether password authentication should be disabled.
enableVMAgentPlatformUpdates	boolean	Indicates whether VMAgent Platform Updates is enabled for the Linux virtual machine. Default value is false.
patchSettings	LinuxPatch Settings	[Preview Feature] Specifies settings related to VM Guest Patching on Linux.
provisionVMAgent	boolean	Indicates whether virtual machine agent should be provisioned on the virtual machine. When this property is not specified in the request body, default behavior is to set it to true. This will ensure that VM Agent is installed on the VM so that extensions can be added to the VM later.
ssh	Ssh Configuration	Specifies the ssh key configuration for a Linux OS.

LinuxPatchAssessmentMode

Enumeration

Specifies the mode of VM Guest Patch Assessment for the IaaS virtual machine.

Possible values are:

ImageDefault - You control the timing of patch assessments on a virtual machine.

AutomaticByPlatform - The platform will trigger periodic patch assessments. The property provisionVMAgent must be true.

Expand table

Value	Description
ImageDefault	
AutomaticByPlatform	

LinuxPatchSettings

Object

Specifies settings related to VM Guest Patching on Linux.

Expand table

Name	Type	Description
assessmentMode	LinuxPatchAssessmentMode	Specifies the mode of VM Guest Patch Assessment for the IaaS virtual machine. Possible values are: **ImageDefault** - You control the timing of patch assessments on a virtual machine. **AutomaticByPlatform** - The platform will trigger periodic patch assessments. The property provisionVMAgent must be true.
automaticByPlatformSettings	LinuxVMGuestPatchAutomaticByPlatformSettings	Specifies additional settings for patch mode AutomaticByPlatform in VM Guest Patching on Linux.
patchMode	LinuxVMGuestPatchMode	Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible. Possible values are: **ImageDefault** - The virtual machine's default patching configuration is used. **AutomaticByPlatform** - The virtual machine will be automatically updated by the platform. The property provisionVMAgent must be true

LinuxVMGuestPatchAutomaticByPlatformRebootSetting

Enumeration

Specifies the reboot setting for all AutomaticByPlatform patch installation operations.

Expand table

Value	Description
Unknown	
IfRequired	
Never	
Always	

LinuxVMGuestPatchAutomaticByPlatformSettings

Object

Specifies additional settings to be applied when patch mode AutomaticByPlatform is selected in Linux patch settings.

Expand table

Name	Type	Description
bypassPlatformSafetyChecksOnUserSchedule	boolean	Enables customer to schedule patching without accidental upgrades
rebootSetting	Linux VMGuest Patch Automatic ByPlatform Reboot Setting	Specifies the reboot setting for all AutomaticByPlatform patch installation operations.

LinuxVMGuestPatchMode

Enumeration

Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible.

Possible values are:

ImageDefault - The virtual machine's default patching configuration is used.

AutomaticByPlatform - The virtual machine will be automatically updated by the platform. The property provisionVMAgent must be true

Expand table

Value	Description
ImageDefault	
AutomaticByPlatform	

MaintenanceOperationResultCodeTypes

Enumeration

The Last Maintenance Operation Result Code.

Expand table

Value	Description
None	
RetryLater	
MaintenanceAborted	
MaintenanceCompleted	

MaintenanceRedeployStatus

Object

Maintenance Operation Status.

Expand table

Name	Type	Description

isCustomerInitiatedMaintenanceAllowed	boolean	True, if customer is allowed to perform Maintenance.
lastOperationMessage	string	Message returned for the last Maintenance Operation.
lastOperationResultCode	MaintenanceOperationResultCodeTypes	The Last Maintenance Operation Result Code.
maintenanceWindowEndTime	string (date-time)	End Time for the Maintenance Window.
maintenanceWindowStartTime	string (date-time)	Start Time for the Maintenance Window.
preMaintenanceWindowEndTime	string (date-time)	End Time for the Pre Maintenance Window.
preMaintenanceWindowStartTime	string (date-time)	Start Time for the Pre Maintenance Window.

ManagedDiskParameters

Object

The parameters of a managed disk.

Expand table

Name	Type	Description
diskEncryptionSet	DiskEncryptionSetParameters	Specifies the customer managed disk encryption set resource id for the managed disk.
id	string	Resource Id
securityProfile	VMDiskSecurityProfile	Specifies the security profile for the managed disk.
storageAccountType	StorageAccountTypes	Specifies the storage account type for the managed disk. NOTE: UltraSSD_LRS can only be used with data disks, it cannot be used with OS Disk.

Mode

Enumeration

Specifies the mode that ProxyAgent will execute on if the feature is enabled. ProxyAgent will start to audit or monitor but not enforce access control over requests to host endpoints in Audit mode, while in Enforce mode it will enforce access control. The default value is Enforce mode.

Expand table

Value	Description
Audit	
Enforce	

Modes

Enumeration

Specifies the execution mode. In Audit mode, the system acts as if it is enforcing the access control policy, including emitting access denial entries in the logs but it does not actually deny any requests to host endpoints. In Enforce mode, the system will enforce the access control and it is the recommended mode of operation.

Expand table

Value	Description
Audit	
Enforce	
Disabled	

NetworkApiVersion

Enumeration

specifies the Microsoft.Network API version used when creating networking resources in the Network Interface Configurations

Expand table

Value	Description
2020-11-01	
2022-11-01	

NetworkInterfaceAuxiliaryMode

Enumeration

Specifies whether the Auxiliary mode is enabled for the Network Interface resource.

Expand table

Value	Description
None	
AcceleratedConnections	
Floating	

NetworkInterfaceAuxiliarySku

Enumeration

Specifies whether the Auxiliary sku is enabled for the Network Interface resource.

Expand table

Value	Description
None	
A1	
A2	
A4	
A8	

NetworkInterfaceReference

Object

Describes a network interface reference.

Expand table

Name	Type	Description
id	string	Resource Id

properties.deleteOption	Delete Options	Specify what happens to the network interface when the VM is deleted
properties.primary	boolean	Specifies the primary network interface in case the virtual machine has more than 1 network interface.

NetworkProfile

Object

Specifies the network interfaces or the networking configuration of the virtual machine.

Expand table

Name	Type	Description
networkApiVersion	NetworkApi Version	specifies the Microsoft.Network API version used when creating networking resources in the Network Interface Configurations
networkInterfaceConfigurations	VirtualMachine Network Interface Configuration[]	Specifies the networking configurations that will be used to create the virtual machine networking resources.
networkInterfaces	Network Interface Reference[]	Specifies the list of resource Ids for the network interfaces associated with the virtual machine.

OperatingSystemTypes

Enumeration

This property allows you to specify the type of the OS that is included in the disk if creating a VM from a custom image. Possible values are: **Windows, Linux.**

Expand table

Value	Description
Windows	
Linux	

OSDisk

Object

Specifies information about the operating system disk used by the virtual machine. For more information about disks, see About disks and VHDs for Azure virtual machines.

Expand table

Name	Type	Description
caching	CachingTypes	Specifies the caching requirements. Possible values are: **None, ReadOnly, ReadWrite.** The defaulting behavior is: **None for Standard storage. ReadOnly for Premium storage.**
createOption	DiskCreate OptionTypes	Specifies how the virtual machine disk should be created. Possible values are **Attach:** This value is used when you are using a specialized disk to create the virtual machine. **FromImage:** This value is used when you are using an image to create the virtual machine. If you are using a platform image, you should also use the imageReference element described above. If you are using a marketplace image, you should also use the plan element previously described.
deleteOption	DiskDelete OptionTypes	Specifies whether OS Disk should be deleted or detached upon VM deletion. Possible values are: **Delete.** If this value

		is used, the OS disk is deleted when VM is deleted. **Detach.** If this value is used, the os disk is retained after VM is deleted. The default value is set to **Detach**. For an ephemeral OS Disk, the default value is set to **Delete**. The user cannot change the delete option for an ephemeral OS Disk.
diffDiskSettings	DiffDisk Settings	Specifies the ephemeral Disk Settings for the operating system disk used by the virtual machine.
diskSizeGB	integer (int32)	Specifies the size of an empty data disk in gigabytes. This element can be used to overwrite the size of the disk in a virtual machine image. The property 'diskSizeGB' is the number of bytes x 1024^3 for the disk and the value cannot be larger than 1023.
encryptionSettings	Disk Encryption Settings	Specifies the encryption settings for the OS Disk. Minimum api-version: 2015-06-15.
image	VirtualHard Disk	The source user image virtual hard disk. The virtual hard disk will be copied before being attached to the virtual machine. If SourceImage is provided, the destination virtual hard drive must not exist.
managedDisk	ManagedDisk Parameters	The managed disk parameters.
name	string	The disk name.
osType	Operating SystemTypes	This property allows you to specify the type of the OS that is included in the disk if creating a VM from user-image or a specialized VHD. Possible values are: **Windows, Linux.**
storageFaultDomainAlignment	StorageFault Domain Alignment Type	Specifies the storage fault domain alignment type for the disk.
vhd	VirtualHard Disk	The virtual hard disk.
writeAcceleratorEnabled	boolean	Specifies whether writeAccelerator should be enabled or disabled on the disk.

OSImageNotificationProfile

Object

Expand table

Name	Type	Description
enable	boolean	Specifies whether the OS Image Scheduled event is enabled or disabled.
notBeforeTimeout	string	Length of time a Virtual Machine being reimaged or having its OS upgraded will have to potentially approve the OS Image Scheduled Event before the event is auto approved (timed out). The configuration is specified in ISO 8601 format, and the value must be 15 minutes (PT15M)

OSProfile

Object

Specifies the operating system settings for the virtual machine. Some of the settings cannot be changed once VM is provisioned.

Expand table

Name	Type	Description

adminPassword	string (password)	Specifies the password of the administrator account. **Minimum-length (Windows):** 8 characters **Minimum-length (Linux):** 6 characters **Max-length (Windows):** 123 characters **Max-length (Linux):** 72 characters **Complexity requirements:** 3 out of 4 conditions below need to be fulfilled Has lower characters Has upper characters Has a digit Has a special character (Regex match [\W_]) **Disallowed values:** "abc@123", "P@$$w0rd", "P@ssw0rd", "P@ssword123", "Pa$$word", "pass@word1", "Password!", "Password1", "Password22", "iloveyou!" For resetting the password, see How to reset the Remote Desktop service or its login password in a Windows VM For resetting root password, see Manage users, SSH, and check or repair disks on Azure Linux VMs using the VMAccess Extension
adminUsername	string	Specifies the name of the administrator account. This property cannot be updated after the VM is created. **Windows-only restriction:** Cannot end in "." **Disallowed values:** "administrator", "admin", "user", "user1", "test", "user2", "test1", "user3", "admin1", "1", "123", "a", "actuser", "adm", "admin2", "aspnet", "backup", "console", "david", "guest", "john", "owner", "root", "server", "sql", "support", "support_388945a0", "sys", "test2", "test3", "user4", "user5". **Minimum-length (Linux):** 1 character **Max-length (Linux):** 64 characters **Max-length (Windows):** 20 characters.
allowExtensionOperations	boolean	Specifies whether extension operations should be allowed on the virtual machine. This may only be set to False when no extensions are present on the virtual machine.
computerName	string	Specifies the host OS name of the virtual machine. This name cannot be updated after the VM is created. **Max-length (Windows):** 15 characters. **Max-length (Linux):** 64 characters. For naming conventions and restrictions see Azure infrastructure services implementation guidelines.
customData	string	Specifies a base-64 encoded string of custom data. The base-64 encoded string is decoded to a binary array that is saved as a file on the Virtual Machine. The maximum length of the binary array is 65535 bytes. **Note: Do not pass any secrets or passwords in customData property.** This property cannot be updated after the VM is created. The property 'customData' is passed to the VM to be saved as a file, for more information see Custom Data on Azure VMs. For using cloud-init for your Linux VM, see Using cloud-init to customize a Linux VM during creation.
linuxConfiguration	Linux Configuration	Specifies the Linux operating system settings on the virtual machine. For a list of supported Linux distributions, see Linux on Azure-Endorsed Distributions.
requireGuestProvisionSignal	boolean	Optional property which must either be set to True or omitted.

secrets	VaultSecret Group[]	Specifies set of certificates that should be installed onto the virtual machine. To install certificates on a virtual machine it is recommended to use the Azure Key Vault virtual machine extension for Linux or the Azure Key Vault virtual machine extension for Windows.
windowsConfiguration	Windows Configuration	Specifies Windows operating system settings on the virtual machine.

PassNames

Enumeration

Expand table

Value	Description
OobeSystem	

PatchOperationStatus

Enumeration

The overall success or failure status of the operation. It remains "InProgress" until the operation completes. At that point it will become "Unknown", "Failed", "Succeeded", or "CompletedWithWarnings."

Expand table

Value	Description
Unknown	
InProgress	
Failed	
Succeeded	
CompletedWithWarnings	

PatchSettings

Object

Specifies settings related to VM Guest Patching on Windows.

Expand table

Name	Type	Description
assessmentMode	Windows Patch Assessment Mode	Specifies the mode of VM Guest patch assessment for the IaaS virtual machine. Possible values are: **ImageDefault** - You control the timing of patch assessments on a virtual machine. **AutomaticByPlatform** - The platform will trigger periodic patch assessments. The property provisionVMAgent must be true.
automaticByPlatformSettings	Windows VMGuest Patch Automatic ByPlatform Settings	Specifies additional settings for patch mode AutomaticByPlatform in VM Guest Patching on Windows.
enableHotpatching	boolean	Enables customers to patch their Azure VMs without requiring a reboot. For enableHotpatching, the

		'provisionVMAgent' must be set to true and 'patchMode' must be set to 'AutomaticByPlatform'.
patchMode	Windows VMGuest PatchMode	Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible. Possible values are: **Manual** - You control the application of patches to a virtual machine. You do this by applying patches manually inside the VM. In this mode, automatic updates are disabled; the property WindowsConfiguration.enableAutomaticUpdates must be false **AutomaticByOS** - The virtual machine will automatically be updated by the OS. The property WindowsConfiguration.enableAutomaticUpdates must be true. **AutomaticByPlatform** - the virtual machine will automatically updated by the platform. The properties provisionVMAgent and WindowsConfiguration.enableAutomaticUpdates must be true

Placement

Object

Describes the user-defined constraints for resource hardware placement.

Expand table

Name	Type	Description
excludeZones	string[]	This property supplements the 'zonePlacementPolicy' property. If 'zonePlacementPolicy' is set to 'Any'/'Auto', availability zone selected by the system must not be present in the list of availability zones passed with 'excludeZones'. If 'excludeZones' is not provided, all availability zones in region will be considered for selection.
includeZones	string[]	This property supplements the 'zonePlacementPolicy' property. If 'zonePlacementPolicy' is set to 'Any'/'Auto', availability zone selected by the system must be present in the list of availability zones passed with 'includeZones'. If 'includeZones' is not provided, all availability zones in region will be considered for selection.
zonePlacementPolicy	Zone Placement PolicyType	Specifies the policy for resource's placement in availability zone. Possible values are: **Any** (used for Virtual Machines), **Auto** (used for Virtual Machine Scale Sets) - An availability zone will be automatically picked by system as part of resource creation.

Plan

Object

Specifies information about the marketplace image used to create the virtual machine. This element is only used for marketplace images. Before you can use a marketplace image from an API, you must enable the image for programmatic use. In the Azure portal, find the marketplace image that you want to use and then click **Want to deploy programmatically, Get Started ->**. Enter any required information and then click **Save**.

Expand table

Name	Type	Description
name	string	The plan ID.
product	string	Specifies the product of the image from the marketplace. This is the same value as Offer under the imageReference element.

promotionCode	string	The promotion code.
publisher	string	The publisher ID.

ProtocolTypes

Enumeration

Specifies the protocol of WinRM listener. Possible values are: **http, https.**

Expand table

Value	Description
Http	
Https	

ProxyAgentSettings

Object

Specifies ProxyAgent settings for the virtual machine or virtual machine scale set. Minimum api-version: 2023-09-01.

Expand table

Name	Type	Description
addProxyAgentExtension	boolean	Specify whether to implicitly install the ProxyAgent Extension. This option is currently applicable only for Linux Os.
enabled	boolean	Specifies whether ProxyAgent feature should be enabled on the virtual machine or virtual machine scale set.
imds	HostEndpoint Settings	Specifies the IMDS endpoint settings while creating the virtual machine or virtual machine scale set. Minimum api-version: 2024-03-01.
keyIncarnationId	integer (int32)	Increase the value of this property allows users to reset the key used for securing communication channel between guest and host.
mode	Mode	Specifies the mode that ProxyAgent will execute on. Warning: this property has been deprecated, please specify 'mode' under particular hostendpoint setting.
wireServer	HostEndpoint Settings	Specifies the Wire Server endpoint settings while creating the virtual machine or virtual machine scale set. Minimum api-version: 2024-03-01.

PublicIPAddressSku

Object

Describes the public IP Sku. It can only be set with OrchestrationMode as Flexible.

Expand table

Name	Type	Description
name	Public IPAddressSku Name	Specify public IP sku name
tier	Public IPAddressSku Tier	Specify public IP sku tier

PublicIPAddressSkuName

Enumeration

Specify public IP sku name

Expand table

Value	Description
Basic	
Standard	

PublicIPAddressSkuTier

Enumeration

Specify public IP sku tier

Expand table

Value	Description
Regional	
Global	

PublicIPAllocationMethod

Enumeration

Specify the public IP allocation type

Expand table

Value	Description
Dynamic	
Static	

ResiliencyProfile

Object

Gets resiliency solutions enabled on the VM. This includes backup or disaster recovery solutions.

Expand table

Name	Type	Description
zoneMovement	Zone Movement	Zone movement configuration.

ResourceIdentityType

Enumeration

The type of identity used for the virtual machine. The type 'SystemAssigned, UserAssigned' includes both an implicitly created identity and a set of user assigned identities. The type 'None' will remove any identities from the virtual machine.

Expand table

Value	Description
SystemAssigned	
UserAssigned	
SystemAssigned, UserAssigned	
None	

ScheduledEventsAdditionalPublishingTargets

Object

Expand table

Name	Type	Description
eventGridAndResourceGraph	EventGridAndResourceGraph	The configuration parameters used while creating eventGridAndResourceGraph Scheduled Event setting.

ScheduledEventsPolicy

Object

Specifies Redeploy, Reboot and ScheduledEventsAdditionalPublishingTargets Scheduled Event related configurations.

Expand table

Name	Type	Description
allInstancesDown	AllInstancesDown	The configuration parameters used while creating AllInstancesDown scheduled event setting creation.
scheduledEventsAdditionalPublishingTargets	ScheduledEventsAdditionalPublishingTargets	The configuration parameters used while publishing scheduledEventsAdditionalPublishingTargets.
userInitiatedReboot	UserInitiatedReboot	The configuration parameters used while creating userInitiatedReboot scheduled event setting creation.
userInitiatedRedeploy	UserInitiatedRedeploy	The configuration parameters used while creating userInitiatedRedeploy scheduled event setting creation.

ScheduledEventsProfile

Object

Expand table

Name	Type	Description
osImageNotificationProfile	OSImageNotificationProfile	Specifies OS Image Scheduled Event related configurations.
terminateNotificationProfile	TerminateNotificationProfile	Specifies Terminate Scheduled Event related configurations.

SecurityEncryptionTypes

Enumeration

Specifies the EncryptionType of the managed disk. It is set to DiskWithVMGuestState for encryption of the managed disk along with VMGuestState blob, VMGuestStateOnly for encryption of just the VMGuestState blob, and NonPersistedTPM for not persisting firmware state in the VMGuestState blob.. **Note:** It can be set for only Confidential VMs.

Expand table

Value	Description
VMGuestStateOnly	

DiskWithVMGuestState
NonPersistedTPM

SecurityProfile

Object

Specifies the Security profile settings for the virtual machine or virtual machine scale set.

Expand table

Name	Type	Description
encryptionAtHost	boolean	This property can be used by user in the request to enable or disable the Host Encryption for the virtual machine or virtual machine scale set. This will enable the encryption for all the disks including Resource/Temp disk at host itself. The default behavior is: The Encryption at host will be disabled unless this property is set to true for the resource.
encryptionIdentity	Encryption Identity	Specifies the Managed Identity used by ADE to get access token for keyvault operations.
proxyAgentSettings	ProxyAgent Settings	Specifies ProxyAgent settings while creating the virtual machine. Minimum api-version: 2023-09-01.
securityType	SecurityTypes	Specifies the SecurityType of the virtual machine. It has to be set to any specified value to enable UefiSettings. The default behavior is: UefiSettings will not be enabled unless this property is set and is not Standard. If not specified, Standard will be returned starting api version 2025-11-01.
uefiSettings	UefiSettings	Specifies the security settings like secure boot and vTPM used while creating the virtual machine. Minimum api-version: 2020-12-01.

SecurityTypes

Enumeration

Specifies the VM securityType; UefiSettings are enabled only when set to TrustedLaunch or ConfidentialVM, and returns a Standard value starting API version 2025-11-01.

Expand table

Value	Description
Standard	Indicates a VM without UEFI features such as SecureBoot or vTPM; returned as the default value when securityType is not specified.
TrustedLaunch	
ConfidentialVM	

SettingNames

Enumeration

Specifies the name of the setting to which the content applies. Possible values are: FirstLogonCommands and AutoLogon.

Expand table

Value	Description
AutoLogon	
FirstLogonCommands	

SshConfiguration

Object

SSH configuration for Linux based VMs running on Azure

Expand table

Name	Type	Description
publicKeys	SshPublicKey[]	The list of SSH public keys used to authenticate with linux based VMs.

SshPublicKey

Object

Contains information about SSH certificate public key and the path on the Linux VM where the public key is placed.

Expand table

Name	Type	Description
keyData	string	SSH public key certificate used to authenticate with the VM through ssh. The key needs to be at least 2048-bit and in ssh-rsa format. For creating ssh keys, see [Create SSH keys on Linux and Mac for Linux VMs in Azure]https://docs.microsoft.com/azure/virtual-machines/linux/create-ssh-keys-detailed).
path	string	Specifies the full path on the created VM where ssh public key is stored. If the file already exists, the specified key is appended to the file. Example: /home/user/.ssh/authorized_keys

StatusLevelTypes

Enumeration

The level code.

Expand table

Value	Description
Info	
Warning	
Error	

StorageAccountTypes

Enumeration

Specifies the storage account type for the managed disk. Managed OS disk storage account type can only be set when you create the scale set. NOTE: UltraSSD_LRS can only be used with data disks. It cannot be used with OS Disk. Standard_LRS uses Standard HDD. StandardSSD_LRS uses Standard SSD. Premium_LRS uses Premium SSD. UltraSSD_LRS uses Ultra disk. Premium_ZRS uses Premium SSD zone redundant storage. StandardSSD_ZRS uses Standard SSD zone redundant storage. For more information regarding disks supported for Windows Virtual Machines, refer to https://docs.microsoft.com/azure/virtual-machines/windows/disks-types and, for Linux Virtual Machines, refer to https://docs.microsoft.com/azure/virtual-machines/linux/disks-types

Expand table

Value	Description
Standard_LRS	
Premium_LRS	
StandardSSD_LRS	

UltraSSD_LRS
Premium_ZRS
StandardSSD_ZRS
PremiumV2_LRS

StorageAlignmentStatus

Enumeration

Specifies the storage alignment status for the disk.

Expand table

Value	Description
Unaligned	Disk does not have Storage Fault Domain to Compute Fault Domain mapping. A single Storage Fault Domain failure may impact all VMs that reference this disk profile.
Aligned	Disk has Storage Fault Domain to Compute Fault Domain mapping. Storage Fault Domain failure is contained to VMs in a single Compute Fault Domain.

StorageFaultDomainAlignmentType

Enumeration

Specifies the storage fault domain alignment type for the disk.

Expand table

Value	Description
Aligned	Disk Storage Fault Domains are mapped to Compute Fault Domains. Deployment fails if disk does not support enough Fault Domains.
BestEffortAligned	Attempt to map Storage Fault Domains to Compute Fault Domains. Disks are unaligned if disk does not support enough Fault Domains.

StorageProfile

Object

Specifies the storage settings for the virtual machine disks.

Expand table

Name	Type	Description
alignRegionalDisksToVMZone	boolean	Specifies whether the regional disks should be aligned/moved to the VM zone. This is applicable only for VMs with placement property set. Please note that this change is irreversible. Minimum api-version: 2024-11-01.
dataDisks	DataDisk[]	Specifies the parameters that are used to add a data disk to a virtual machine. For more information about disks, see About disks and VHDs for Azure virtual machines.
diskControllerType	Disk Controller Types	Specifies the disk controller type configured for the VM. **Note:** This property will be set to the default disk controller type if not specified provided virtual machine is being created with 'hyperVGeneration' set to V2 based on the capabilities of the operating system disk and VM size from the the specified minimum api version. You need to deallocate the VM before updating its disk controller type unless you are updating the VM size in the VM configuration which implicitly deallocates and reallocates the VM. Minimum api-version: 2022-08-01.
imageReference	Image Reference	Specifies information about the image to use. You can specify information about platform images, marketplace images, or virtual machine images. This element is required

		when you want to use a platform image, marketplace image, or virtual machine image, but is not used in other creation operations.
osDisk	OSDisk	Specifies information about the operating system disk used by the virtual machine. For more information about disks, see About disks and VHDs for Azure virtual machines.

SubResource

Object

Expand table

Name	Type	Description
id	string	Resource Id

systemData

Object

Metadata pertaining to creation and last modification of the resource.

Expand table

Name	Type	Description
createdAt	string (date-time)	The timestamp of resource creation (UTC).
createdBy	string	The identity that created the resource.
createdByType	createdByType	The type of identity that created the resource.
lastModifiedAt	string (date-time)	The timestamp of resource last modification (UTC)
lastModifiedBy	string	The identity that last modified the resource.
lastModifiedByType	createdByType	The type of identity that last modified the resource.

TerminateNotificationProfile

Object

Expand table

Name	Type	Description
enable	boolean	Specifies whether the Terminate Scheduled event is enabled or disabled.
notBeforeTimeout	string	Configurable length of time a Virtual Machine being deleted will have to potentially approve the Terminate Scheduled Event before the event is auto approved (timed out). The configuration must be specified in ISO 8601 format, the default value is 5 minutes (PT5M)

UefiSettings

Object

Specifies the security settings like secure boot and vTPM used while creating the virtual machine. Minimum api-version: 2020-12-01.

Expand table

Name	Type	Description
secureBootEnabled	boolean	Specifies whether secure boot should be enabled on the virtual

		machine. Minimum api-version: 2020-12-01.
vTpmEnabled	boolean	Specifies whether vTPM should be enabled on the virtual machine. Minimum api-version: 2020-12-01.

UserInitiatedReboot

Object

Specifies Reboot related Scheduled Event related configurations.

Expand table

Name	Type	Description
automaticallyApprove	boolean	Specifies Reboot Scheduled Event related configurations.

UserInitiatedRedeploy

Object

Specifies Redeploy related Scheduled Event related configurations.

Expand table

Name	Type	Description
automaticallyApprove	boolean	Specifies Redeploy Scheduled Event related configurations.

VaultCertificate

Object

Describes a single certificate reference in a Key Vault, and where the certificate should reside on the VM.

Expand table

Name	Type	Description
certificateStore	string	For Windows VMs, specifies the certificate store on the Virtual Machine to which the certificate should be added. The specified certificate store is implicitly in the LocalMachine account. For Linux VMs, the certificate file is placed under the /var/lib/waagent directory, with the file name <UppercaseThumbprint>.crt for the X509 certificate file and <UppercaseThumbprint>.prv for private key. Both of these files are .pem formatted.

certificateUrl	string	This is the URL of a certificate that has been uploaded to Key Vault as a secret. For adding a secret to the Key Vault, see Add a key or secret to the key vault. In this case, your certificate needs to be It is the Base64 encoding of the following JSON Object which is encoded in UTF-8:
		
		{
		"data":"<Base64-encoded-certificate>",
		"dataType":"pfx",
		"password":"<pfx-file-password>"
		}
		To install certificates on a virtual machine it is recommended to use the Azure Key Vault virtual machine extension for Linux or the Azure Key Vault virtual machine extension for Windows.

VaultSecretGroup

Object

Describes a set of certificates which are all in the same Key Vault.

Expand table

Name	Type	Description

sourceVault	SubResource	The relative URL of the Key Vault containing all of the certificates in VaultCertificates.
vaultCertificates	Vault Certificate[]	The list of key vault references in SourceVault which contain certificates.

VirtualHardDisk

Object

Describes the uri of a disk.

Expand table

Name	Type	Description
uri	string	Specifies the virtual hard disk's uri.

VirtualMachine

Object

Describes a Virtual Machine.

Expand table

Name	Type	Description
etag	string	Etag is property returned in Create/Update/Get response of the VM, so that customer ca
extendedLocation	Extended Location	The extended location of the Virtual Machine.
id	string	Fully qualified resource ID for the resource. Ex - /subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/{resour
identity	Virtual Machine Identity	The identity of the virtual machine, if configured.
location	string	The geo-location where the resource lives
managedBy	string	ManagedBy is set to Virtual Machine Scale Set(VMSS) flex ARM resourceID, if the VM is p internal resource group delete optimization.
name	string	The name of the resource
placement	Placement	Placement section specifies the user-defined constraints for virtual machine hardware pla provisioned. Minimum api-version: 2024-11-01.
plan	Plan	Specifies information about the marketplace image used to create the virtual machine. Tl you can use a marketplace image from an API, you must enable the image for programm image that you want to use and then click **Want to deploy programmatically, Get Starte Save**.
properties.additionalCapabilities	Additional Capabilities	Specifies additional capabilities enabled or disabled on the virtual machine.
properties.applicationProfile	Application Profile	Specifies the gallery applications that should be made available to the VM/VMSS.
properties.availabilitySet	Sub Resource	Specifies information about the availability set that the virtual machine should be assigne set are allocated to different nodes to maximize availability. For more information about information on Azure planned maintenance, see Maintenance and updates for Virtual Ma availability set at creation time. The availability set to which the VM is being added shoul set resource. An existing VM cannot be added to an availability set. This property cannot properties.virtualMachineScaleSet reference.
properties.billingProfile	Billing Profile	Specifies the billing related details of a Azure Spot virtual machine. Minimum api-version
properties.capacityReservation	Capacity Reservation Profile	Specifies information about the capacity reservation that is used to allocate virtual machi

properties.diagnosticsProfile	Diagnostics Profile	Specifies the boot diagnostic settings state. Minimum api-version: 2015-06-15.
properties.evictionPolicy	Virtual Machine Eviction PolicyTypes	Specifies the eviction policy for the Azure Spot virtual machine and Azure Spot scale set. 'Delete' are supported and the minimum api-version is 2019-03-01. For Azure Spot scale the minimum api-version is 2017-10-30-preview.
properties.extensionsTimeBudget	string	Specifies the time alloted for all extensions to start. The time duration should be betweer specified in ISO 8601 format. The default value is 90 minutes (PT1H30M). Minimum api-v
properties.hardwareProfile	Hardware Profile	Specifies the hardware settings for the virtual machine.
properties.host	Sub Resource	Specifies information about the dedicated host that the virtual machine resides in. Minim
properties.hostGroup	Sub Resource	Specifies information about the dedicated host group that the virtual machine resides in. properties. Minimum api-version: 2020-06-01.
properties.instanceView	Virtual Machine Instance View	The virtual machine instance view.
properties.licenseType	string	Specifies that the image or disk that is being used was licensed on-premises. Possible values for Windows Server operating system are: Windows_Client Windows_Server Possible values for Linux Server operating system are: RHEL_BYOS (for RHEL) SLES_BYOS (for SUSE) For more information, see Azure Hybrid Use Benefit for Windows Server Azure Hybrid Use Benefit for Linux Server Minimum api-version: 2015-06-15
properties.networkProfile	Network Profile	Specifies the network interfaces of the virtual machine.
properties.osProfile	OSProfile	Specifies the operating system settings used while creating the virtual machine. Some of provisioned.
properties.platformFaultDomain	integer (int32)	Specifies the scale set logical fault domain into which the Virtual Machine will be created assigned to a fault domain that best maintains balance across available fault domains. Th property of this Virtual Machine is set. The Virtual Machine Scale Set that is referenced, n This property cannot be updated once the Virtual Machine is created. Fault domain assig View. Minimum api-version: 2020-12-01.
properties.priority	Virtual Machine Priority Types	Specifies the priority for the virtual machine. Minimum api-version: 2019-03-01
properties.provisioningState	string	The provisioning state, which only appears in the response.
properties.proximityPlacementGroup	Sub Resource	Specifies information about the proximity placement group that the virtual machine shou
properties.resiliencyProfile	Resiliency Profile	Resiliency profile for the virtual machine.
properties.scheduledEventsPolicy	Scheduled Events Policy	Specifies Redeploy, Reboot and ScheduledEventsAdditionalPublishingTargets Scheduled
properties.scheduledEventsProfile	Scheduled Events Profile	Specifies Scheduled Event related configurations.

properties.securityProfile	Security Profile	Specifies the Security related profile settings for the virtual machine.
properties.storageProfile	Storage Profile	Specifies the storage settings for the virtual machine disks.
properties.timeCreated	string (date-time)	Specifies the time at which the Virtual Machine resource was created. Minimum api-versi
properties.userData	string	UserData for the VM, which must be base-64 encoded. Customer should not pass any se
properties.virtualMachineScaleSet	Sub Resource	Specifies information about the virtual machine scale set that the virtual machine should virtual machine scale set are allocated to different nodes to maximize availability. Current at creation time. An existing VM cannot be added to a virtual machine scale set. This pro properties.availabilitySet reference. Minimum api-version: 2019-03-01.
properties.vmId	string	Specifies the VM unique ID which is a 128-bits identifier that is encoded and stored in all platform BIOS commands.
resources	Virtual Machine Extension[]	The virtual machine child extension resources.
systemData	system Data	Azure Resource Manager metadata containing createdBy and modifiedBy information.
tags	object	Resource tags.
type	string	The type of the resource. E.g. "Microsoft.Compute/virtualMachines" or "Microsoft.Storag
zones	string[]	The availability zones.

VirtualMachineAgentInstanceView

Object

The instance view of the VM Agent running on the virtual machine.

Expand table

Name	Type	Description
extensionHandlers	Virtual Machine Extension Handler Instance View[]	The virtual machine extension handler instance view.
statuses	InstanceView Status[]	The resource status information.
vmAgentVersion	string	The VM Agent full version.

VirtualMachineEvictionPolicyTypes

Enumeration

Specifies the eviction policy for the Azure Spot VM/VMSS

Expand table

Value	Description
Deallocate	
Delete	

VirtualMachineExtension

Object

Describes a Virtual Machine Extension.

Expand table

Name	Type	Description
id	string	Fully qualified resource ID for the resource. Ex - /subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/{res
location	string	The geo-location where the resource lives
name	string	The name of the resource
properties.autoUpgradeMinorVersion	boolean	Indicates whether the extension should use a newer minor version if one is available a extension will not upgrade minor versions unless redeployed, even with this property
properties.enableAutomaticUpgrade	boolean	Indicates whether the extension should be automatically upgraded by the platform if
properties.forceUpdateTag	string	How the extension handler should be forced to update even if the extension configur
properties.instanceView	Virtual Machine Extension Instance View	The virtual machine extension instance view.
properties.protectedSettings		The extension can contain either protectedSettings or protectedSettingsFromKeyVau
properties.protectedSettingsFromKeyVault	KeyVault Secret Reference	The extensions protected settings that are passed by reference, and consumed from
properties.provisionAfterExtensions	string[]	Collection of extension names after which this extension needs to be provisioned.
properties.provisioningState	string	The provisioning state, which only appears in the response.
properties.publisher	string	The name of the extension handler publisher.
properties.settings		Json formatted public settings for the extension.
properties.suppressFailures	boolean	Indicates whether failures stemming from the extension will be suppressed (Operatio suppressed regardless of this value). The default is false.
properties.type	string	Specifies the type of the extension; an example is "CustomScriptExtension".
properties.typeHandlerVersion	string	Specifies the version of the script handler.
systemData	system Data	Azure Resource Manager metadata containing createdBy and modifiedBy informatior
tags	object	Resource tags.
type	string	The type of the resource. E.g. "Microsoft.Compute/virtualMachines" or "Microsoft.Sto

VirtualMachineExtensionHandlerInstanceView

Object

The instance view of a virtual machine extension handler.

Expand table

Name	Type	Description
status	InstanceView Status	The extension handler status.
type	string	Specifies the type of the extension; an example is "CustomScriptExtension".
typeHandlerVersion	string	Specifies the version of the script handler.

VirtualMachineExtensionInstanceView

Object

The instance view of a virtual machine extension.

Expand table

Name	Type	Description
name	string	The virtual machine extension name.
statuses	InstanceView Status[]	The resource status information.
substatuses	InstanceView Status[]	The resource status information.
type	string	Specifies the type of the extension; an example is "CustomScriptExtension".
typeHandlerVersion	string	Specifies the version of the script handler.

VirtualMachineHealthStatus

Object

The health status of the VM.

Expand table

Name	Type	Description
status	InstanceView Status	The health status information for the VM.

VirtualMachineIdentity

Object

Identity for the virtual machine.

Expand table

Name	Type	Description
principalId	string	The principal id of virtual machine identity. This property will only be provided for a system assigned ider
tenantId	string	The tenant id associated with the virtual machine. This property will only be provided for a system assign
type	Resource Identity Type	The type of identity used for the virtual machine. The type 'SystemAssigned, UserAssigned' includes both user assigned identities. The type 'None' will remove any identities from the virtual machine.
userAssignedIdentities	<string, Common. User Assigned Identities Value>	The list of user identities associated with the Virtual Machine. The user identity dictionary key references '/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.ManagedIde

VirtualMachineInstanceView

Object

The instance view of a virtual machine.

Expand table

Name	Type	Description
assignedHost	string	Resource id of the dedicated host, on which the virtual machine is allocated through automatic placement, when the virtual machine is associated with a dedicated host group that has automatic placement enabled. Minimum api-version: 2020-06-01.
bootDiagnostics	Boot Diagnostics	Boot Diagnostics is a debugging feature which allows you to view Console Output and Screenshot to diagnose VM status.

	InstanceView	You can easily view the output of your console log. Azure also enables you to see a screenshot of the VM from the hypervisor.
computerName	string	The computer name assigned to the virtual machine.
disks	DiskInstanceView[]	The virtual machine disk information.
extensions	VirtualMachineExtensionInstanceView[]	The extensions information.
hyperVGeneration	HyperVGenerationType	Specifies the HyperVGeneration Type associated with a resource
isVMInStandbyPool	boolean	[Preview Feature] Specifies whether the VM is currently in or out of the Standby Pool.
maintenanceRedeployStatus	MaintenanceRedeployStatus	The Maintenance Operation status on the virtual machine.
osName	string	The Operating System running on the virtual machine.
osVersion	string	The version of Operating System running on the virtual machine.
patchStatus	VirtualMachinePatchStatus	[Preview Feature] The status of virtual machine patch operations.
platformFaultDomain	integer (int32)	Specifies the fault domain of the virtual machine.
platformUpdateDomain	integer (int32)	Specifies the update domain of the virtual machine.
rdpThumbPrint	string	The Remote desktop certificate thumbprint.
statuses	InstanceViewStatus[]	The resource status information.
vmAgent	VirtualMachineAgentInstanceView	The VM Agent running on the virtual machine.
vmHealth	VirtualMachineHealthStatus	The health status for the VM.

VirtualMachineIpTag

Object

Contains the IP tag associated with the public IP address.

Expand table

Name	Type	Description
ipTagType	string	IP tag type. Example: FirstPartyUsage.
tag	string	IP tag associated with the public IP. Example: SQL, Storage etc.

VirtualMachineNetworkInterfaceConfiguration

Object

Describes a virtual machine network interface configurations.

Expand table

Name	Type	Description
name	string	The network interface configuration name.
properties.auxiliaryMode	Network Interface AuxiliaryMode	Specifies whether the Auxiliary mode is enabled for the Network Interface resource.
properties.auxiliarySku	Network Interface AuxiliarySku	Specifies whether the Auxiliary sku is enabled for the Network Interface resource.
properties.deleteOption	DeleteOptions	Specify what happens to the network interface when the VM is deleted
properties.disableTcpStateTracking	boolean	Specifies whether the network interface is disabled for tcp state tracking.
properties.dnsSettings	VirtualMachine Network InterfaceDns Settings Configuration	The dns settings to be applied on the network interfaces.
properties.dscpConfiguration	SubResource	
properties.enableAcceleratedNetworking	boolean	Specifies whether the network interface is accelerated networking-enabled.
properties.enableFpga	boolean	Specifies whether the network interface is FPGA networking-enabled.
properties.enableIPForwarding	boolean	Whether IP forwarding enabled on this NIC.
properties.ipConfigurations	VirtualMachine Network Interface IPConfiguration[]	Specifies the IP configurations of the network interface.
properties.networkSecurityGroup	SubResource	The network security group.
properties.primary	boolean	Specifies the primary network interface in case the virtual machine has more than 1 network interface.
tags	object	Resource tags applied to the networkInterface address created by this NetworkInterfaceConfiguration

VirtualMachineNetworkInterfaceDns SettingsConfiguration

Object

Describes a virtual machines network configuration's DNS settings.

Expand table

Name	Type	Description
dnsServers	string[]	List of DNS servers IP addresses

VirtualMachineNetworkInterfaceIPConfiguration

Object

Describes a virtual machine network profile's IP configuration.

Expand table

Name	Type	Description

name	string	The IP configuration name.
properties.applicationGatewayBackendAddressPools	Sub Resource[]	Specifies an array of references to backend address pools of application gateways. A virtual machine can reference backend address pools of multiple application gateways. Multiple virtual machines cannot use the same application gateway.
properties.applicationSecurityGroups	Sub Resource[]	Specifies an array of references to application security group.
properties.loadBalancerBackendAddressPools	Sub Resource[]	Specifies an array of references to backend address pools of load balancers. A virtual machine can reference backend address pools of one public and one internal load balancer. [Multiple virtual machines cannot use the same basic sku load balancer].
properties.primary	boolean	Specifies the primary network interface in case the virtual machine has more than 1 network interface.
properties.privateIPAddressVersion	IPVersions	Available from Api-Version 2017-03-30 onwards, it represents whether the specific ipconfiguration is IPv4 or IPv6. Default is taken as IPv4. Possible values are: 'IPv4' and 'IPv6'.
properties.publicIPAddressConfiguration	Virtual Machine Public IPAddress Configuration	The publicIPAddressConfiguration.
properties.subnet	SubResource	Specifies the identifier of the subnet.

VirtualMachinePatchStatus

Object

The status of virtual machine patch operations.

Expand table

Name	Type	Description
availablePatchSummary	Available Patch Summary	The available patch summary of the latest assessment operation for the virtual machine.
configurationStatuses	InstanceView Status[]	The enablement status of the specified patchMode
lastPatchInstallationSummary	LastPatch Installation Summary	The installation summary of the latest installation operation for the virtual machine.

VirtualMachinePriorityTypes

Enumeration

Specifies the priority for a standalone virtual machine or the virtual machines in the scale set. 'Low' enum will be deprecated in the future, please use 'Spot' as the enum to deploy Azure Spot VM/VMSS.

Expand table

Value	Description

Regular
Low
Spot

VirtualMachinePublicIPAddressConfiguration

Object

Describes a virtual machines IP Configuration's PublicIPAddress configuration

Expand table

Name	Type	Description
name	string	The publicIP address configuration name.
properties.deleteOption	Delete Options	Specify what happens to the public IP address when the VM is deleted
properties.dnsSettings	Virtual Machine Public IPAddressDns Settings Configuration	The dns settings to be applied on the publicIP addresses .
properties.idleTimeoutInMinutes	integer (int32)	The idle timeout of the public IP address.
properties.ipTags	Virtual Machine IpTag[]	The list of IP tags associated with the public IP address.
properties.publicIPAddressVersion	IPVersions	Available from Api-Version 2019-07-01 onwards, it represents whether the specific ipconfiguration is IPv4 or IPv6. Default is taken as IPv4. Possible values are: 'IPv4' and 'IPv6'.
properties.publicIPAllocationMethod	Public IPAllocation Method	Specify the public IP allocation type
properties.publicIPPrefix	SubResource	The PublicIPPrefix from which to allocate publicIP addresses.
sku	Public IPAddressSku	Describes the public IP Sku. It can only be set with OrchestrationMode as Flexible.
tags	object	Resource tags applied to the publicIP address created by this PublicIPAddressConfiguration

VirtualMachinePublicIPAddressDns SettingsConfiguration

Object

Describes a virtual machines network configuration's DNS settings.

Expand table

Name	Type	Description
domainNameLabel	string	The Domain name label prefix of the PublicIPAddress resources that will be created. The generated name label is the concatenation of the domain name label and vm network profile unique ID.
domainNameLabelScope	DomainName LabelScope Types	The Domain name label scope of the PublicIPAddress resources that will be created. The generated name label is the concatenation of the hashed domain name label with policy according to the domain name label scope and vm network profile unique ID.

VirtualMachineSizeTypes

Enumeration

Specifies the size of the virtual machine. The enum data type is currently deprecated and will be removed by December 23rd 2023. The recommended way to get the list of available sizes is using these APIs: List all available virtual machine sizes in an availability set, List all available virtual machine sizes in a region, List all available virtual machine sizes for resizing. For more information about virtual machine sizes, see Sizes for virtual machines. The available VM sizes depend on region and availability set.

Expand table

Value	Description
Basic_A0	
Basic_A1	
Basic_A2	
Basic_A3	
Basic_A4	
Standard_A0	
Standard_A1	
Standard_A2	
Standard_A3	
Standard_A4	
Standard_A5	
Standard_A6	
Standard_A7	
Standard_A8	
Standard_A9	
Standard_A10	
Standard_A11	
Standard_A1_v2	
Standard_A2_v2	
Standard_A4_v2	
Standard_A8_v2	
Standard_A2m_v2	
Standard_A4m_v2	
Standard_A8m_v2	
Standard_B1s	
Standard_B1ms	
Standard_B2s	
Standard_B2ms	
Standard_B4ms	
Standard_B8ms	
Standard_D1	
Standard_D2	

Standard_D3
Standard_D4
Standard_D11
Standard_D12
Standard_D13
Standard_D14
Standard_D1_v2
Standard_D2_v2
Standard_D3_v2
Standard_D4_v2
Standard_D5_v2
Standard_D2_v3
Standard_D4_v3
Standard_D8_v3
Standard_D16_v3
Standard_D32_v3
Standard_D64_v3
Standard_D2s_v3
Standard_D4s_v3
Standard_D8s_v3
Standard_D16s_v3
Standard_D32s_v3
Standard_D64s_v3
Standard_D11_v2
Standard_D12_v2
Standard_D13_v2
Standard_D14_v2
Standard_D15_v2
Standard_DS1
Standard_DS2
Standard_DS3
Standard_DS4
Standard_DS11
Standard_DS12
Standard_DS13
Standard_DS14
Standard_DS1_v2
Standard_DS2_v2
Standard_DS3_v2
Standard_DS4_v2
Standard_DS5_v2

Standard_DS11_v2
Standard_DS12_v2
Standard_DS13_v2
Standard_DS14_v2
Standard_DS15_v2
Standard_DS13-4_v2
Standard_DS13-2_v2
Standard_DS14-8_v2
Standard_DS14-4_v2
Standard_E2_v3
Standard_E4_v3
Standard_E8_v3
Standard_E16_v3
Standard_E32_v3
Standard_E64_v3
Standard_E2s_v3
Standard_E4s_v3
Standard_E8s_v3
Standard_E16s_v3
Standard_E32s_v3
Standard_E64s_v3
Standard_E32-16_v3
Standard_E32-8s_v3
Standard_E64-32s_v3
Standard_E64-16s_v3
Standard_F1
Standard_F2
Standard_F4
Standard_F8
Standard_F16
Standard_F1s
Standard_F2s
Standard_F4s
Standard_F8s
Standard_F16s
Standard_F2s_v2
Standard_F4s_v2
Standard_F8s_v2

Standard_F16s_v2
Standard_F32s_v2
Standard_F64s_v2
Standard_F72s_v2
Standard_G1
Standard_G2
Standard_G3
Standard_G4
Standard_G5
Standard_GS1
Standard_GS2
Standard_GS3
Standard_GS4
Standard_GS5
Standard_GS4-8
Standard_GS4-4
Standard_GS5-16
Standard_GS5-8
Standard_H8
Standard_H16
Standard_H8m
Standard_H16m
Standard_H16r
Standard_H16mr
Standard_L4s
Standard_L8s
Standard_L16s
Standard_L32s
Standard_M64s
Standard_M64ms
Standard_M128s
Standard_M128ms
Standard_M64-32ms
Standard_M64-16ms
Standard_M128-64ms
Standard_M128-32ms
Standard_NC6
Standard_NC12
Standard_NC24

Standard_NC24r
Standard_NC6s_v2
Standard_NC12s_v2
Standard_NC24s_v2
Standard_NC24rs_v2
Standard_NC6s_v3
Standard_NC12s_v3
Standard_NC24s_v3
Standard_NC24rs_v3
Standard_ND6s
Standard_ND12s
Standard_ND24s
Standard_ND24rs
Standard_NV6
Standard_NV12
Standard_NV24

VMDiskSecurityProfile

Object

Specifies the security profile settings for the managed disk. **Note:** It can only be set for Confidential VMs.

Expand table

Name	Type	Description
diskEncryptionSet	Disk EncryptionSet Parameters	Specifies the customer managed disk encryption set resource id for the managed disk that is used for Customer Managed Key encrypted ConfidentialVM OS Disk and VMGuest blob.
securityEncryptionType	Security Encryption Types	Specifies the EncryptionType of the managed disk. It is set to DiskWithVMGuestState for encryption of the managed disk along with VMGuestState blob, VMGuestStateOnly for encryption of just the VMGuestState blob, and NonPersistedTPM for not persisting firmware state in the VMGuestState blob.. **Note:** It can be set for only Confidential VMs.

VMGalleryApplication

Object

Specifies the required information to reference a compute gallery application version

Expand table

Name	Type	Description
configurationReference	string	Optional, Specifies the uri to an azure blob that will replace the default configuration for the pac
enableAutomaticUpgrade	boolean	If set to true, when a new Gallery Application version is available in PIR/SIG, it will be automatica
order	integer (int32)	Optional, Specifies the order in which the packages have to be installed
packageReferenceId	string	Specifies the GalleryApplicationVersion resource id on the form of /subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Microsoft.Com
tags	string	Optional, Specifies a passthrough value for more generic context.

treatFailureAsDeploymentFailure	boolean	Optional, If true, any failure for any operation in the VmApplication will fail the deployment

VMSizeProperties

Object

Specifies VM Size Property settings on the virtual machine.

Expand table

Name	Type	Description
vCPUsAvailable	integer (int32)	Specifies the number of vCPUs available for the VM. When this property is not specified in the request body the default behavior is to set it to the value of vCPUs available for that VM size exposed in api response of List all available virtual machine sizes in a region.
vCPUsPerCore	integer (int32)	Specifies the vCPU to physical core ratio. When this property is not specified in the request body the default behavior is set to the value of vCPUsPerCore for the VM Size exposed in api response of List all available virtual machine sizes in a region. **Setting this property to 1 also means that hyper-threading is disabled.**

WindowsConfiguration

Object

Specifies Windows operating system settings on the virtual machine.

Expand table

Name	Type	Description
additionalUnattendContent	Additional Unattend Content[]	Specifies additional base-64 encoded XML formatted information that can be included in the Unattend.xml file, which is used by Windows Setup.
enableAutomaticUpdates	boolean	Indicates whether Automatic Updates is enabled for the Windows virtual machine. Default value is true. For virtual machine scale sets, this property can be updated and updates will take effect on OS reprovisioning.
enableVMAgentPlatformUpdates	boolean	Indicates whether VMAgent Platform Updates are enabled for the Windows Virtual Machine.
patchSettings	PatchSettings	[Preview Feature] Specifies settings related to VM Guest Patching on Windows.
provisionVMAgent	boolean	Indicates whether virtual machine agent should be provisioned on the virtual machine. When this property is not specified in the request body, it is set to true by default. This will ensure that VM Agent is installed on the VM so that extensions can be added to the VM later.
timeZone	string	Specifies the time zone of the virtual machine. e.g. "Pacific Standard Time". Possible values can be TimeZoneInfo.Id value from time zones returned by TimeZoneInfo.GetSystemTimeZones.
winRM	Win RMConfiguration	Specifies the Windows Remote Management listeners. This enables remote Windows PowerShell.

WindowsPatchAssessmentMode

Enumeration

Specifies the mode of VM Guest patch assessment for the IaaS virtual machine.

Possible values are:

ImageDefault - You control the timing of patch assessments on a virtual machine.

AutomaticByPlatform - The platform will trigger periodic patch assessments. The property provisionVMAgent must be true.

Expand table

Value	Description
ImageDefault	
AutomaticByPlatform	

WindowsVMGuestPatchAutomaticByPlatformRebootSetting

Enumeration

Specifies the reboot setting for all AutomaticByPlatform patch installation operations.

Expand table

Value	Description
Unknown	
IfRequired	
Never	
Always	

WindowsVMGuestPatchAutomaticByPlatformSettings

Object

Specifies additional settings to be applied when patch mode AutomaticByPlatform is selected in Windows patch settings.

Expand table

Name	Type	Description
bypassPlatformSafetyChecksOnUserSchedule	boolean	Enables customer to schedule patching without accidental upgrades
rebootSetting	WindowsVMGuestPatchAutomaticByPlatformRebootSetting	Specifies the reboot setting for all AutomaticByPlatform patch installation operations.

WindowsVMGuestPatchMode

Enumeration

Specifies the mode of VM Guest Patching to IaaS virtual machine or virtual machines associated to virtual machine scale set with OrchestrationMode as Flexible.

Possible values are:

Manual - You control the application of patches to a virtual machine. You do this by applying patches manually inside the VM. In this mode, automatic updates are disabled; the property WindowsConfiguration.enableAutomaticUpdates must be false

AutomaticByOS - The virtual machine will automatically be updated by the OS. The property

WindowsConfiguration.enableAutomaticUpdates must be true.

AutomaticByPlatform - the virtual machine will automatically updated by the platform. The properties provisionVMAgent and WindowsConfiguration.enableAutomaticUpdates must be true

Expand table

Value	Description
Manual	
AutomaticByOS	
AutomaticByPlatform	

WinRMConfiguration

Object

Describes Windows Remote Management configuration of the VM

Expand table

Name	Type	Description
listeners	WinRMListener[]	The list of Windows Remote Management listeners

WinRMListener

Object

Describes Protocol and thumbprint of Windows Remote Management listener

Expand table

Name	Type	Description

certificateUrl	string	This is the URL of a certificate that has been uploaded to Key Vault as a secret. For adding a secret to the Key Vault, see Add a key or secret to the key vault. In this case, your certificate needs to be the Base64 encoding of the following JSON Object which is encoded in UTF-8:
		
		{
		"data":"<Base64-encoded-certificate>",
		"dataType":"pfx",
		"password":"<pfx-file-password>"
		}
		To install certificates on a virtual machine it is recommended to use the Azure Key Vault virtual machine extension for Linux or the Azure Key Vault virtual machine extension for Windows.

protocol	ProtocolTypes	Specifies the protocol of WinRM listener. Possible values are: **http, https.**

ZoneMovement

Object

Describes zone movement configuration. This allows VM to be moved across availability zones during an outage.

Expand table

Name	Type	Description
isEnabled	boolean	Indicates if zone movement is enabled. By default isEnabled is set to false i.e VM can't be moved from one zone to another.

ZonePlacementPolicyType

Enumeration

Specifies the policy for resource's placement in availability zone. Possible values are: **Any** (used for Virtual Machines), **Auto** (used for Virtual Machine Scale Sets) - An availability zone will be automatically picked by system as part of resource creation.

Expand table

Value	Description
Any	
Auto	Automatic zone placement in a Virtual Machine Scale Set.

Tutorial: Install a LAMP stack on an Azure Linux VM

Article • 05/02/2025

Applies to: ✔ Linux VMs

This article walks you through how to deploy an Apache web server, MySQL, and PHP (the LAMP stack) on an Ubuntu VM in Azure. To see the LAMP server in action, you can optionally install and configure a WordPress site. In this tutorial you learn how to:

- ✔ Create an Ubuntu VM
- ✔ Open port 80 for web traffic
- ✔ Install Apache, MySQL, and PHP
- ✔ Verify installation and configuration
- ✔ Install WordPress

This setup is for quick tests or proof of concept. For more on the LAMP stack, including recommendations for a production environment, see the Ubuntu documentation.

This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

If you choose to install and use the CLI locally, this tutorial requires that you're running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Create a resource group

Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group using environment variables and appends a random suffix to ensure uniqueness.

Azure CLI

```
export REGION="eastus2"
export RANDOM_SUFFIX="$(openssl rand -hex 3)"
export MY_RESOURCE_GROUP_NAME="myResourceGroup${RANDOM_SUFFIX}"
az group create --name "${MY_RESOURCE_GROUP_NAME}" --location $REGION
```

Results:

JSON

```
{
  "id": "/subscriptions/xxxxx-xxxxx-xxxxx-
xxxxx/resourceGroups/myResourceGroupxxxxx",
  "location": "eastus",
  "name": "myResourceGroupxxxxx",
  "properties": {
      "provisioningState": "Succeeded"
  }
}
```

Create a virtual machine

Create a VM with the az vm create command.

The following example creates a VM using environment variables. It creates a VM named *myVM* and creates SSH keys if they don't already exist in a default key location. To use a specific set of keys, use the `--ssh-key-value` option. The command also sets *azureuser* as an administrator user name. You use this name later to connect to the VM.

Azure CLI

```
export MY_VM_NAME="myVM${RANDOM_SUFFIX}"
export IMAGE="Ubuntu2204"
export ADMIN_USERNAME="azureuser"
az vm create \
    --resource-group "${MY_RESOURCE_GROUP_NAME}" \
    --name $MY_VM_NAME \
    --image $IMAGE \
    --admin-username $ADMIN_USERNAME \
    --generate-ssh-keys
```

When the VM has been created, the Azure CLI shows information similar to the following example. Take note of the `publicIpAddress`. This address is used to access the VM in later steps.

Output

```
{
  "fqdns": "",
  "id": "/subscriptions/<subscription
ID>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myV
M",
  "location": "eastus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
```

```
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "40.68.254.142",
  "resourceGroup": "myResourceGroup"
}
```

Open port 80 for web traffic

By default, only SSH connections are allowed into Linux VMs deployed in Azure. Because this VM is going to be a web server, you need to open port 80 from the internet. Use the az vm open-port command to open the desired port.

Azure CLI

```
az vm open-port --port 80 --resource-group "${MY_RESOURCE_GROUP_NAME}" --name
$MY_VM_NAME
```

For more information about opening ports to your VM, see Open ports.

SSH into your VM

If you don't already know the public IP address of your VM, run the az network public-ip list command. You need this IP address for several later steps.

Azure CLI

```
export PUBLIC_IP=$(az network public-ip list --resource-group
"${MY_RESOURCE_GROUP_NAME}" --query [].ipAddress -o tsv)
```

Use the `ssh` command to create an SSH session with the virtual machine. Substitute the correct public IP address of your virtual machine.

Install Apache, MySQL, and PHP

Run the following command to update Ubuntu package sources and install Apache, MySQL, and PHP. Note the caret (^) at the end of the command, which is part of the `lamp-server^` package name.

Bash

```
ssh -o StrictHostKeyChecking=no azureuser@$PUBLIC_IP "sudo apt-get update && sudo
DEBIAN_FRONTEND=noninteractive apt-get -y install lamp-server^"
```

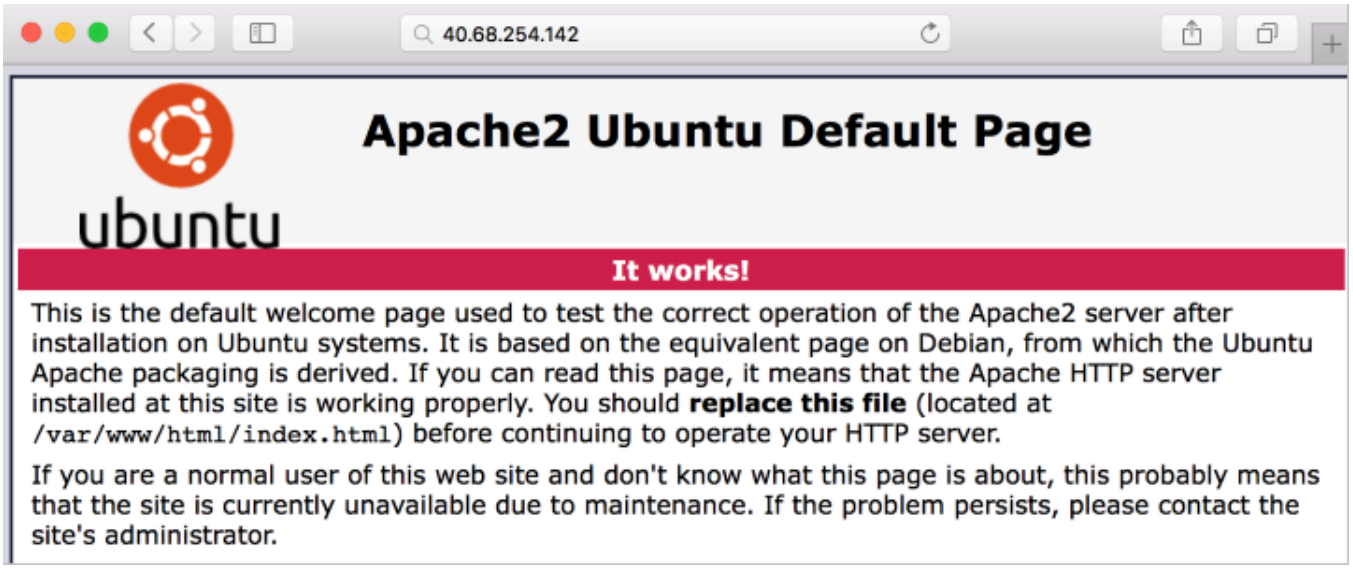
You're prompted to install the packages and other dependencies. This process installs the minimum required PHP extensions needed to use PHP with MySQL.

Verify Apache

Check the version of Apache with the following command:

```bash
ssh -o StrictHostKeyChecking=no azureuser@$PUBLIC_IP "apache2 -v"
```

With Apache installed, and port 80 open to your VM, the web server can now be accessed from the internet. To view the Apache2 Ubuntu Default Page, open a web browser, and enter the public IP address of the VM. Use the public IP address you used to SSH to the VM:



Verify and secure MySQL

Check the version of MySQL with the following command (note the capital `V` parameter):

```bash
ssh -o StrictHostKeyChecking=no azureuser@$PUBLIC_IP "mysql -V"
```

To help secure the installation of MySQL, including setting a root password, you can run the `sudo mysql_secure_installation` command. This command prompts you to answer several questions to help secure your MySQL installation.

You can optionally set up the Validate Password Plugin (recommended). Then, set a password for the MySQL root user, and configure the remaining security settings for your environment.

We recommend that you answer "Y" (yes) to all questions.

If you want to try MySQL features (create a MySQL database, add users, or change configuration settings), login to MySQL. This step isn't required to complete this tutorial. For doing this, you can use the `sudo mysql -u root -p` command in your VM and then enter your root password when prompted. This command connects to your VM via SSH and launches the MySQL command line client as the root user.

When done, exit the mysql prompt by typing `\q`.

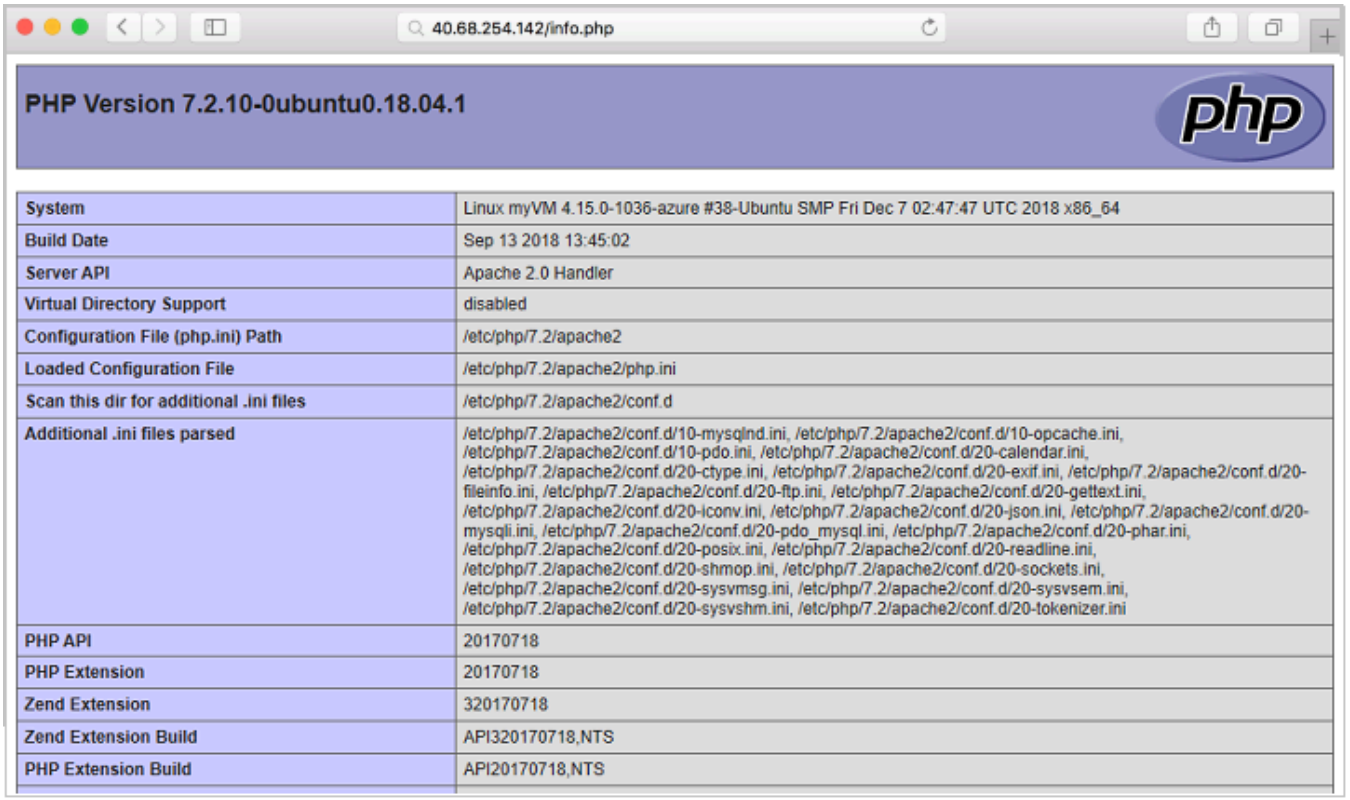
Verify PHP

Check the version of PHP with the following command:

```
Bash

ssh -o StrictHostKeyChecking=no azureuser@$PUBLIC_IP "php -v"
```

If you want to test further, you can create a quick PHP info page to view in a browser. The following command creates the PHP info page `sudo sh -c 'echo \"<?php phpinfo\(\)\; ?>\" > /var/www/html/info.php`

Now you can check the PHP info page you created. Open a browser and go to `http://yourPublicIPAddress/info.php`. Substitute the public IP address of your VM. It should look similar to this image.



Install WordPress

If you want to try your stack, install a sample app. As an example, the following steps install the open source WordPress platform to create websites and blogs. Other workloads to try include Drupal and Moodle.

This WordPress setup is only for proof of concept. To install the latest WordPress in production with recommended security settings, see the WordPress documentation.

Install the WordPress package

Run the following command:

```
Bash

sudo apt install wordpress
```

Configure WordPress

Configure WordPress to use MySQL and PHP.

In a working directory, create a text file `wordpress.sql` to configure the MySQL database for WordPress:

```
Bash

sudo sensible-editor wordpress.sql
```

Add the following commands, substituting a database password of your choice for *yourPassword* (leave other values unchanged). If you previously set up a MySQL security policy to validate password strength, make sure the password meets the strength requirements. Save the file.

```
SQL

CREATE DATABASE wordpress;
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
ON wordpress.*
TO wordpress@localhost
IDENTIFIED BY 'yourPassword';
```

Run the following command to create the database:

Bash

```
cat wordpress.sql | sudo mysql --defaults-extra-file=/etc/mysql/debian.cnf
```

Because the file `wordpress.sql` contains database credentials, delete it after use:

Bash

```
sudo rm wordpress.sql
```

To configure PHP, run the following command to open a text editor of your choice and create the file `/etc/wordpress/config-localhost.php`:

Bash

```
sudo sensible-editor /etc/wordpress/config-localhost.php
```

Copy the following lines to the file, substituting your WordPress database password for *yourPassword* (leave other values unchanged). Then save the file.

PHP

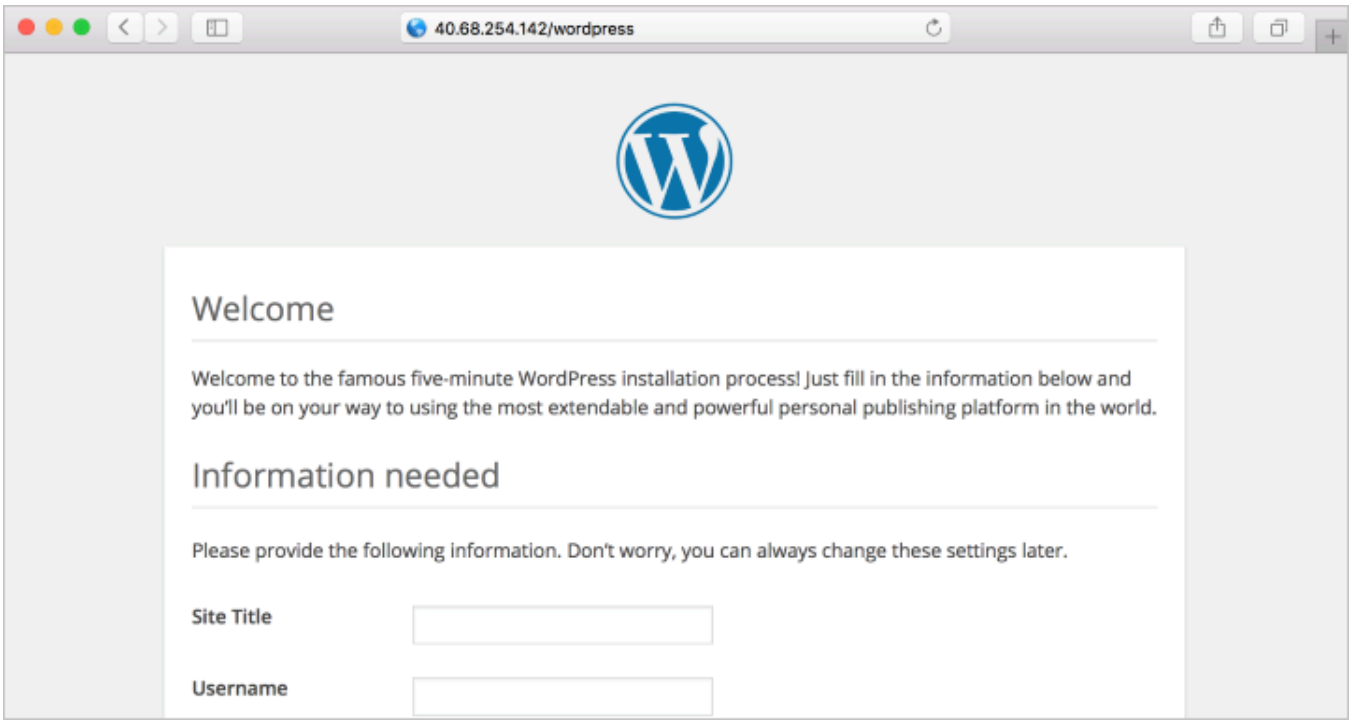
```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'yourPassword');
define('DB_HOST', 'localhost');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

Move the WordPress installation to the web server document root:

Bash

```
sudo ln -s /usr/share/wordpress /var/www/html/wordpress

sudo mv /etc/wordpress/config-localhost.php /etc/wordpress/config-default.php
```

Now you can complete the WordPress setup and publish on the platform. Open a browser and go to `http://yourPublicIPAddress/wordpress`. Substitute the public IP address of your VM. It should look similar to this image.



Next steps

In this tutorial, you deployed a LAMP server in Azure. You learned how to:

- ✓ Create an Ubuntu VM
- ✓ Open port 80 for web traffic
- ✓ Install Apache, MySQL, and PHP
- ✓ Verify installation and configuration
- ✓ Install WordPress on the LAMP server

Advance to the next tutorial to learn how to secure web servers with TLS/SSL certificates.

Secure web server with TLS

Tutorial: Install a LEMP stack on an Azure Linux VM

Article • 09/19/2024

Applies to: ✔ Linux VMs

Deploy to Azure

This article walks you through how to deploy an NGINX web server, Azure MySQL Flexible Server, and PHP (the LEMP stack) on an Ubuntu Linux VM in Azure. To see the LEMP server in action, you can optionally install and configure a WordPress site. In this tutorial you learn how to:

- ✔ Create an Ubuntu VM
- ✔ Open ports 80 and 443 for web traffic
- ✔ Install and Secure NGINX, Azure Flexible MySQL Server, and PHP
- ✔ Verify installation and configuration
- ✔ Install WordPress This setup is for quick tests or proof of concept. For more on the LEMP stack, including recommendations for a production environment, see the Ubuntu documentation.

This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

If you choose to install and use the CLI locally, this tutorial requires that you're running the Azure CLI version 2.0.30 or later. Find the version by running the `az --version` command. If you need to install or upgrade, see Install Azure CLI.

Variable declaration

First we need to define a few variables that help with the configuration of the LEMP workload.

Bash

```
export NETWORK_PREFIX="$(($RANDOM % 254 + 1))"
export RANDOM_ID="$(openssl rand -hex 3)"
export MY_RESOURCE_GROUP_NAME="myLEMPResourceGroup$RANDOM_ID"
export REGION="westeurope"
export MY_VM_NAME="myVM$RANDOM_ID"
export MY_VM_USERNAME="azureadmin"
export MY_VM_SIZE='Standard_DS2_v2'
```

```
export MY_VM_IMAGE='Canonical:0001-com-ubuntu-minimal-jammy:minimal-22_04-lts-gen2:latest'
export MY_PUBLIC_IP_NAME="myPublicIP$RANDOM_ID"
export MY_DNS_LABEL="mydnslabel$RANDOM_ID"
export MY_NSG_NAME="myNSG$RANDOM_ID"
export MY_NSG_SSH_RULE="Allow-Access$RANDOM_ID"
export MY_VM_NIC_NAME="myVMNic$RANDOM_ID"
export MY_VNET_NAME="myVNet$RANDOM_ID"
export MY_VNET_PREFIX="10.$NETWORK_PREFIX.0.0/22"
export MY_SN_NAME="mySN$RANDOM_ID"
export MY_SN_PREFIX="10.$NETWORK_PREFIX.0.0/24"
export MY_MYSQL_DB_NAME="mydb$RANDOM_ID"
export MY_MYSQL_ADMIN_USERNAME="dbadmin$RANDOM_ID"
export MY_MYSQL_ADMIN_PW="$(openssl rand -base64 32)"
export MY_MYSQL_SN_NAME="myMySQLSN$RANDOM_ID"
export MY_WP_ADMIN_PW="$(openssl rand -base64 32)"
export MY_WP_ADMIN_USER="wpcliadmin"
export MY_AZURE_USER=$(az account show --query user.name --output tsv)
export FQDN="${MY_DNS_LABEL}.${REGION}.cloudapp.azure.com"
```

Create a resource group

Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named `$MY_RESOURCE_GROUP_NAME` in the `eastus` location.

Bash

```
az group create \
    --name $MY_RESOURCE_GROUP_NAME \
    --location $REGION -o JSON
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx",
  "location": "eastus",
  "managedBy": null,
  "name": "myLEMPResourceGroupxxxxxx",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
```

```
  "type": "Microsoft.Resources/resourceGroups"
}
```


Setup LEMP networking

Create an Azure Virtual Network

A virtual network is the fundamental building block for private networks in Azure. Azure Virtual Network enables Azure resources like VMs to securely communicate with each other and the internet. Use az network vnet create to create a virtual network named `$MY_VNET_NAME` with a subnet named `$MY_SN_NAME` in the `$MY_RESOURCE_GROUP_NAME` resource group.

Bash

```
az network vnet create \
    --name $MY_VNET_NAME \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --location $REGION \
    --address-prefix $MY_VNET_PREFIX \
    --subnet-name $MY_SN_NAME \
    --subnet-prefixes $MY_SN_PREFIX -o JSON
```

Results:

JSON


```
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.19.0.0/22"
      ]
    },
    "enableDdosProtection": false,
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx",
    "location": "eastus",
    "name": "myVNetxxxxxx",
    "provisioningState": "Succeeded",
    "resourceGroup": "myLEMPResourceGroupxxxxxx",
    "subnets": [
      {
        "addressPrefix": "10.19.0.0/24",
        "delegations": [],
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
```

```
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Network/virtualNetworks/myVNetxxxxxx/subnets/mySNxxxxxx",
        "name": "mySNxxxxxx",
        "privateEndpointNetworkPolicies": "Disabled",
        "privateLinkServiceNetworkPolicies": "Enabled",
        "provisioningState": "Succeeded",
        "resourceGroup": "myLEMPResourceGroupxxxxxx",
        "type": "Microsoft.Network/virtualNetworks/subnets"
      }
    ],
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
```


Create an Azure Public IP

Use az network public-ip create to create a standard zone-redundant public IPv4 address named `MY_PUBLIC_IP_NAME` in `$MY_RESOURCE_GROUP_NAME`.

> **Note**
>
> The below options for zones are only valid selections in regions with Availability Zones.

Bash

```
az network public-ip create \
    --name $MY_PUBLIC_IP_NAME \
    --location $REGION \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --dns-name $MY_DNS_LABEL \
    --sku Standard \
    --allocation-method static \
    --version IPv4 \
    --zone 1 2 3 -o JSON
```

Results:

JSON


```
{
  "publicIp": {
    "ddosSettings": {
      "protectionMode": "VirtualNetworkInherited"
    },
    "dnsSettings": {
```

```
      "domainNameLabel": "mydnslabelxxxxxx",
      "fqdn": "mydnslabelxxxxxx.eastus.cloudapp.azure.com"
    },
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/publicIPAddresses/myPublicIPxxxxxx",
    "idleTimeoutInMinutes": 4,
    "ipTags": [],
    "location": "eastus",
    "name": "myPublicIPxxxxxx",
    "provisioningState": "Succeeded",
    "publicIPAddressVersion": "IPv4",
    "publicIPAllocationMethod": "Static",
    "resourceGroup": "myLEMPResourceGroupxxxxxx",
    "sku": {
      "name": "Standard",
      "tier": "Regional"
    },
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": [
      "1",
      "2",
      "3"
    ]
  }
}
```


Create an Azure Network Security Group

Security rules in network security groups enable you to filter the type of network traffic that can flow in and out of virtual network subnets and network interfaces. To learn more about network security groups, see Network security group overview.

Bash

```
az network nsg create \
    --name $MY_NSG_NAME \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --location $REGION -o JSON
```

Results:

JSON


```
{
  "NewNSG": {
    "defaultSecurityRules":
      {
        "access": "Allow",
```

```
        "description": "Allow inbound traffic from all VMs in VNET",
        "destinationAddressPrefix": "VirtualNetwork",
        "destinationAddressPrefixes": [],
        "destinationPortRange": "*",
        "destinationPortRanges": [],
        "direction": "Inbound",
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroup104/providers/Microsoft.Netwo
rk/networkSecurityGroups/protect-vms/defaultSecurityRules/AllowVnetInBound",
        "name": "AllowVnetInBound",
        "priority": 65000,
        "protocol": "*",
        "provisioningState": "Succeeded",
        "resourceGroup": "myLEMPResourceGroup104",
        "sourceAddressPrefix": "VirtualNetwork",
        "sourceAddressPrefixes": [],
        "sourcePortRange": "*",
        "sourcePortRanges": [],
        "type":
"Microsoft.Network/networkSecurityGroups/defaultSecurityRules"
      },
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroup104/providers/Microsoft.Netwo
rk/networkSecurityGroups/protect-vms",
    "location": "eastus",
    "name": "protect-vms",
    "provisioningState": "Succeeded",
    "resourceGroup": "myLEMPResourceGroup104",
    "securityRules": [],
    "type": "Microsoft.Network/networkSecurityGroups"
  }
}
```


Create Azure Network Security Group rules

Create a rule to allow connections to the virtual machine on port 22 for SSH and ports 80, 443 for HTTP and HTTPS. An extra rule is created to allow all ports for outbound connections. Use az network nsg rule create to create a network security group rule.

Bash

```
az network nsg rule create \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --nsg-name $MY_NSG_NAME \
    --name $MY_NSG_SSH_RULE \
    --access Allow \
    --protocol Tcp \
    --direction Inbound \
    --priority 100 \
    --source-address-prefix '*' \
    --source-port-range '*' \
```

```
    --destination-address-prefix '*' \
    --destination-port-range 22 80 443 -o JSON
```

Results:

JSON


```
{
  "access": "Allow",
  "destinationAddressPrefix": "*",
  "destinationAddressPrefixes": [],
  "destinationPortRanges": [
    "22",
    "80",
    "443"
  ],
  "direction": "Inbound",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/networkSecurityGroups/myNSGNamexxxxxx/securityRules/Allow-
Accessxxxxxx",
  "name": "Allow-Accessxxxxxx",
  "priority": 100,
  "protocol": "Tcp",
  "provisioningState": "Succeeded",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "sourceAddressPrefix": "*",
  "sourceAddressPrefixes": [],
  "sourcePortRange": "*",
  "sourcePortRanges": [],
  "type": "Microsoft.Network/networkSecurityGroups/securityRules"
}
```


Create an Azure Network Interface

Use az network nic create to create the network interface for the virtual machine. The public IP addresses and the NSG created previously are associated with the NIC. The network interface is attached to the virtual network you created previously.

Bash

```
az network nic create \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --name $MY_VM_NIC_NAME \
    --location $REGION \
    --ip-forwarding false \
    --subnet $MY_SN_NAME \
    --vnet-name $MY_VNET_NAME \
```

```
    --network-security-group $MY_NSG_NAME \
    --public-ip-address $MY_PUBLIC_IP_NAME -o JSON
```

Results:

JSON


```
{
  "NewNIC": {
    "auxiliaryMode": "None",
    "auxiliarySku": "None",
    "disableTcpStateTracking": false,
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": []
    },
    "enableAcceleratedNetworking": false,
    "enableIPForwarding": false,
    "hostedWorkloads": [],
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/networkInterfaces/myVMNicNamexxxxxx",
    "ipConfigurations": [
      {
        "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/networkInterfaces/myVMNicNamexxxxxx/ipConfigurations/ipconfig1",
        "name": "ipconfig1",
        "primary": true,
        "privateIPAddress": "10.19.0.4",
        "privateIPAddressVersion": "IPv4",
        "privateIPAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "resourceGroup": "myLEMPResourceGroupxxxxxx",
        "subnet": {
          "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx/subnets/mySNxxxxxx",
          "resourceGroup": "myLEMPResourceGroupxxxxxx"
        },
        "type": "Microsoft.Network/networkInterfaces/ipConfigurations"
      }
    ],
    "location": "eastus",
    "name": "myVMNicNamexxxxxx",
    "networkSecurityGroup": {
      "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/networkSecurityGroups/myNSGNamexxxxxx",
      "resourceGroup": "myLEMPResourceGroupxxxxxx"
    },
    "nicType": "Standard",
    "provisioningState": "Succeeded",
```

```
    "resourceGroup": "myLEMPResourceGroupxxxxxx",
    "tapConfigurations": [],
    "type": "Microsoft.Network/networkInterfaces",
    "vnetEncryptionSupported": false
  }
}
```

Cloud-init overview

Cloud-init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. As cloud-init runs during the initial boot process, there are no other steps or required agents to apply to your configuration.

Cloud-init also works across distributions. For example, you don't use apt-get install or yum install to install a package. Instead you can define a list of packages to install. Cloud-init automatically uses the native package management tool for the distro you select.

We're working with our partners to get cloud-init included and working in the images that they provide to Azure. For detailed information cloud-init support for each distribution, see Cloud-init support for VMs in Azure.

Create cloud-init config file

To see cloud-init in action, create a VM that installs a LEMP stack and runs a simple Wordpress app secured with an SSL certificate. The following cloud-init configuration installs the required packages, creates the Wordpress website, then initialize and starts the website.

Bash

```
cat << EOF > cloud-init.txt
#cloud-config
# Install, update, and upgrade packages
package_upgrade: true
package_update: true
package_reboot_if_require: true
# Install packages
packages:
  - vim
  - certbot
  - python3-certbot-nginx
  - bash-completion
  - nginx
  - mysql-client
```

```
  - php
  - php-cli
  - php-bcmath
  - php-curl
  - php-imagick
  - php-intl
  - php-json
  - php-mbstring
  - php-mysql
  - php-gd
  - php-xml
  - php-xmlrpc
  - php-zip
  - php-fpm
write_files:
  - owner: www-data:www-data
    path: /etc/nginx/sites-available/default.conf
    content: |
        server {
            listen 80 default_server;
            listen [::]:80 default_server;
            root /var/www/html;
            server_name $FQDN;
        }
write_files:
  - owner: www-data:www-data
    path: /etc/nginx/sites-available/$FQDN.conf
    content: |
        upstream php {
            server unix:/run/php/php8.1-fpm.sock;
        }
        server {
            listen 443 ssl http2;
            listen [::]:443 ssl http2;
            server_name $FQDN;
            ssl_certificate /etc/letsencrypt/live/$FQDN/fullchain.pem;
            ssl_certificate_key /etc/letsencrypt/live/$FQDN/privkey.pem;
            root /var/www/$FQDN;
            index index.php;
            location / {
                try_files \$uri \$uri/ /index.php?\$args;
            }
            location ~ \.php$ {
                include fastcgi_params;
                fastcgi_intercept_errors on;
                fastcgi_pass php;
                fastcgi_param  SCRIPT_FILENAME
\$document_root\$fastcgi_script_name;
            }
            location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
                    expires max;
                    log_not_found off;
            }
            location = /favicon.ico {
                    log_not_found off;
```

```
                access_log off;
        }
        location = /robots.txt {
                allow all;
                log_not_found off;
                access_log off;
        }
    }
    server {
        listen 80;
        listen [::]:80;
        server_name $FQDN;
        return 301 https://$FQDN\$request_uri;
    }
runcmd:
  - sed -i 's/;cgi.fix_pathinfo.*/cgi.fix_pathinfo = 1/'
/etc/php/8.1/fpm/php.ini
  - sed -i 's/^max_execution_time \= .*/max_execution_time \= 300/g'
/etc/php/8.1/fpm/php.ini
  - sed -i 's/^upload_max_filesize \= .*/upload_max_filesize \= 64M/g'
/etc/php/8.1/fpm/php.ini
  - sed -i 's/^post_max_size \= .*/post_max_size \= 64M/g'
/etc/php/8.1/fpm/php.ini
  - systemctl restart php8.1-fpm
  - systemctl restart nginx
  - certbot --nginx certonly --non-interactive --agree-tos -d $FQDN -m
dummy@dummy.com --redirect
  - ln -s /etc/nginx/sites-available/$FQDN.conf /etc/nginx/sites-enabled/
  - rm /etc/nginx/sites-enabled/default
  - systemctl restart nginx
  - curl --url https://raw.githubusercontent.com/wp-cli/builds/gh-
pages/phar/wp-cli.phar --output /tmp/wp-cli.phar
  - mv /tmp/wp-cli.phar /usr/local/bin/wp
  - chmod +x /usr/local/bin/wp
  - wp cli update
  - mkdir -m 0755 -p /var/www/$FQDN
  - chown -R azureadmin:www-data /var/www/$FQDN
  - sudo -u azureadmin -i -- wp core download --path=/var/www/$FQDN
  - sudo -u azureadmin -i -- wp config create --
dbhost=$MY_MYSQL_DB_NAME.mysql.database.azure.com --dbname=wp001 --
dbuser=$MY_MYSQL_ADMIN_USERNAME --dbpass="$MY_MYSQL_ADMIN_PW" --
path=/var/www/$FQDN
  - sudo -u azureadmin -i -- wp core install --url=$FQDN --title="Azure
hosted blog" --admin_user=$MY_WP_ADMIN_USER --
admin_password="$MY_WP_ADMIN_PW" --admin_email=$MY_AZURE_USER --
path=/var/www/$FQDN
  - sudo -u azureadmin -i -- wp plugin update --all --path=/var/www/$FQDN
  - chmod 600 /var/www/$FQDN/wp-config.php
  - mkdir -p -m 0775 /var/www/$FQDN/wp-content/uploads
  - chgrp www-data /var/www/$FQDN/wp-content/uploads
EOF
```

Create an Azure Private DNS Zone for Azure MySQL Flexible Server

Azure Private DNS Zone integration allows you to resolve the private DNS within the current VNET or any in-region peered VNET where the private DNS Zone is linked. Use az network private-dns zone create to create the private DNS zone.

Bash

```
az network private-dns zone create \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --name $MY_DNS_LABEL.private.mysql.database.azure.com -o JSON
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/privateDnsZones/mydnslabelxxxxxx.private.mysql.database.azure.com",
  "location": "global",
  "maxNumberOfRecordSets": 25000,
  "maxNumberOfVirtualNetworkLinks": 1000,
  "maxNumberOfVirtualNetworkLinksWithRegistration": 100,
  "name": "mydnslabelxxxxxx.private.mysql.database.azure.com",
  "numberOfRecordSets": 1,
  "numberOfVirtualNetworkLinks": 0,
  "numberOfVirtualNetworkLinksWithRegistration": 0,
  "provisioningState": "Succeeded",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "tags": null,
  "type": "Microsoft.Network/privateDnsZones"
}
```


Create an Azure Database for MySQL - Flexible Server

Azure Database for MySQL - Flexible Server is a managed service that you can use to run, manage, and scale highly available MySQL servers in the cloud. Create a flexible server with the az mysql flexible-server create command. A server can contain multiple databases. The following command creates a server using service defaults and variable values from your Azure CLI's local environment:

Bash

```
az mysql flexible-server create \
    --admin-password $MY_MYSQL_ADMIN_PW \
    --admin-user $MY_MYSQL_ADMIN_USERNAME \
    --auto-scale-iops Disabled \
    --high-availability Disabled \
    --iops 500 \
    --location $REGION \
    --name $MY_MYSQL_DB_NAME \
    --database-name wp001 \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --sku-name Standard_B2s \
    --storage-auto-grow Disabled \
    --storage-size 20 \
    --subnet $MY_MYSQL_SN_NAME \
    --private-dns-zone $MY_DNS_LABEL.private.mysql.database.azure.com \
    --tier Burstable \
    --version 8.0.21 \
    --vnet $MY_VNET_NAME \
    --yes -o JSON
```

Results:

JSON


```
{
  "databaseName": "wp001",
  "host": "mydbxxxxxx.mysql.database.azure.com",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.DB
forMySQL/flexibleServers/mydbxxxxxx",
  "location": "East US",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "skuname": "Standard_B2s",
  "subnetId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Ne
twork/virtualNetworks/myVNetxxxxxx/subnets/myMySQLSNxxxxxx",
  "username": "dbadminxxxxxx",
  "version": "8.0.21"
}
```


The server created has the below attributes:

- The server name, admin username, admin password, resource group name, location are already specified in local context environment of the cloud shell. They're created in the same location as your resource group and other Azure components.
- Service defaults for remaining server configurations: compute tier (Burstable), compute size/SKU (Standard_B2s), backup retention period (7 days), and MySQL

version (8.0.21)
- The default connectivity method is Private access (VNet Integration) with a linked virtual network and an auto-generated subnet.

> **Note**
>
> The connectivity method cannot be changed after creating the server. For example, if you selected `Private access (VNet Integration)` during create then you cannot change to `Public access (allowed IP addresses)` after create. We highly recommend creating a server with Private access to securely access your server using VNet Integration. Learn more about Private access in the concepts article. If you'd like to change any defaults, refer to the Azure CLI reference documentation for the complete list of configurable CLI parameters.

Check the Azure Database for MySQL - Flexible Server status

It takes a few minutes to create the Azure Database for MySQL - Flexible Server and supporting resources.

Bash

```
runtime="10 minute";
endtime=$(date -ud "$runtime" +%s);
while [[ $(date -u +%s) -le $endtime ]]; do
  STATUS=$(az mysql flexible-server show -g $MY_RESOURCE_GROUP_NAME -n
$MY_MYSQL_DB_NAME --query state -o tsv);
  echo $STATUS;
  if [ "$STATUS" == 'Ready' ]; then
    break;
  else
    sleep 10;
  fi;
done
```

Configure server parameters in Azure Database for MySQL - Flexible Server

You can manage Azure Database for MySQL - Flexible Server configuration using server parameters. The server parameters are configured with the default and recommended value when you create the server.

Show server parameter details:

Run the az mysql flexible-server parameter show command to show details about any particular parameter for the server.

Disable Azure Database for MySQL - Flexible Server SSL connection parameter for Wordpress integration

Modify a server parameter value:

You can also modify the value of a certain server parameter, which updates the underlying configuration value for the MySQL server engine. To update the server parameter, use the az mysql flexible-server parameter set command.

Bash

```
az mysql flexible-server parameter set \
    -g $MY_RESOURCE_GROUP_NAME \
    -s $MY_MYSQL_DB_NAME \
    -n require_secure_transport -v "OFF" -o JSON
```

Results:

JSON


```
{
  "allowedValues": "ON,OFF",
  "currentValue": "OFF",
  "dataType": "Enumeration",
  "defaultValue": "ON",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.DB
forMySQL/flexibleServers/mydbxxxxxx/configurations/require_secure_transport"
,
  "isConfigPendingRestart": "False",
  "isDynamicConfig": "True",
  "isReadOnly": "False",
  "name": "require_secure_transport",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "source": "user-override",
  "systemData": null,
  "type": "Microsoft.DBforMySQL/flexibleServers/configurations",
  "value": "OFF"
}
```

Create an Azure Linux Virtual Machine

The following example creates a VM named `$MY_VM_NAME` and creates SSH keys if they don't already exist in a default key location. The command also sets `$MY_VM_USERNAME` as an administrator user name.

To improve the security of Linux virtual machines in Azure, you can integrate with Microsoft Entra ID authentication. Now you can use Microsoft Entra ID as a core authentication platform. You can also SSH into the Linux VM by using Microsoft Entra ID and OpenSSH certificate-based authentication. This functionality allows organizations to manage access to VMs with Azure role-based access control and Conditional Access policies.

Create a VM with the az vm create command.

Bash

```
az vm create \
    --name $MY_VM_NAME \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --admin-username $MY_VM_USERNAME \
    --authentication-type ssh \
    --assign-identity \
    --image $MY_VM_IMAGE \
    --location $REGION \
    --nic-delete-option Delete \
    --os-disk-caching ReadOnly \
    --os-disk-delete-option Delete \
    --os-disk-size-gb 30 \
    --size $MY_VM_SIZE \
    --generate-ssh-keys \
    --storage-sku Premium_LRS \
    --nics $MY_VM_NIC_NAME \
    --custom-data cloud-init.txt -o JSON
```

Results:

JSON


```
{
  "fqdns": "mydnslabelxxxxxx.eastus.cloudapp.azure.com",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Co
mpute/virtualMachines/myVMNamexxxxxx",
  "identity": {
    "principalId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",
    "tenantId": "zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz",
    "type": "SystemAssigned",
    "userAssignedIdentities": null
```

```
  },
  "location": "eastus",
  "macAddress": "60-45-BD-D8-1D-84",
  "powerState": "VM running",
  "privateIpAddress": "10.19.0.4",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "zones": ""
}
```


Check the Azure Linux Virtual Machine status

It takes a few minutes to create the VM and supporting resources. The provisioningState value of Succeeded appears when the extension is successfully installed on the VM. The VM must have a running VM agent to install the extension.

Bash

```
runtime="5 minute";
endtime=$(date -ud "$runtime" +%s);
while [[ $(date -u +%s) -le $endtime ]]; do
    STATUS=$(ssh -o StrictHostKeyChecking=no $MY_VM_USERNAME@$FQDN "cloud-
init status --wait");
    echo $STATUS;
    if [[ "$STATUS" == *'status: done'* ]]; then
        break;
    else
        sleep 10;
    fi;
done
```

JSON


```
{
  "condition": null,
  "conditionVersion": null,
  "createdBy": null,
  "createdOn": "2023-09-04T09:29:16.895907+00:00",
  "delegatedManagedIdentityResourceId": null,
  "description": null,
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Au
thorization/roleAssignments/yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",
  "name": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",
  "principalId": "zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz",
  "principalType": "User",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "roleDefinitionId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/providers/Microsoft.Authorization/roleDefinitions/zzzzzzzz-
zzzz-zzzz-zzzz-zzzzzzzzzzzz",
  "scope": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
```

```
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx",
  "type": "Microsoft.Authorization/roleAssignments",
  "updatedBy": "wwwwwwww-wwww-wwww-wwww-wwwwwwwwwwww",
  "updatedOn": "2023-09-04T09:29:17.237445+00:00"
}
```


Enable Microsoft Entra ID login for a Linux Virtual Machine in Azure

The following installs the extension to enable Microsoft Entra ID login for a Linux VM. VM extensions are small applications that provide post-deployment configuration and automation tasks on Azure virtual machines.

Bash

```
az vm extension set \
    --publisher Microsoft.Azure.ActiveDirectory \
    --name AADSSHLoginForLinux \
    --resource-group $MY_RESOURCE_GROUP_NAME \
    --vm-name $MY_VM_NAME -o JSON
```

Results:

JSON


```
{
  "autoUpgradeMinorVersion": true,
  "enableAutomaticUpgrade": null,
  "forceUpdateTag": null,
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myLEMPResourceGroupxxxxxx/providers/Microsoft.Co
mpute/virtualMachines/myVMNamexxxxxx/extensions/AADSSHLoginForLinux",
  "instanceView": null,
  "location": "eastus",
  "name": "AADSSHLoginForLinux",
  "protectedSettings": null,
  "protectedSettingsFromKeyVault": null,
  "provisioningState": "Succeeded",
  "publisher": "Microsoft.Azure.ActiveDirectory",
  "resourceGroup": "myLEMPResourceGroupxxxxxx",
  "settings": null,
  "suppressFailures": null,
  "tags": null,
  "type": "Microsoft.Compute/virtualMachines/extensions",
  "typeHandlerVersion": "1.0",
  "typePropertiesType": "AADSSHLoginForLinux"
}
```

Check and browse your WordPress website

WordPress is an open source content management system (CMS) used by over 40% of the web to create websites, blogs, and other applications. WordPress can be run on a few different Azure services: AKS, Virtual Machines, and App Service. For a full list of WordPress options on Azure, see WordPress on Azure Marketplace.

This WordPress setup is only for proof of concept. To install the latest WordPress in production with recommended security settings, see the WordPress documentation.

Validate that the application is running by curling the application url:

Bash

```
runtime="5 minute";
endtime=$(date -ud "$runtime" +%s);
while [[ $(date -u +%s) -le $endtime ]]; do
    if curl -I -s -f $FQDN > /dev/null ; then
        curl -L -s -f $FQDN 2> /dev/null | head -n 9
        break
    else
        sleep 10
    fi;
done
```

Results:

HTML

```
<!DOCTYPE html>
<html lang="en-US">
<head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1"
/>
<meta name='robots' content='max-image-preview:large' />
<title>Azure hosted blog</title>
<link rel="alternate" type="application/rss+xml" title="Azure hosted blog
&raquo; Feed" href="https://mydnslabelxxxxxx.eastus.cloudapp.azure.com/?
feed=rss2" />
<link rel="alternate" type="application/rss+xml" title="Azure hosted blog
&raquo; Comments Feed"
href="https://mydnslabelxxxxxx.eastus.cloudapp.azure.com/?feed=comments-
rss2" />
```

Bash

```
echo "You can now visit your web server at https://$FQDN"
```

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Configure the rolling deployment strategy for Azure Linux virtual machines

Article • 08/22/2024

Applies to: ✔ Linux VMs

Azure Pipelines provides a fully featured set of CI/CD automation tools for deployments to virtual machines. This article will show you how to set up a classic release pipeline that uses the rolling strategy to deploy your web applications to Linux virtual machines.

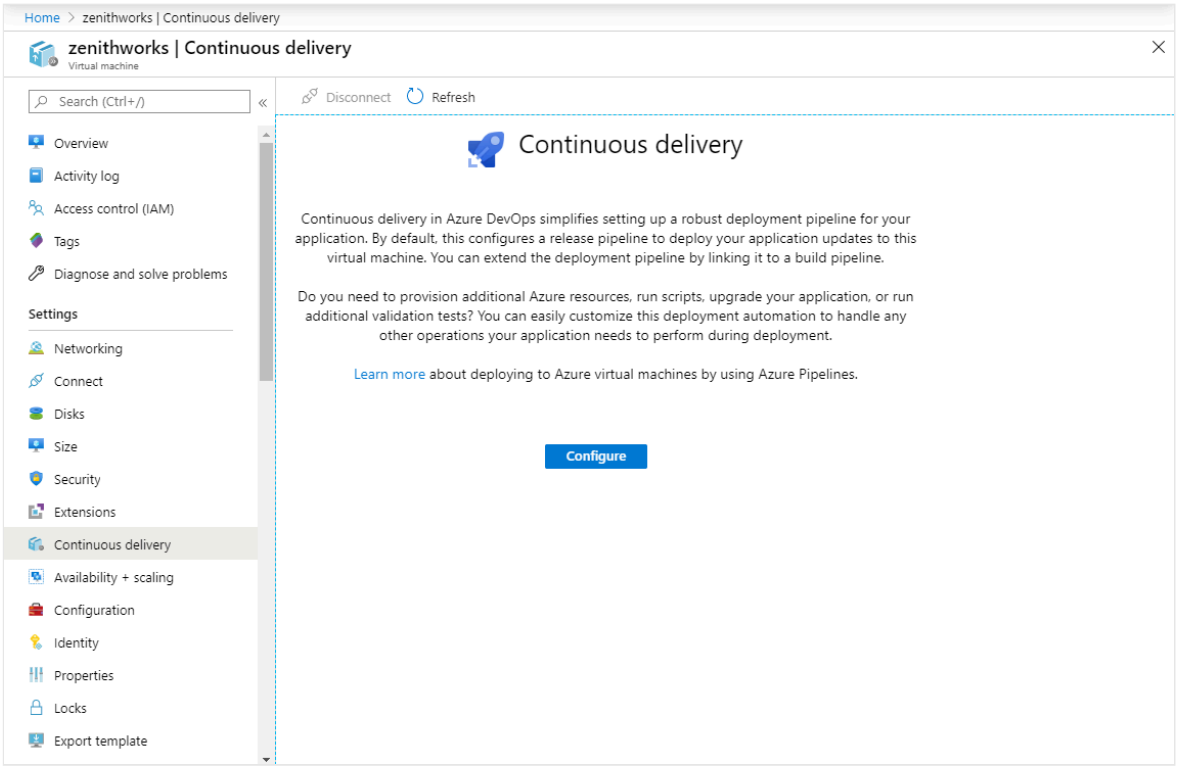
Rolling deployments

In each iteration, a rolling deployment replaces instances of an application's previous version. It replaces them with instances of the new version on a fixed set of machines (rolling set). The following walk-through shows how to configure a rolling update to virtual machines.

Using **Continuous-delivery**, you can configure rolling updates to your virtual machines within the Azure portal.

[!IMPORTANT] Virtual Machine's Continuous delivery setting will be retired on March 31, 2023. Learn more

1. Sign in to Azure portal and navigate to a virtual machine.
2. Select **Continuous delivery**, and then select **Configure**.

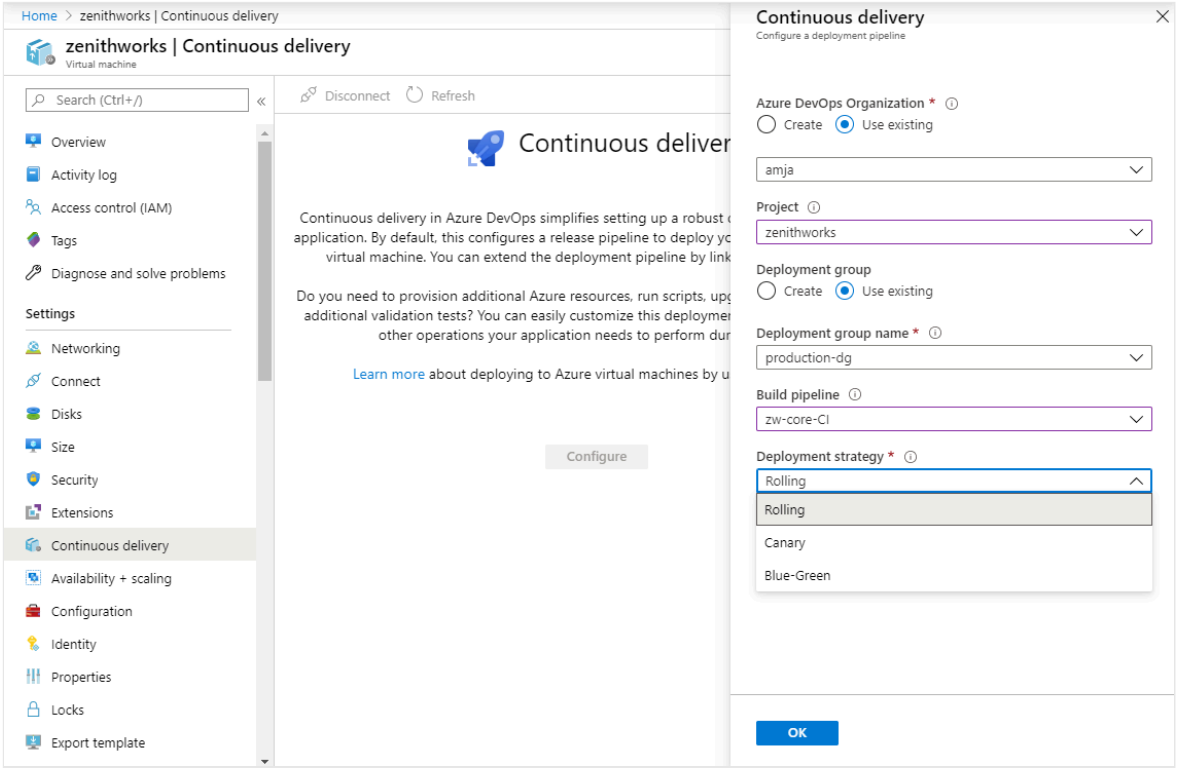


3. Select your **Azure DevOps Organization** and your **Project** from the dropdown menu or **Create** a new one.

4. Select your **Deployment group** from the dropdown menu or **Create** a new one.

5. Select your **Build pipeline**.

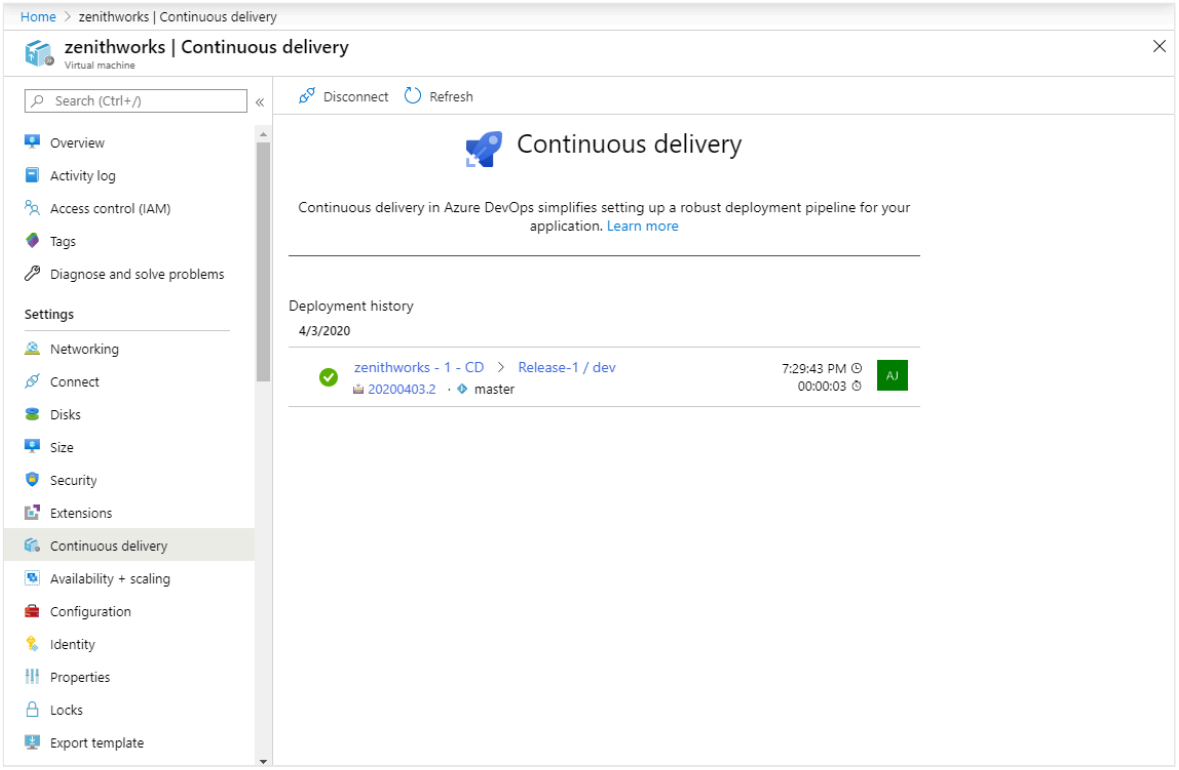
6. Select **Deployment strategy**, and then select **Rolling**.



7. Optionally, you can tag each machine with its role such as *web* or *db*. These tags help you target only VMs that have a specific role.

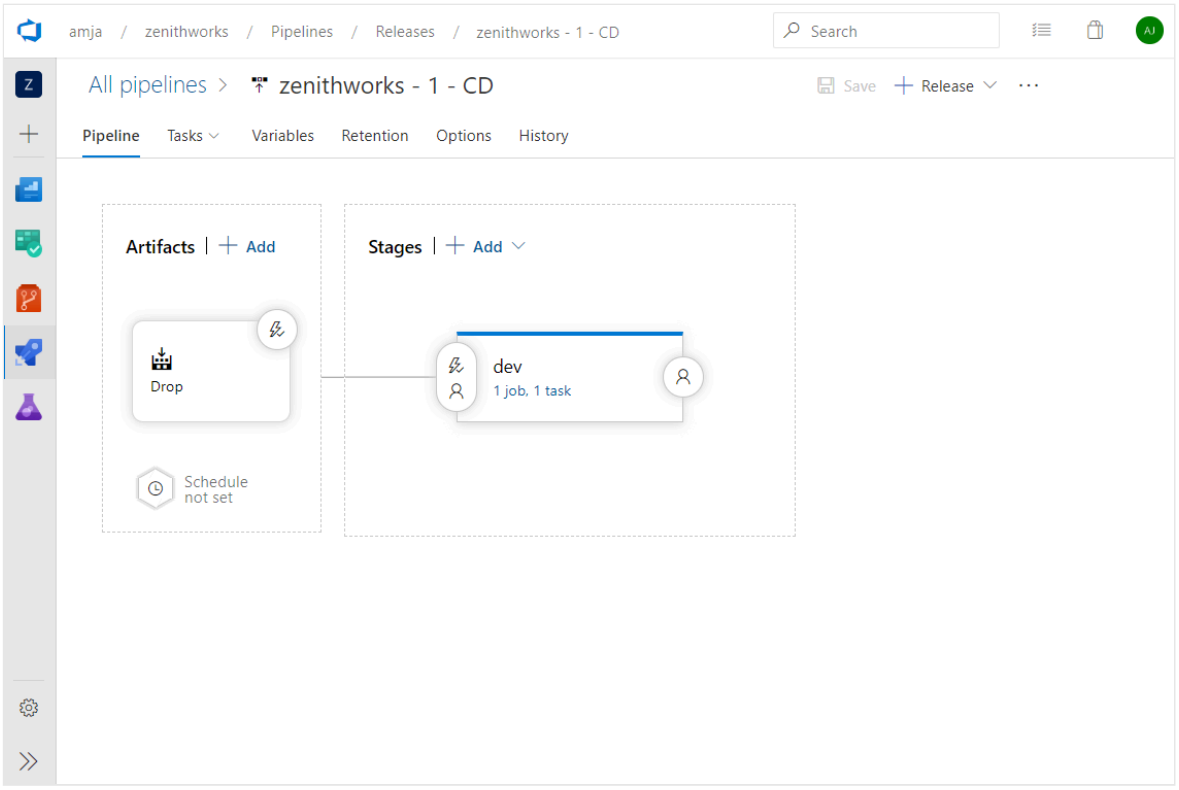
8. Select **OK** to configure the continuous delivery pipeline.

9. After completion, your continuous delivery pipeline should look similar to the following.

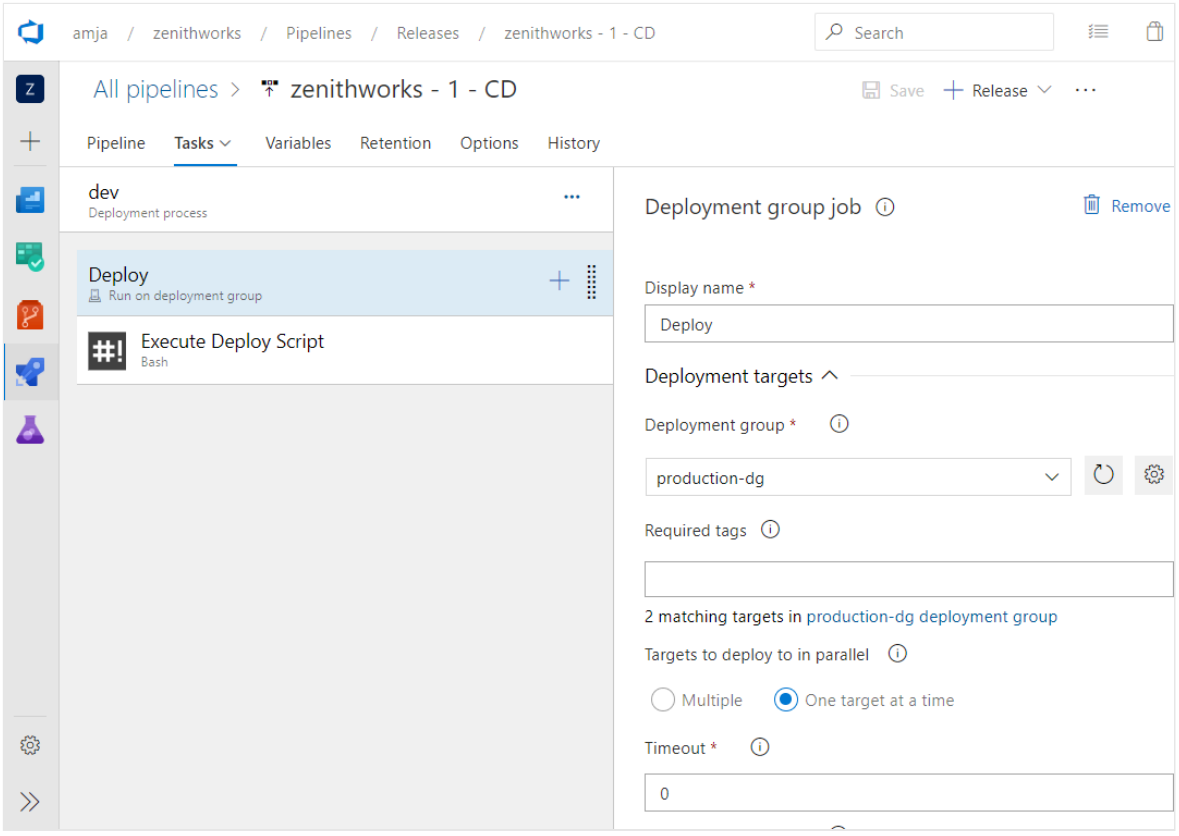


10. If you want to configure multiple VMs, repeat steps 2 through 4 for the other VMs. If you use the same deployment group that already has a configured pipeline, the new VMs will just be added to the deployment group and no new pipelines will be created.

11. Select the link to navigate to your pipeline, and then select**Edit** to modify the pipeline definition.

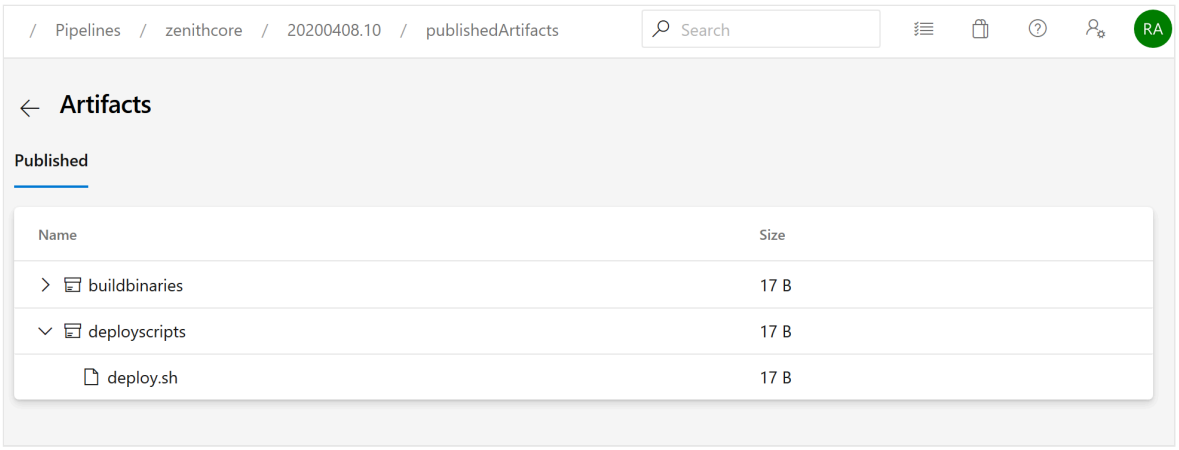


12. Select the tasks in the **dev** stage to navigate to the pipeline tasks, and then select **Deploy**.



13. You can specify the number of target machines to deploy to in parallel in each iteration. If you want to deploy to multiple machines, you can specify the number of machines as a percentage by using the slider.

14. The **Execute Deploy Script** task will execute the deployment script located in the root folder of the published artifacts.



Resources

- Deploy to Azure virtual machines with Azure DevOps
- Deploy to Azure virtual machine scale set

Related articles

- Configure the canary deployment strategy
- Configure the blue-green deployment strategy

Retirement

Continuous delivery setting of Virtual Machines will be retired on March 31, 2023. Please switch to directly using Azure DevOps to create customized pipelines for deployment to Azure VMs. Release pipeline Stage Templates and Deployments Groups Azure DevOps' features provide similar experiences.

Migration Steps

There is no migration required as VM CD experience does not store any information itself, it just helps users with their Day 0 getting started experience on Azure and Azure DevOps. Users will still be able to perform all operations from Azure DevOps after retirement. You won't be able to create and view pipelines from the Azure portal anymore.

FAQ

Where can I set up my CD pipeline after this experience is deprecated?

You won't be able to view or create Azure DevOps pipelines from an Azure portal Virtual Machine blade after retirement. You still can go to Azure DevOps portal and view or update pipelines.

Will I lose my earlier configured pipelines?

No. Your pipelines will still be available in Azure DevOps.

How can I configure different deployment strategies?

The current experience uses deployment groups to create deployment strategies. You can use deployment groups or release pipeline Stage Templates to build your pipeline with templates.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Use TLS/SSL certificates to secure a web server

Article • 08/22/2024

Applies to: ✔ Linux VMs

To secure web servers, a Transport Layer Security (TLS), previously known as Secure Sockets Layer (SSL), certificate can be used to encrypt web traffic. These TLS/SSL certificates can be stored in Azure Key Vault, and allow secure deployments of certificates to Linux virtual machines (VMs) in Azure. In this tutorial you learn how to:

- ✔ Create an Azure Key Vault
- ✔ Generate or upload a certificate to the Key Vault
- ✔ Create a VM and install the NGINX web server
- ✔ Inject the certificate into the VM and configure NGINX with a TLS binding

This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

If you choose to install and use the CLI locally, this tutorial requires that you're running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Overview

Azure Key Vault safeguards cryptographic keys and secrets, such as certificates or passwords. Key Vault helps streamline the certificate management process and enables you to maintain control of keys that access those certificates. You can create a self-signed certificate inside Key Vault, or upload an existing, trusted certificate that you already own.

Rather than using a custom VM image that includes certificates baked-in, you inject certificates into a running VM. This process ensures that the most up-to-date certificates are installed on a web server during deployment. If you renew or replace a certificate, you don't also have to create a new custom VM image. The latest certificates are automatically injected as you create more VMs. During the whole process, the certificates never leave the Azure platform or are exposed in a script, command-line history, or template.

Create an Azure Key Vault

Before you can create a Key Vault and certificates, create a resource group with az group create. The following example creates a resource group named *myResourceGroupSecureWeb* in the *eastus* location:

Azure CLI

```
az group create --name myResourceGroupSecureWeb --location eastus
```

Next, create a Key Vault with az keyvault create and enable it for use when you deploy a VM. Each Key Vault requires a unique name, and should be all lowercase. Replace <*mykeyvault*> in the following example with your own unique Key Vault name:

Azure CLI

```
keyvault_name=<mykeyvault>
az keyvault create \
    --resource-group myResourceGroupSecureWeb \
    --name $keyvault_name \
    --enabled-for-deployment
```

Generate a certificate and store in Key Vault

For production use, you should import a valid certificate signed by trusted provider with az keyvault certificate import. For this tutorial, the following example shows how you can generate a self-signed certificate with az keyvault certificate create that uses the default certificate policy:

Azure CLI

```
az keyvault certificate create \
    --vault-name $keyvault_name \
    --name mycert \
    --policy "$(az keyvault certificate get-default-policy)"
```

Prepare a certificate for use with a VM

To use the certificate during the VM create process, obtain the ID of your certificate with az keyvault secret list-versions. Convert the certificate with az vm secret format. The following example assigns the output of these commands to variables for ease of use in the next steps:

Azure CLI

```
secret=$(az keyvault secret list-versions \
          --vault-name $keyvault_name \
          --name mycert \
          --query "[?attributes.enabled].id" --output tsv)
vm_secret=$(az vm secret format --secrets "$secret" -g
myResourceGroupSecureWeb --keyvault $keyvault_name)
```

Create a cloud-init config to secure NGINX

Cloud-init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. As cloud-init runs during the initial boot process, there are no extra steps or required agents to apply your configuration.

When you create a VM, certificates and keys are stored in the protected */var/lib/waagent/* directory. To automate adding the certificate to the VM and configuring the web server, use cloud-init. In this example, you install and configure the NGINX web server. You can use the same process to install and configure Apache.

Create a file named *cloud-init-web-server.txt* and paste the following configuration:

YAML

```yaml
#cloud-config
package_upgrade: true
packages:
  - nginx
write_files:
  - owner: www-data:www-data
  - path: /etc/nginx/sites-available/default
    content: |
      server {
        listen 443 ssl;
        ssl_certificate /etc/nginx/ssl/mycert.cert;
        ssl_certificate_key /etc/nginx/ssl/mycert.prv;
      }
runcmd:
  - secretsname=$(find /var/lib/waagent/ -name "*.prv" | cut -c -57)
  - mkdir /etc/nginx/ssl
  - cp $secretsname.crt /etc/nginx/ssl/mycert.cert
  - cp $secretsname.prv /etc/nginx/ssl/mycert.prv
  - service nginx restart
```

Create a secure VM

Now create a VM with az vm create. The certificate data is injected from Key Vault with the `--secrets` parameter. You pass in the cloud-init config with the `--custom-data` parameter:

Azure CLI

```
az vm create \
    --resource-group myResourceGroupSecureWeb \
    --name myVM \
    --image Ubuntu2204 \
    --admin-username azureuser \
    --generate-ssh-keys \
    --custom-data cloud-init-web-server.txt \
    --secrets "$vm_secret"
```

It takes a few minutes for the VM to be created, the packages to install, and the app to start. When the VM has been created, take note of the `publicIpAddress` displayed by the Azure CLI. This address is used to access your site in a web browser.

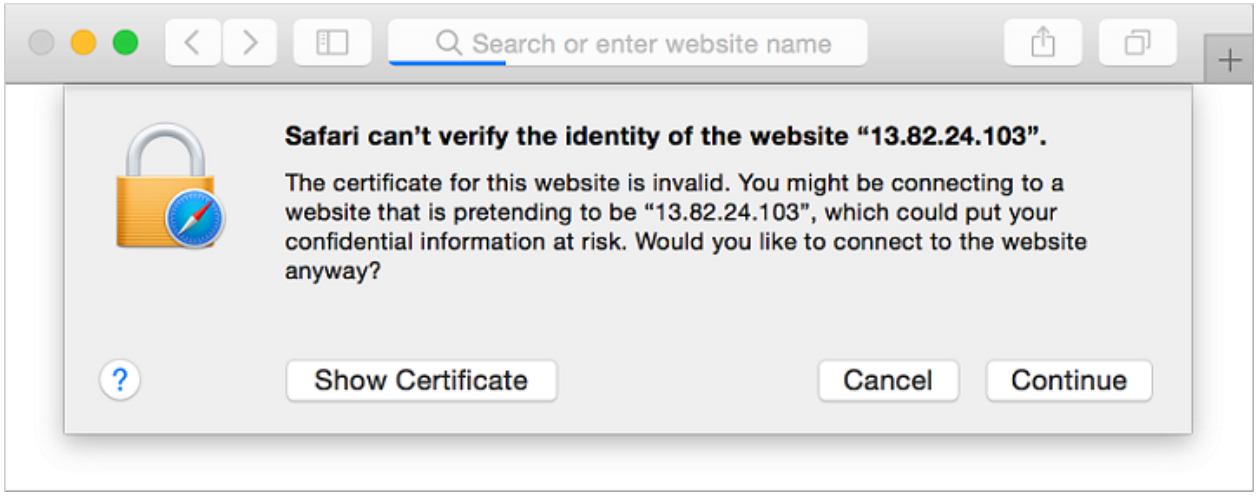
To allow secure web traffic to reach your VM, open port 443 from the Internet with az vm open-port:

Azure CLI

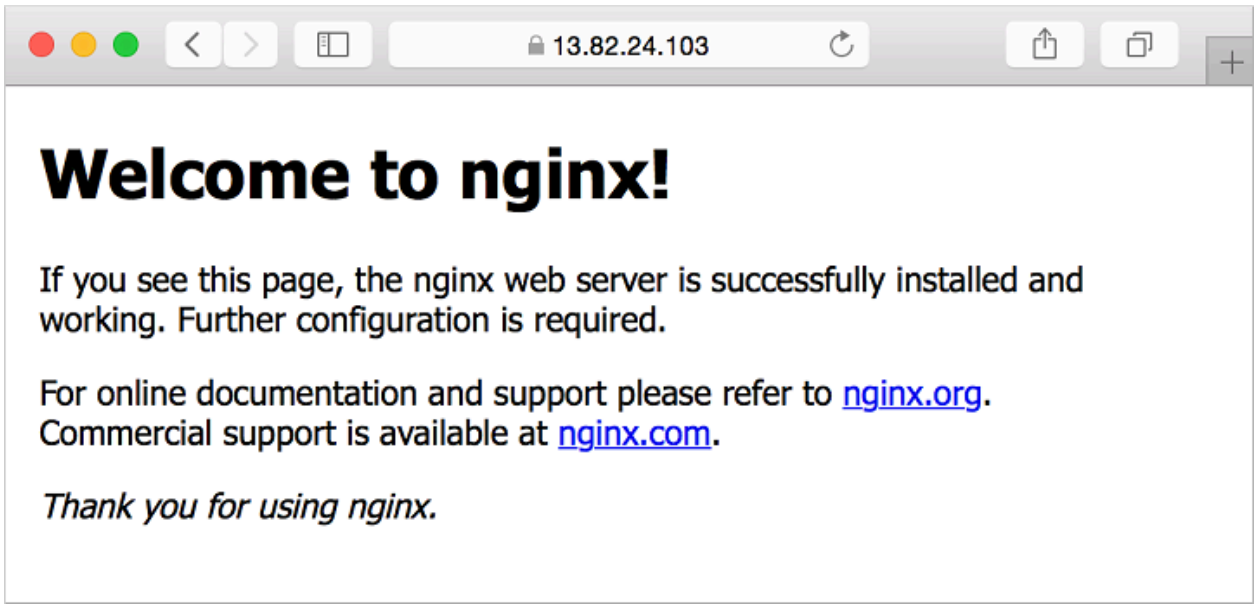
```
az vm open-port \
    --resource-group myResourceGroupSecureWeb \
    --name myVM \
    --port 443
```

Test the secure web app

Now you can open a web browser and enter *https://<publicIpAddress>* in the address bar. Provide your own public IP address from the VM create process. Accept the security warning if you used a self-signed certificate:



Your secured NGINX site is then displayed as in the following example:



Next steps

In this tutorial, you secured an NGINX web server with a TLS/SSL certificate stored in Azure Key Vault. You learned how to:

- ✓ Create an Azure Key Vault
- ✓ Generate or upload a certificate to the Key Vault
- ✓ Create a VM and install the NGINX web server
- ✓ Inject the certificate into the VM and configure NGINX with a TLS binding

Follow this link to see pre-built virtual machine script samples.

Linux virtual machine script samples

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a Windows virtual machine from a Resource Manager template

Article • 08/22/2024

Applies to: ✔ Windows VMs

Learn how to create a Windows virtual machine by using an Azure Resource Manager template and Azure PowerShell from the Azure Cloud shell. The template used in this article deploys a single virtual machine running Windows Server in a new virtual network with a single subnet. For creating a Linux virtual machine, see How to create a Linux virtual machine with Azure Resource Manager templates.

An alternative is to deploy the template from the Azure portal. To open the template in the portal, select the **Deploy to Azure** button.



Create a virtual machine

Creating an Azure virtual machine usually includes two steps:

- Create a resource group. An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine.
- Create a virtual machine.

The following example creates a generation 2 VM with Trusted Launch enabled by default from an Azure Quickstart template. Here is a copy of the template:

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "metadata": {
    "_generator": {
      "name": "bicep",
      "version": "0.16.2.56959",
      "templateHash": "14427937023370378081"
    }
  },
  "parameters": {
    "adminUsername": {
```

```
      "type": "string",
      "metadata": {
        "description": "Username for the Virtual Machine."
      }
    },
    "adminPassword": {
      "type": "securestring",
      "minLength": 12,
      "metadata": {
        "description": "Password for the Virtual Machine."
      }
    },
    "dnsLabelPrefix": {
      "type": "string",
      "defaultValue": "[toLower(format('{0}-{1}', parameters('vmName'),
uniqueString(resourceGroup().id, parameters('vmName'))))]",
      "metadata": {
        "description": "Unique DNS Name for the Public IP used to access the
Virtual Machine."
      }
    },
    "publicIpName": {
      "type": "string",
      "defaultValue": "myPublicIP",
      "metadata": {
        "description": "Name for the Public IP used to access the Virtual
Machine."
      }
    },
    "publicIPAllocationMethod": {
      "type": "string",
      "defaultValue": "Dynamic",
      "allowedValues": [
        "Dynamic",
        "Static"
      ],
      "metadata": {
        "description": "Allocation method for the Public IP used to access
the Virtual Machine."
      }
    },
    "publicIpSku": {
      "type": "string",
      "defaultValue": "Basic",
      "allowedValues": [
        "Basic",
        "Standard"
      ],
      "metadata": {
        "description": "SKU for the Public IP used to access the Virtual
Machine."
      }
    },
    "OSVersion": {
      "type": "string",
```

```
      "defaultValue": "2022-datacenter-azure-edition",
      "allowedValues": [
        "2016-datacenter-gensecond",
        "2016-datacenter-server-core-g2",
        "2016-datacenter-server-core-smalldisk-g2",
        "2016-datacenter-smalldisk-g2",
        "2016-datacenter-with-containers-g2",
        "2016-datacenter-zhcn-g2",
        "2019-datacenter-core-g2",
        "2019-datacenter-core-smalldisk-g2",
        "2019-datacenter-core-with-containers-g2",
        "2019-datacenter-core-with-containers-smalldisk-g2",
        "2019-datacenter-gensecond",
        "2019-datacenter-smalldisk-g2",
        "2019-datacenter-with-containers-g2",
        "2019-datacenter-with-containers-smalldisk-g2",
        "2019-datacenter-zhcn-g2",
        "2022-datacenter-azure-edition",
        "2022-datacenter-azure-edition-core",
        "2022-datacenter-azure-edition-core-smalldisk",
        "2022-datacenter-azure-edition-smalldisk",
        "2022-datacenter-core-g2",
        "2022-datacenter-core-smalldisk-g2",
        "2022-datacenter-g2",
        "2022-datacenter-smalldisk-g2"
      ],
      "metadata": {
        "description": "The Windows version for the VM. This will pick a
fully patched image of this given Windows version."
      }
    },
    "vmSize": {
      "type": "string",
      "defaultValue": "Standard_D2s_v5",
      "metadata": {
        "description": "Size of the virtual machine."
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Location for all resources."
      }
    },
    "vmName": {
      "type": "string",
      "defaultValue": "simple-vm",
      "metadata": {
        "description": "Name of the virtual machine."
      }
    },
    "securityType": {
      "type": "string",
      "defaultValue": "TrustedLaunch",
```

```
      "allowedValues": [
        "Standard",
        "TrustedLaunch"
      ],
      "metadata": {
        "description": "Security Type of the Virtual Machine."
      }
    }
  },
  "variables": {
    "storageAccountName": "[format('bootdiags{0}',
uniqueString(resourceGroup().id))]",
    "nicName": "myVMNic",
    "addressPrefix": "10.0.0.0/16",
    "subnetName": "Subnet",
    "subnetPrefix": "10.0.0.0/24",
    "virtualNetworkName": "MyVNET",
    "networkSecurityGroupName": "default-NSG",
    "securityProfileJson": {
      "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true
      },
      "securityType": "[parameters('securityType')]"
    },
    "extensionName": "GuestAttestation",
    "extensionPublisher": "Microsoft.Azure.Security.WindowsAttestation",
    "extensionVersion": "1.0",
    "maaTenantName": "GuestAttestation",
    "maaEndpoint": "[substring('emptyString', 0, 0)]"
  },
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2022-05-01",
      "name": "[variables('storageAccountName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "Standard_LRS"
      },
      "kind": "Storage"
    },
    {
      "type": "Microsoft.Network/publicIPAddresses",
      "apiVersion": "2022-05-01",
      "name": "[parameters('publicIpName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "[parameters('publicIpSku')]"
      },
      "properties": {
        "publicIPAllocationMethod": "
[parameters('publicIPAllocationMethod')]",
        "dnsSettings": {
          "domainNameLabel": "[parameters('dnsLabelPrefix')]"
```

```
        }
      }
    },
    {
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2022-05-01",
      "name": "[variables('networkSecurityGroupName')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "default-allow-3389",
            "properties": {
              "priority": 1000,
              "access": "Allow",
              "direction": "Inbound",
              "destinationPortRange": "3389",
              "protocol": "Tcp",
              "sourcePortRange": "*",
              "sourceAddressPrefix": "*",
              "destinationAddressPrefix": "*"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2022-05-01",
      "name": "[variables('virtualNetworkName')]",
      "location": "[parameters('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets": [
          {
            "name": "[variables('subnetName')]",
            "properties": {
              "addressPrefix": "[variables('subnetPrefix')]",
              "networkSecurityGroup": {
                "id": "
[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
              }
            }
          }
        ]
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
      ]
```

```
    },
    {
      "type": "Microsoft.Network/networkInterfaces",
      "apiVersion": "2022-05-01",
      "name": "[variables('nicName')]",
      "location": "[parameters('location')]",
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": {
                "id": "[resourceId('Microsoft.Network/publicIPAddresses',
parameters('publicIpName'))]"
              },
              "subnet": {
                "id": "
[resourceId('Microsoft.Network/virtualNetworks/subnets',
variables('virtualNetworkName'), variables('subnetName'))]"
              }
            }
          }
        ]
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses',
parameters('publicIpName'))]",
        "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]"
      ]
    },
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2022-03-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "hardwareProfile": {
          "vmSize": "[parameters('vmSize')]"
        },
        "osProfile": {
          "computerName": "[parameters('vmName')]",
          "adminUsername": "[parameters('adminUsername')]",
          "adminPassword": "[parameters('adminPassword')]"
        },
        "storageProfile": {
          "imageReference": {
            "publisher": "MicrosoftWindowsServer",
            "offer": "WindowsServer",
            "sku": "[parameters('OSVersion')]",
            "version": "latest"
          },
          "osDisk": {
            "createOption": "FromImage",
```

```
            "managedDisk": {
              "storageAccountType": "StandardSSD_LRS"
            }
          },
          "dataDisks": [
            {
              "diskSizeGB": 1023,
              "lun": 0,
              "createOption": "Empty"
            }
          ]
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('nicName'))]"
            }
          ]
        },
        "diagnosticsProfile": {
          "bootDiagnostics": {
            "enabled": true,
            "storageUri": "
[reference(resourceId('Microsoft.Storage/storageAccounts',
variables('storageAccountName')), '2022-05-01').primaryEndpoints.blob]"
          }
        },
        "securityProfile": "[if(equals(parameters('securityType'),
'TrustedLaunch'), variables('securityProfileJson'), null())]"
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkInterfaces',
variables('nicName'))]",
        "[resourceId('Microsoft.Storage/storageAccounts',
variables('storageAccountName'))]"
      ]
    },
    {
      "condition": "[and(equals(parameters('securityType'),
'TrustedLaunch'),
and(equals(variables('securityProfileJson').uefiSettings.secureBootEnabled,
true()), equals(variables('securityProfileJson').uefiSettings.vTpmEnabled,
true())))]",
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "apiVersion": "2022-03-01",
      "name": "[format('{0}/{1}', parameters('vmName'),
variables('extensionName'))]",
      "location": "[parameters('location')]",
      "properties": {
        "publisher": "[variables('extensionPublisher')]",
        "type": "[variables('extensionName')]",
        "typeHandlerVersion": "[variables('extensionVersion')]",
        "autoUpgradeMinorVersion": true,
        "enableAutomaticUpgrade": true,
```

```
        "settings": {
          "AttestationConfig": {
            "MaaSettings": {
              "maaEndpoint": "[variables('maaEndpoint')]",
              "maaTenantName": "[variables('maaTenantName')]"
            }
          }
        }
      },
      "dependsOn": [
        "[resourceId('Microsoft.Compute/virtualMachines',
parameters('vmName'))]"
      ]
    }
  ],
  "outputs": {
    "hostname": {
      "type": "string",
      "value": "[reference(resourceId('Microsoft.Network/publicIPAddresses',
parameters('publicIpName')), '2022-05-01').dnsSettings.fqdn]"
    }
  }
}
```

To run the PowerShell script, Select **Try it** to open the Azure Cloud shell. To paste the script, right-click the shell, and then select **Paste**:

Azure PowerShell

```
$resourceGroupName = Read-Host -Prompt "Enter the Resource Group name"
$location = Read-Host -Prompt "Enter the location (i.e. centralus)"
$adminUsername = Read-Host -Prompt "Enter the administrator username"
$adminPassword = Read-Host -Prompt "Enter the administrator password" -
AsSecureString
$dnsLabelPrefix = Read-Host -Prompt "Enter an unique DNS name for the public
IP"

New-AzResourceGroup -Name $resourceGroupName -Location "$location"
New-AzResourceGroupDeployment `
    -ResourceGroupName $resourceGroupName `
    -TemplateUri "https://raw.githubusercontent.com/Azure/azure-quickstart-
templates/master/quickstarts/microsoft.compute/vm-simple-
windows/azuredeploy.json" `
    -adminUsername $adminUsername `
    -adminPassword $adminPassword `
    -dnsLabelPrefix $dnsLabelPrefix

 (Get-AzVm -ResourceGroupName $resourceGroupName).name
```

If you choose to install and use the PowerShell locally instead of from the Azure Cloud shell, this tutorial requires the Azure PowerShell module. Run `Get-Module -ListAvailable Az` to find the version. If you need to upgrade, see Install Azure PowerShell module. If you're running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

In the previous example, you specified a template stored in GitHub. You can also download or create a template and specify the local path with the `-template-file` parameter.

Here are some additional resources:

- To learn how to develop Resource Manager templates, see Azure Resource Manager documentation.
- To see the Azure virtual machine schemas, see Azure template reference.
- To see more virtual machine template samples, see Azure Quickstart templates.

Connect to the virtual machine

The last PowerShell command from the previous script shows the virtual machine name. To connect to the virtual machine, see How to connect and sign on to an Azure virtual machine running Windows.

Next Steps

- If there were issues with the deployment, you might take a look at Troubleshoot common Azure deployment errors with Azure Resource Manager.
- Learn how to create and manage a virtual machine in Create and manage Windows VMs with the Azure PowerShell module.

To learn more about creating templates, view the JSON syntax and properties for the resources types you deployed:

- Microsoft.Network/publicIPAddresses
- Microsoft.Network/virtualNetworks
- Microsoft.Network/networkInterfaces
- Microsoft.Compute/virtualMachines

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Secure a web server on a Windows virtual machine in Azure with TLS certificates stored in Key Vault

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

> **Note**
>
> Currently, this doc only works for Generalized images. If you attempt this tutorial by using a Specialized disk you will receive an error.

To secure web servers, a Transport Layer Security (TLS) certificate can be used to encrypt web traffic. TLS certificates can be stored in Azure Key Vault and allow secure deployments of certificates to Windows virtual machines (VMs) in Azure. In this tutorial you learn how to:

- ✔ Create an Azure Key Vault.
- ✔ Generate or upload a certificate to the Key Vault.
- ✔ Create a VM and install the IIS web server.
- ✔ Inject the certificate into the VM and configure IIS with a TLS binding.

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Open Cloudshell** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste them into the Cloud Shell, and press enter to run them.

Overview

Azure Key Vault safeguards cryptographic keys and secrets, such as certificates or passwords. Key Vault helps streamline the certificate management process and enables you to maintain control of keys that access those certificates. You can create a self-

signed certificate inside Key Vault, or you can upload an existing, trusted certificate that you already own.

Rather than by using a custom VM image that includes certificates baked-in, inject certificates into a running VM. This process ensures that the most up-to-date certificates are installed on a web server during deployment. If you renew or replace a certificate, you don't also have to create a new custom VM image. The latest certificates are automatically injected as you create more VMs. During the whole process, the certificates never leave the Azure platform or are exposed in a script, command-line history, or template.

Create an Azure Key Vault

Before you can create a Key Vault and certificates, create a resource group with New-AzResourceGroup. The following example creates a resource group named *myResourceGroupSecureWeb* in the *East US* location:

Azure PowerShell

```
$resourceGroup = "myResourceGroupSecureWeb"
$location = "East US"
New-AzResourceGroup -ResourceGroupName $resourceGroup -Location $location
```

Next, create a Key Vault with New-AzKeyVault. Each Key Vault requires a unique name and should be all lower case. Replace `mykeyvault` with your own unique Key Vault name in the following example:

Azure PowerShell

```
$keyvaultName="mykeyvault"
New-AzKeyVault -VaultName $keyvaultName `
    -ResourceGroup $resourceGroup `
    -Location $location `
    -EnabledForDeployment
```

Generate a certificate and store it in Key Vault

For production use, you should import a valid certificate signed by a trusted provider with Import-AzKeyVaultCertificate. For this tutorial, the following example shows how you can generate a self-signed certificate with Add-AzKeyVaultCertificate that uses the default certificate policy from New-AzKeyVaultCertificatePolicy.

Azure PowerShell

```
$policy = New-AzKeyVaultCertificatePolicy `
    -SubjectName "CN=www.contoso.com" `
    -SecretContentType "application/x-pkcs12" `
    -IssuerName Self `
    -ValidityInMonths 12

Add-AzKeyVaultCertificate `
    -VaultName $keyvaultName `
    -Name "mycert" `
    -CertificatePolicy $policy
```

Create a virtual machine

Set an administrator username and password for the VM with Get-Credential:

Azure PowerShell

```
$cred = Get-Credential
```

Now you can create the VM with New-AzVM. The following example creates a VM named *myVM* in the *EastUS* location. If they don't already exist, the supporting network resources are created. To allow secure web traffic, the cmdlet also opens port *443*.

Azure PowerShell

```
# Create a VM
New-AzVm `
    -ResourceGroupName $resourceGroup `
    -Name "myVM" `
    -Location $location `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -PublicIpAddressName "myPublicIpAddress" `
    -Credential $cred `
    -OpenPorts 443

# Use the Custom Script Extension to install IIS
Set-AzVMExtension -ResourceGroupName $resourceGroup `
    -ExtensionName "IIS" `
    -VMName "myVM" `
    -Location $location `
    -Publisher "Microsoft.Compute" `
    -ExtensionType "CustomScriptExtension" `
    -TypeHandlerVersion 1.8 `
```

```
    -SettingString '{"commandToExecute":"powershell Add-WindowsFeature Web-
Server -IncludeManagementTools"}'
```

It takes a few minutes for the VM to be created. The last step uses the Azure Custom Script Extension to install the IIS web server with Set-AzVmExtension.

Add a certificate to VM from Key Vault

To add the certificate from Key Vault to a VM, obtain the ID of your certificate with Get-AzKeyVaultSecret. Add the certificate to the VM with Add-AzVMSecret:

Azure PowerShell

```
$certURL=(Get-AzKeyVaultSecret -VaultName $keyvaultName -Name "mycert").id

$vm=Get-AzVM -ResourceGroupName $resourceGroup -Name "myVM"
$vaultId=(Get-AzKeyVault -ResourceGroupName $resourceGroup -VaultName
$keyVaultName).ResourceId
$vm = Add-AzVMSecret -VM $vm -SourceVaultId $vaultId -CertificateStore "My"
-CertificateUrl $certURL | Update-AzVM
```

Configure IIS to use the certificate

Use the Custom Script Extension again with Set-AzVMExtension to update the IIS configuration. This update applies the certificate injected from Key Vault to IIS and configures the web binding:

Azure PowerShell

```
$publicSettings = '{
    "fileUris":["https://raw.githubusercontent.com/Azure-Samples/compute-
automation-configurations/master/secure-iis.ps1"],
    "commandToExecute":"powershell -ExecutionPolicy Unrestricted -File
secure-iis.ps1"
}'

Set-AzVMExtension -ResourceGroupName $resourceGroup `
    -ExtensionName "IIS" `
    -VMName "myVM" `
    -Location $location `
    -Publisher "Microsoft.Compute" `
    -ExtensionType "CustomScriptExtension" `
    -TypeHandlerVersion 1.8 `
    -SettingString $publicSettings
```

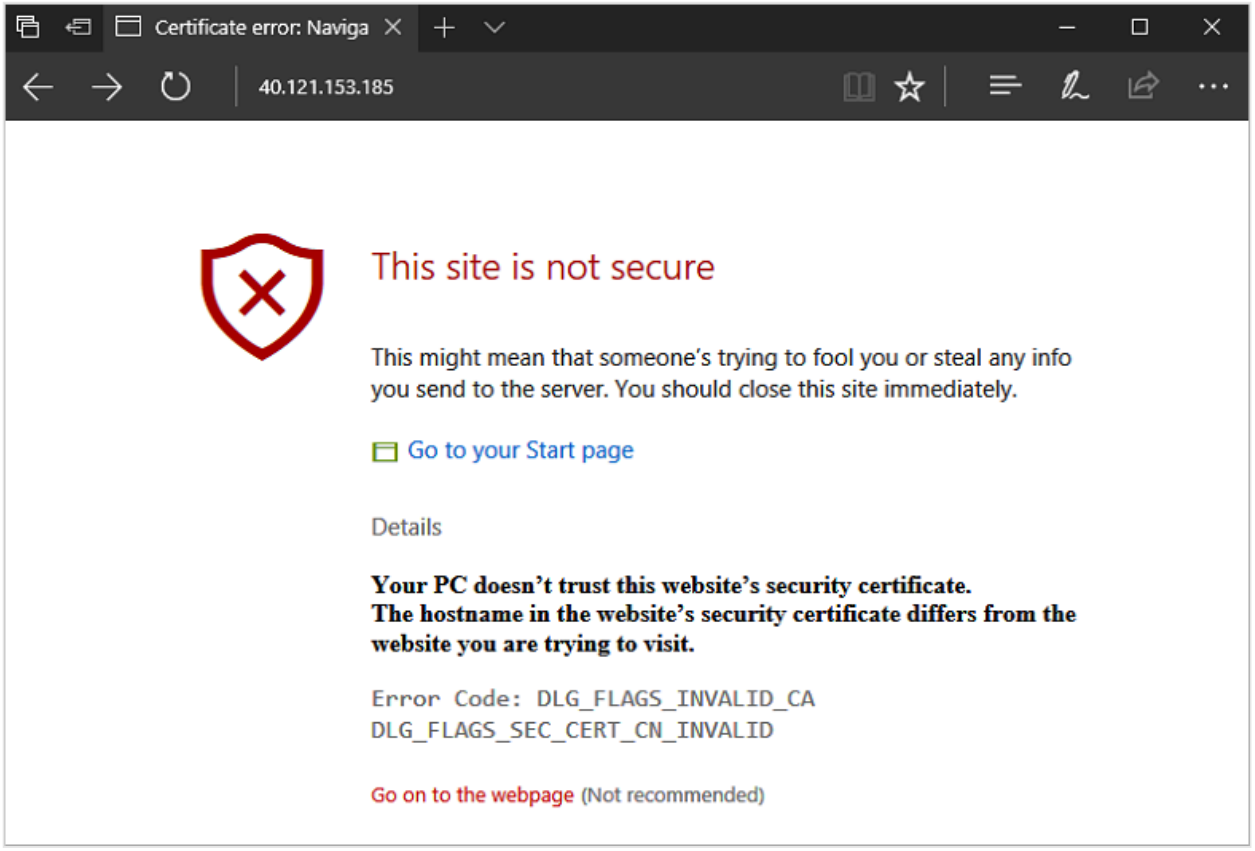
Test the secure web app

Obtain the public IP address of your VM with Get-AzPublicIPAddress. The following example obtains the IP address for `myPublicIP` created earlier:

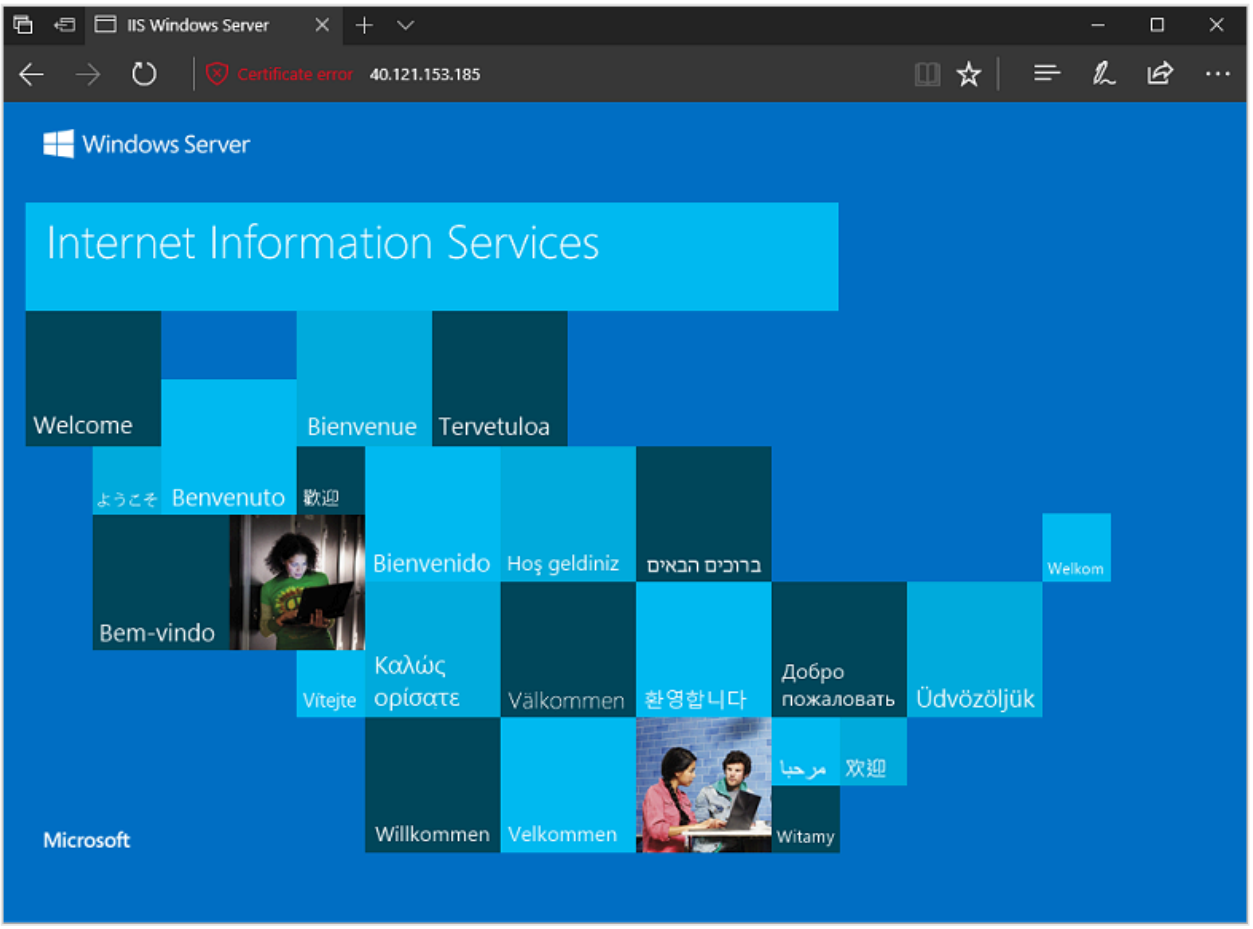
Azure PowerShell

```
Get-AzPublicIPAddress -ResourceGroupName $resourceGroup -Name
"myPublicIPAddress" | select "IpAddress"
```

Now you can open a web browser and enter `https://<myPublicIP>` in the address bar. To accept the security warning if you used a self-signed certificate, select **Details** and then **Go on to the webpage**:



Your secured IIS website is then displayed as in the following example:



Next steps

In this tutorial, you secured an IIS web server with a TLS certificate stored in Azure Key Vault. You learned how to:

- ✓ Create an Azure Key Vault.
- ✓ Generate or upload a certificate to the Key Vault.
- ✓ Create a VM and install the IIS web server.
- ✓ Inject the certificate into the VM and configure IIS with a TLS binding.

For prebuilt virtual machine script samples, see:

Windows virtual machine script samples

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a VM from a specialized disk using PowerShell

Article • 12/12/2024

Applies to: ✔ Windows VMs

Create a new VM by attaching an existing OS disk to a new VM. This option is useful if you have a VM that isn't working correctly. You can delete the VM and then reuse the disk to create a new VM.

> **Important**
>
> You can also use the VHD as a source to create an Azure Compute Gallery image. For more information, see Create an image definition and image version. Customers are encouraged to use Azure Compute Gallery as all new features like ARM64, Trusted Launch, and Confidential VM, are only supported through Azure Compute Gallery. Creating an image instead of just attaching a disk means you can create multiple VMs from the same source disk.
>
> When you use a specialized disk to create a new VM, the new VM retains the computer name of the original VM. Other computer-specific information (like the CMID) is also kept and, in some cases, this duplicate information could cause issues. When copying a VM, be aware of what types of computer-specific information your applications rely on.

We recommend that you limit the number of concurrent deployments to 20 VMs from a single VHD or snapshot.

Portal

Create a snapshot and then create a disk from the snapshot. This strategy allows you to keep the original VHD as a fallback:

1. Open the Azure portal.
2. In the search box, enter **disks** and then select **Disks** to display the list of available disks.
3. Select the disk that you would like to use. The **Disk** page for that disk appears.
4. From the menu at the top, select **Create snapshot**.
5. Choose a **Resource group** for the snapshot. You can use either an existing resource group or create a new one.

6. Enter a **Name** for the snapshot.
7. For **Snapshot type**, choose **Full**.
8. For **Storage type**, choose **Standard HDD**, **Premium SSD**, or **Zone-redundant** storage.
9. When you're done, select **Review + create** to create the snapshot.
10. After the snapshot has been created, select **Home** > **Create a resource**.
11. In the search box, enter **managed disk** and then select **Managed Disks** from the list.
12. On the **Managed Disks** page, select **Create**.
13. Choose a **Resource group** for the disk. You can use either an existing resource group or create a new one. This selection will also be used as the resource group where you create the VM from the disk.
14. For **Region**, you must select the same region where the snapshot is located.
15. Enter a **Name** for the disk.
16. In **Source type**, ensure **Snapshot** is selected.
17. In the **Source snapshot** drop-down, select the snapshot you want to use.
18. For **Size**, you can change the storage type and size as needed.
19. Make any other adjustments as needed and then select **Review + create** to create the disk. Once validation passes, select **Create**.

After you have the disk that you want to use, you can create the VM in the portal:

1. In the search box, enter **disks** and then select **Disks** to display the list of available disks.
2. Select the disk that you would like to use. The **Disk** page for that disk opens.
3. In the **Essentials** section, ensure that **Disk state** is listed as **Unattached**. If it isn't, you might need to either detach the disk from the VM or delete the VM to free up the disk.
4. In the menu at the top of the page, select **Create VM**.
5. On the **Basics** page for the new VM, enter a **Virtual machine name** and either select an existing **Resource group** or create a new one.
6. For **Size**, select **Change size** to access the **Size** page.
7. The disk name should be pre-filled in the **Image** section.
8. On the **Disks** page, you may notice that the **OS Disk Type** cannot be changed. This preselected value is configured at the point of Snapshot or VHD creation and will carry over to the new VM. If you need to modify disk type take a new snapshot from an existing VM or disk.
9. On the **Networking** page, you can either let the portal create all new resources or you can select an existing **Virtual network** and **Network security group**. The portal always creates a new network interface and public IP address for the new VM.

10. On the **Management** page, make any changes to the monitoring options.
11. On the **Guest config** page, add any extensions as needed.
12. When you're done, select **Review + create**.
13. If the VM configuration passes validation, select **Create** to start the deployment.

Next steps Learn more about Azure Compute Gallery.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Understand VM reboots - maintenance vs. downtime

Article • 01/15/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

There are three scenarios that can lead to virtual machines in Azure being impacted: unplanned hardware maintenance, unexpected downtime, and planned maintenance.

Unplanned hardware maintenance event

Unplanned hardware maintenance occurs when the Azure platform predicts that the hardware or any platform component associated to a physical machine, is about to fail. When the platform predicts a failure, it issues an unplanned hardware maintenance event to reduce the impact to the virtual machines hosted on that hardware. Azure uses Live Migration technology to migrate the Virtual Machines from the failing hardware to a healthy physical machine. Live Migration is a VM preserving operation that only pauses the Virtual Machine for a short time. Memory, open files, and network connections are maintained, but performance might be reduced before and/or after the event. In cases where Live Migration can't be used, the VM experiences Unexpected Downtime. Live migration is only possible when the physical machine is in a degraded but working state.

Unexpected downtime

Unexpected downtime is when the hardware or the physical infrastructure for the virtual machine encounters an issue and fails unexpectedly. Issues can include local network failures, local disk failures, or other rack level failures. When detected, the Azure platform automatically attempts to recover the hardware and make the resources available in the same Host Node. If these efforts fail, then automatically migrates (heals) your virtual machine to a healthy physical machine in the same data center. During the healing procedure, virtual machines experience downtime (reboot) and in some cases loss of the temporary drive. The attached OS and data disks are always preserved.

Virtual machines can also experience downtime in the unlikely event of an outage or disaster that affects an entire data center, or even an entire region. For these scenarios, Azure provides protection options including availability zones and paired regions.

Planned maintenance events

Planned maintenance events are periodic updates made by Microsoft to the underlying Azure platform to improve overall reliability, performance, and security of the platform infrastructure that your virtual machines run on. Most of these updates are performed without any impact upon your Virtual Machines or Cloud Services (see Maintenance that doesn't require a reboot). While the Azure platform attempts to use VM Preserving Maintenance in all possible occasions, there are rare instances when these updates require a reboot of your virtual machine to apply the required updates to the underlying infrastructure. In this case, you can perform Azure Planned Maintenance with Maintenance-Redeploy operation by initiating the maintenance for their VMs in the suitable time window. For more information, see Planned Maintenance for Virtual Machines.

Reduce downtime

To reduce the impact of downtime due to one or more of these events, we recommend the following high availability best practices for your virtual machines:

- Use Availability Zones to protect from data center failures
- Configure multiple virtual machines in an availability set for redundancy
- Use scheduled events for Linux or scheduled events for Windows to proactively respond to VM impacting events
- Configure each application tier into separate availability sets
- Combine a load balancer with availability zones or sets

Next steps

To learn more about availability options in Azure see, see Availability overview.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Auto-shutdown a virtual machine

Learn how to automatically shut down virtual machines (VMs) in Azure. The auto-shutdown feature for Azure VMs can help reduce costs by shutting down the VMs during off hours when they aren't needed and automatically restarting them when they're needed again.

Configure auto-shutdown for a virtual machine

Portal

Sign in to the Azure portal.

1. In the Azure portal, navigate to the virtual machine you want to configure auto-shutdown for.
2. In the virtual machine's detail page, select "Auto-shutdown" under the **Operations** section.
3. In the "Auto-shutdown" configuration screen, toggle the switch to "On."
4. Set the time you want the virtual machine to shut down.
5. If you want to receive notification before shutdown, select "Yes" in the "Send notification before shutdown" option and provide details in "Email Address" or "Webhook URL" as per your choice.
6. Select "Save" to save the auto-shutdown configuration.

> **Note**
>
> Remember to configure the time zone to match your requirements, as Coordinated Universal Time (UTC) is the default setting for the time zone.

Clean up resources

If you no longer need the virtual machine, delete it with the following steps:

1. Navigate to the virtual machine's **Overview** page on the left
2. Select on "Delete" from the top middle option.
3. Follow the prompts to delete the virtual machine.

For more information on how to delete a virtual machine, see delete a VM.

Next steps

Learn about sizes and how to resize a VM:

- Types of virtual machine sizes.
- Change the size of a virtual machine.

Last updated on 01/13/2026

Hibernation for Azure virtual machines

Applies to: ✔ Linux VMs ✔ Windows VMs

Hibernation allows you to pause VMs that aren't being used and save on compute costs. It's an effective cost management feature for scenarios such as:

- Virtual desktops, dev/test servers, and other scenarios where the VMs don't need to run 24/7.
- Systems with long boot times due to memory intensive applications. These applications can be initialized on VMs and hibernated. These “prewarmed” VMs can then be quickly started when needed, with the applications already up and running in the desired state.

How hibernation works

When you hibernate a VM, Azure signals the VM's operating system to perform a suspend-to-disk action. Azure stores the memory contents of the VM in the OS disk, then deallocates the VM. When the VM is started again, the memory contents are transferred from the OS disk back into memory. Applications and processes that were previously running in your VM resume from the state prior to hibernation.

Once a VM is in a hibernated state, you aren't billed for the VM usage. Your account is only billed for the storage (OS disk, data disks) and networking resources (IPs, etc.) attached to the VM.

When hibernating a VM:

- Hibernation is triggered on a VM using the Azure portal, CLI, PowerShell, SDKs, or APIs. Azure then signals the guest operating system to perform suspend-to-disk (S4).
- The VM's memory contents are stored on the OS disk. The VM is then deallocated, releases the lease on the underlying hardware, and is powered off. Refer to VM states and billing for more details on the VM deallocated state.
- Data in the temporary disk isn't persisted.
- The OS disk, data disks, and NICs remain attached to your VM. Any static IPs remain unchanged.
- You aren't billed for the VM usage for a hibernated VM.
- You continue to be billed for the storage and networking resources associated with the hibernated VM.

Supported configurations

Hibernation support is limited to certain VM sizes and OS versions. Make sure you have a supported configuration before using hibernation.

Supported operating systems

Supported operating systems, OS specific limitations, and configuration procedures are listed in the OS's documentation section.

Windows VM hibernation documentation

Linux VM hibernation documentation

Supported VM sizes

VM sizes with up to 64-GB RAM from the following General Purpose VM series support hibernation.

- Dasv5-series
- Dadsv5-series
- Dsv5-series
- Ddsv5-series
- Easv5-series
- Eadsv5-series
- Esv5-series
- Edsv5-series

VM sizes with up to 112-GB RAM from the following GPU VM series support hibernation.

- NVv4-series
- NVadsA10v5-series

If you are using any UVM-enabled compute applications, we recommend idling the application before initiating hibernation.

General limitations

- You can resize VMs that have hibernation enabled, but not when the VM is in a *Hibernated* state. The VM should either be in a *Running* or *Stopped* state.
- Hibernation is only supported with Nested Virtualization when Trusted Launch is enabled on the VM
- When a VM is hibernated, you can't attach, detach, or modify any disks or NICs associated with the VM. The VM must instead be moved to a Stop-Deallocated state.

- When a VM is hibernated, there's no capacity guarantee to ensure that there's sufficient capacity to start the VM later. In the rare case that you encounter capacity issues, you can try starting the VM at a later time. Capacity reservations don't guarantee capacity for hibernated VMs.
- You can only hibernate a VM using the Azure portal, CLI, PowerShell, SDKs and API. Hibernating the VM using guest OS operations don't result in the VM moving to a hibernated state and the VM continues to be billed.

Azure feature limitations

- Ephemeral OS disks
- Shared disks
- Availability Sets
- Virtual Machine Scale Sets in Uniform orchestration mode are not supported. Virtual Machine Scale Sets in Flexible orchestration mode are supported.
- Spot VMs
- Managed images
- Azure Backup
- Capacity reservations

Prerequisites to use hibernation

- Hibernation must be enabled on your VM.
- A persistent OS disk large enough to store the contents of the RAM, OS and other applications running on the VM is connected.
- The VM size supports hibernation.
- The VM OS supports hibernation.
- The Azure VM Agent is installed if you're using the Windows or Linux Hibernate Extensions.
- If a VM is being created from an OS disk or a Compute Gallery image, then the OS disk or Gallery Image definition supports hibernation.

Setting up hibernation

Enabling hibernation is detailed in the OS specific setup and configuration documentation:

Linux VMs

To configure hibernation on a Linux VM, check out the Linux hibernation documentation.

Windows VMs

To configure hibernation on a Windows VM, check out the Windows hibernation documentation.

Troubleshooting

Refer to the Hibernation troubleshooting guide for general troubleshooting information.

Refer to the Windows hibernation troubleshooting guide for issues with Windows guest hibernation.

Refer to the Linux hibernation troubleshooting guide for issues with Linux guest hibernation.

FAQs

- What are the charges for using this feature?
  - Once a VM is placed in a hibernated state, you aren't charged for the VM, just like how you aren't charged for VMs in a stop (deallocated) state. You're only charged for the OS disk, data disks and any static IPs associated with the VM.
- Can I enable hibernation on existing VMs?
  - Yes, you can enable hibernation on existing VMs.
- Can I resize a VM with hibernation enabled?
  - Yes, you can resize a VM with hibernation enabled. You cannot resize the VM if it's in a *Hibernated* state. Move the VM to either a *Running* or *Stopped* state before resizing.
- Can I modify a VM once it is in a hibernated state?
  - No, once a VM is in a hibernated state, you can't perform actions like resizing the VM and modifying the disks. Additionally, you can't detach any disks or networking resources that are currently attached to the VM or attach new resources to the VM. You can however stop(deallocate) or delete the VM if you want to detach these resources.
- What is the difference between stop(deallocating) and hibernating a VM?
  - When you stop(deallocate) a VM, the VM shuts down without persisting the memory contents. You can resize stop(deallocated) VMs and detach/attach disks to the VM.
  - When you hibernate a VM, the memory contents are first persisted in the OS disk, then the VM hibernates. You can't resize VMs in a hibernated state, nor detach/attach disks and networking resources to the VM.

- Can I initiate hibernation from within the VM?
  - To hibernate a VM you should use the Azure portal, CLI, PowerShell commands, SDKs and APIs. Triggering hibernation from inside the VM still results in your VM being billed for the compute resources.
- When a VM is hibernated, is there a capacity assurance at the time of starting the VM?
  - No, there's no capacity assurance for starting hibernated VMs. In rare scenarios if you encounter a capacity issue, then you can try starting the VM at a later time.

Next steps

- Learn more about Azure billing
- Learn about Azure Virtual Desktop
- Look into Azure VM Sizes

Last updated on 10/24/2025

Hibernating Linux virtual machines

Article • 01/28/2025

Applies to: ✔ Linux VMs

Hibernation allows you to pause VMs that aren't being used and save on compute costs. It's an effective cost management feature for scenarios such as:

- Virtual desktops, dev/test servers, and other scenarios where the VMs don't need to run 24/7.
- Systems with long boot times due to memory intensive applications. These applications can be initialized on VMs and hibernated. These “prewarmed” VMs can then be quickly started when needed, with the applications already up and running in the desired state.

How hibernation works

To learn how hibernation works, check out the hibernation overview.

Supported configurations

Hibernation support is limited to certain Virtual Machine (VM) sizes and Operating System (OS) versions. Make sure you have a supported configuration before using hibernation.

For a list of hibernation compatible VM sizes, check out the supported VM sizes section in the hibernation overview.

Supported Linux distros

The following Linux operating systems support hibernation:

- Ubuntu 22.04 LTS (Long-Term Support)
- Ubuntu 20.04 LTS (Long-Term Support)
- Ubuntu 18.04 LTS (Long-Term Support)
- Debian 11
- Debian 10 (with backports kernel)

- Red Hat Enterprise Linux 9.0 and higher (with minimum kernel version 5.14.0-70)
- Red Hat Enterprise Linux 8.5 and higher (with minimum kernel version 4.18.0-348)

Prerequisites and configuration limitations

- Hibernation isn't supported with Trusted Launch for Linux VMs

For general limitations, Azure feature limitations supported VM sizes, and feature prerequisites check out the "Supported configurations" section in the hibernation overview.

Creating a Linux VM with hibernation enabled

To hibernate a VM, you must first enable the feature on the VM.

To enable hibernation during VM creation, you can use the Azure portal, CLI, PowerShell, ARM templates, and API.

Portal

To enable hibernation in the Azure portal, check the 'Enable hibernation' box during VM creation.



After creating a VM with hibernation enabled, you'll need to configure the guest OS to successfully hibernate your VM.

Enabling hibernation on an existing Linux VM

To enable hibernation on an existing VM, you can use Azure CLI, PowerShell, or REST API. Before proceeding, ensure that the guest OS version supports hibernation on Azure. For more information, see supported OS versions.

CLI

To enable hibernation on an existing VM using Azure CLI, first deallocate your VM with az vm deallocate. Once the VM is deallocated, update the OS disk and VM.

1. Update the OS disk to set *supportsHibernation* to `true`. If *supportsHibernation* is already set to `true`, you can skip this step and proceed to the next step.

   Azure CLI

   ```
   az disk update --resource-group myResourceGroup \
   --name MyOSDisk \
   --set supportsHibernation=true
   ```

2. To enable hibernation, update the VM setting.

   Azure CLI

   ```
   az vm update --resource-group myResourceGroup \
   --name myVM \
   --enable-hibernation true
   ```

3. Start the VM and then proceed to configuring hibernation in the guest OS.

   Azure CLI

   ```
   az vm start --resource-group myResourceGroup \
   --name myVM \
   ```

Configuring hibernation in the guest OS

After ensuring that your VM configuration is supported, you can enable hibernation on your Linux VM using one of two options:

Option 1: LinuxHibernateExtension

Option 2: hibernation-setup-tool

LinuxHibernateExtension

> ⓘ **Note**

If you've already installed the hibernation-setup-tool, you don't need to install the LinuxHibernateExtension. These steps are redundant ways to enable hibernation on a Linux VM.

When you create a Hibernation-enabled VM via the Azure portal, the LinuxHibernationExtension is automatically installed on the VM.

If the extension is missing, you can manually install the LinuxHibernateExtension on your Linux VM to configure the guest OS for hibernation.

> **Note**
>
> Azure extensions are currently disabled by default for Debian images. To re-enable extensions, check the Linux hibernation troubleshooting guide.

> **Note**
>
> For RHEL LVM, you need to expand the root volume and ensure there's sufficient space available to create the swap file. To expand the volume, check the disk expansion guide.

CLI

To install *LinuxHibernateExtension* with the Azure CLI, run the following command:

Azure CLI

```
az vm extension set -n LinuxHibernateExtension --publisher
Microsoft.CPlat.Core --version 1.0 \    --vm-name MyVm --resource-group
MyResourceGroup --enable-auto-upgrade true
```

Hibernation-setup-tool

> **Note**
>
> If you've already installed the LinuxHibernateExtension, you don't need to install the hibernation-setup-tool. These steps are redundant ways to enable hibernation on a Linux VM.

You can install the hibernation-setup-tool package on your Linux VM from Microsoft's Linux software repository at packages.microsoft.com.

To use the Linux software repository, follow the instructions at Linux package repository for Microsoft software.

Debian/Ubuntu

To use the hibernation-setup-tool in Debian and Ubuntu versions, open git bash and run this command:

```bash
curl -sSL https://packages.microsoft.com/keys/microsoft.asc | sudo tee
etc/apt/trusted.gpg.d/microsoft.asc

sudo apt-add-repository https://packages.microsoft.com/ubuntu/20.04/prod

sudo apt-get update
```

To install the package, run this command in git bash:

```bash
sudo apt-get install hibernation-setup-tool
```

> **Note**
>
> For RHEL LVM, you need to expand the root volume and ensure there's sufficient space available to create the swap file. To expand the volume, check the disk expansion guide.

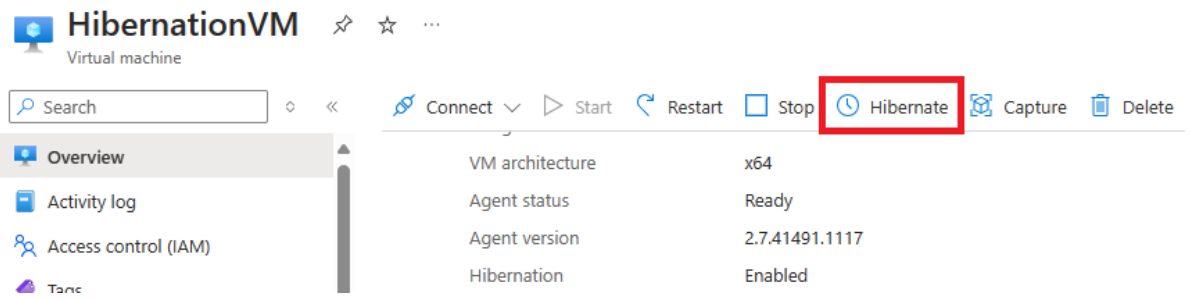
Once the package installs successfully, your Linux guest OS is configured for hibernation. You can also create a new Azure Compute Gallery Image from this VM and use the image to create VMs. VMs created with this image have the hibernation package preinstalled, simplifying your VM creation experience.

Hibernate a VM

Once a VM with hibernation enabled has been created and the guest OS is configured for hibernation, you can hibernate the VM through the Azure portal, the Azure CLI, PowerShell, or REST API.

Portal

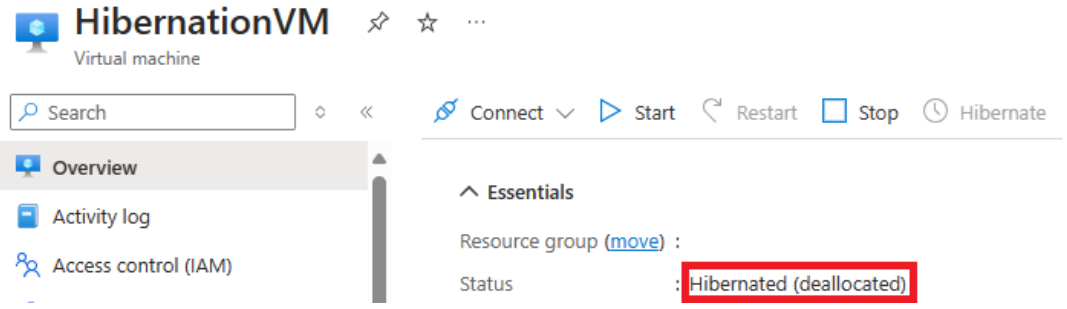
To hibernate a VM in the Azure portal, click the 'Hibernate' button on the VM Overview page.



View state of hibernated VM

Portal

To view the state of a VM in the portal, check the 'Status' on the overview page. It should report as "Hibernated (deallocated)"

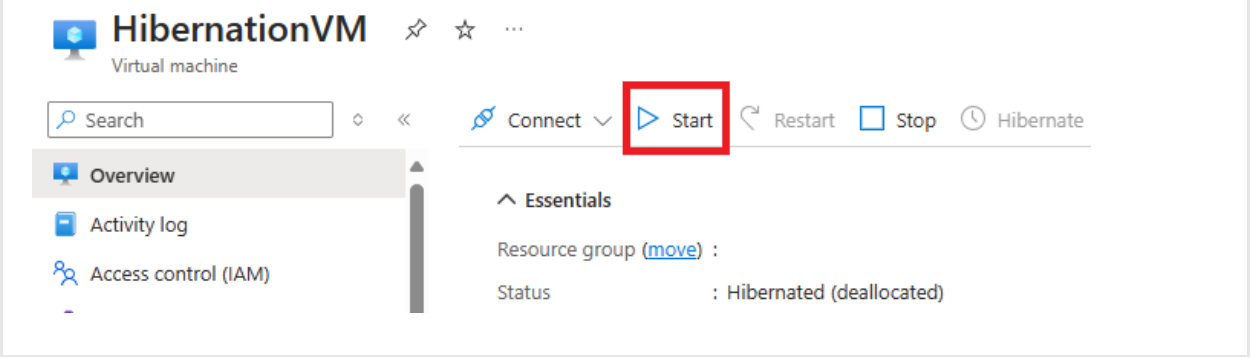


Start hibernated VMs

You can start hibernated VMs just like how you would start a stopped VM. This can be done through the Azure portal, the Azure CLI, PowerShell, or REST API.

Portal

To start a hibernated VM using the Azure portal, click the 'Start' button on the VM Overview page.



Deploy hibernation enabled VMs from the Azure Compute Gallery

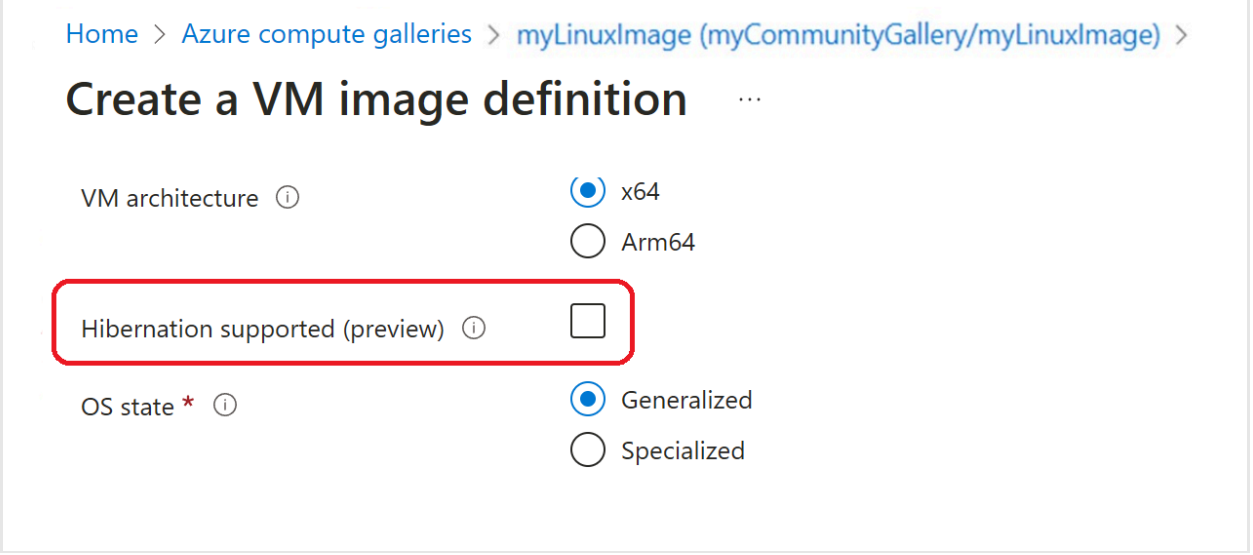
VMs created from Compute Gallery images can also be enabled for hibernation. Ensure that the OS version associated with your Gallery image supports hibernation on Azure. Refer to the list of supported OS versions.

To create VMs with hibernation enabled using Gallery images, you'll first need to create a new image definition with the hibernation property enabled. Once this feature property is enabled on the Gallery Image definition, you can create an image version and use that image version to create hibernation enabled VMs.

> **Note**
>
> For specialized Windows images, the page file location must be set to C: drive in order for Azure to successfully configure your guest OS for hibernation. If you're creating an Image version from an existing VM, you should first move the page file to the OS disk and then use the VM as the source for the Image version.

Portal

To create an image definition with the hibernation property enabled, select the checkmark for 'Enable hibernation'.



Deploy hibernation enabled VMs from an OS disk

VMs created from OS disks can also be enabled for hibernation. Ensure that the OS version associated with your OS disk supports hibernation on Azure. Refer to the list of supported OS versions.

To create VMs with hibernation enabled using OS disks, ensure that the OS disk has the hibernation property enabled. Refer to API example to enable this property on OS disks. Once the hibernation property is enabled on the OS disk, you can create hibernation enabled VMs using that OS disk.

```
PATCH https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/myDisk?
api-version=2021-12-01

{
  "properties": {
    "supportsHibernation": true
  }
}
```

Troubleshooting

For more information, see the Hibernate troubleshooting guide and the Linux VM hibernation troubleshooting guide.

FAQs

For more information, see the Hibernate FAQs.

Next steps

- Learn more about Azure billing
- Look into Azure VM Sizes

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Hibernating Windows virtual machines

Article • 08/22/2024

Applies to: ✔ Windows VMs

Hibernation allows you to pause VMs that aren't being used and save on compute costs. It's an effective cost management feature for scenarios such as:

- Virtual desktops, dev/test servers, and other scenarios where the VMs don't need to run 24/7.
- Systems with long boot times due to memory intensive applications. These applications can be initialized on VMs and hibernated. These “prewarmed” VMs can then be quickly started when needed, with the applications already up and running in the desired state.

How hibernation works

To learn how hibernation works, check out the hibernation overview.

Supported configurations

Hibernation support is limited to certain VM sizes and OS versions. Make sure you have a supported configuration before using hibernation.

For a list of hibernation compatible VM sizes, check out the supported VM sizes section in the hibernation overview.

Supported Windows versions

The following Windows operating systems support hibernation:

- Windows Server 2022
- Windows Server 2019
- Windows 11 Pro
- Windows 11 Enterprise
- Windows 11 Enterprise multi-session
- Windows 10 Pro
- Windows 10 Enterprise
- Windows 10 Enterprise multi-session

Prerequisites and configuration limitations

- The Windows page file can't be on the temp disk.
- Applications such as Device Guard and Credential Guard that require virtualization-based security (VBS) work with hibernation when you enable Trusted Launch on the VM and Nested Virtualization in the guest OS.

> **Note**
>
> To use VBS and Hibernation, Windows 11 24H2 OS build must be 26100.3037 or later. You can verify OS build using `wmic qfe list` command. Marketplace images of Windows 11 24H2 installed after February 2025 support VBS and Hibernation.

For general limitations, Azure feature limitations supported VM sizes, and feature prerequisites check out the "Supported configurations" section in the hibernation overview.

Creating a Windows VM with hibernation enabled

To hibernate a VM, you must first enable the feature on the VM.

To enable hibernation during VM creation, you can use the Azure portal, CLI, PowerShell, ARM templates and API.

Portal

To enable hibernation in the Azure portal, check the 'Enable hibernation' box during VM creation.



Once you've created a VM with hibernation enabled, you need to configure the guest OS to successfully hibernate your VM.

Enabling hibernation on an existing Windows VM

To enable hibernation on an existing VM, you can use Azure CLI, PowerShell, or REST API. Before proceeding, ensure that the guest OS version supports hibernation on Azure. For more information, see supported OS versions.

> **Note**
>
> Ensure that the page file is located on the `C: drive`. Move the page file to the `C: drive` if necessary, before continuing.

CLI

To enable hibernation on an existing VM using Azure CLI, first deallocate your VM with az vm deallocate. Once the VM is deallocated, update the OS disk and VM.

1. Update the OS disk to set *supportsHibernation* to `true`. If *supportsHibernation* is already set to `true`, you can skip this step and proceed to the next step.

   Azure CLI

   ```
   az disk update --resource-group myResourceGroup \
   --name MyOSDisk \
   --set supportsHibernation=true
   ```

2. Update the VM to enable hibernation.

   Azure CLI

   ```
   az vm update --resource-group myResourceGroup \
   --name myVM \
   --enable-hibernation true
   ```

3. Start the VM and then proceed to configuring hibernation in the guest OS.

   Azure CLI

   ```
   az vm start --resource-group myResourceGroup \
   --name myVM \
   ```

Configuring hibernation in the guest OS

Enabling hibernation while creating a Windows VM automatically installs the 'Microsoft.CPlat.Core.WindowsHibernateExtension' VM extension. This extension configures the guest OS for hibernation. This extension doesn't need to be manually installed or updated, as this extension is managed by the Azure platform.

> **Note**
>
> When you create a VM with hibernation enabled, Azure automatically places the page file on the `C: drive`. If you're enabling hibernation on an existing VM or using a specialized image, then you'll need to follow additional steps to ensure that the page file is located on the `C: drive`.

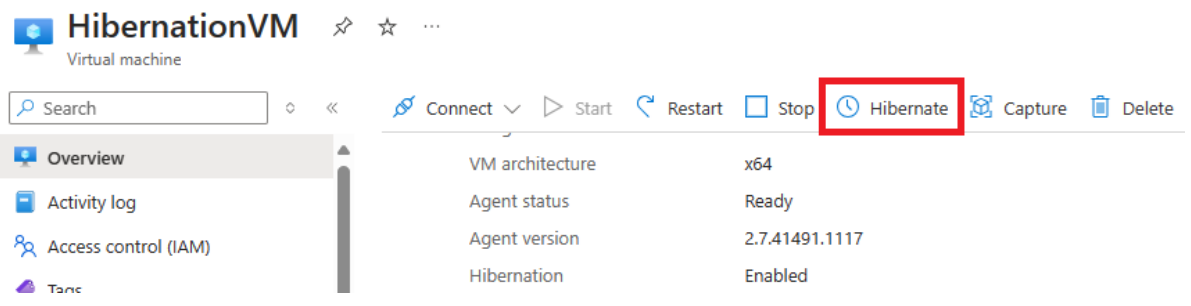
> **Note**
>
> Using the WindowsHibernateExtension requires the Azure VM Agent to be installed on the VM. If you choose to opt-out of the Azure VM Agent, then you can configure the OS for hibernation by running powercfg /h /type full inside the guest. You can then verify if hibernation is enabled inside guest using the powercfg /a command.

Hibernate a VM

Once a VM with hibernation enabled has been created and the guest OS is configured for hibernation, you can hibernate the VM through the Azure portal, the Azure CLI, PowerShell, or REST API.

Portal

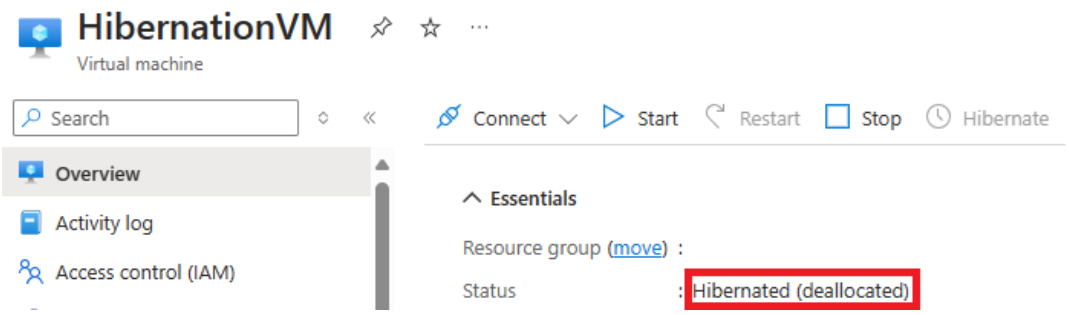
To hibernate a VM in the Azure portal, click the 'Hibernate' button on the VM Overview page.



View state of hibernated VM

Portal

To view the state of a VM in the portal, check the 'Status' on the overview page. It should report as "Hibernated (deallocated)"

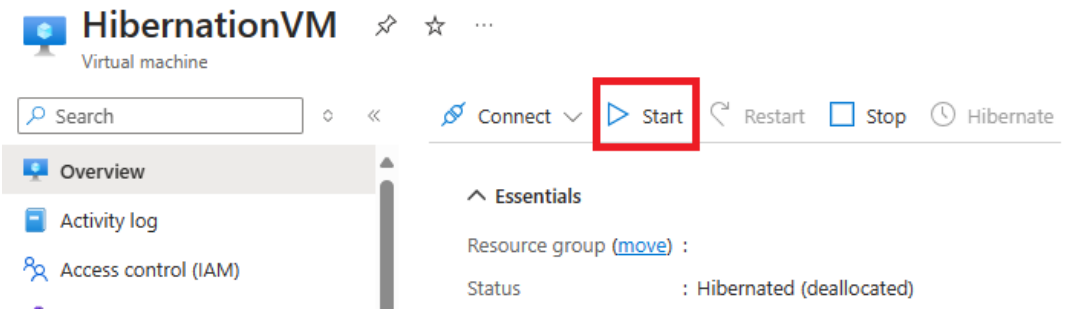


Start hibernated VMs

You can start hibernated VMs just like how you would start a stopped VM. This can be done through the Azure portal, the Azure CLI, PowerShell, or REST API.

Portal

To start a hibernated VM using the Azure portal, click the 'Start' button on the VM Overview page.



Deploy hibernation enabled VMs from the Azure Compute Gallery

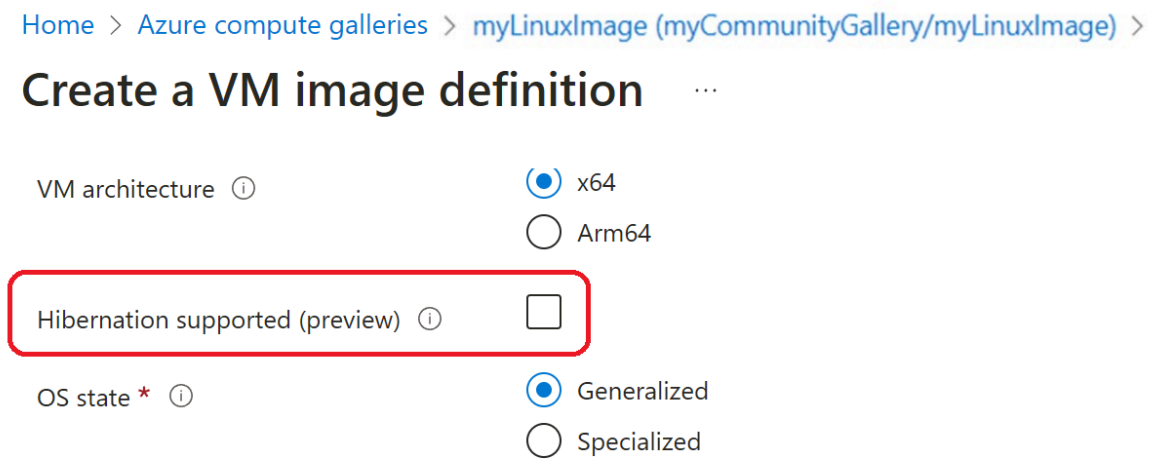
VMs created from Compute Gallery images can also be enabled for hibernation. Ensure that the OS version associated with your Gallery image supports hibernation on Azure. Refer to the list of supported OS versions.

To create VMs with hibernation enabled using Gallery images, you'll first need to create a new image definition with the hibernation property enabled. Once this feature property is enabled on the Gallery Image definition, you can create an image version and use that image version to create hibernation enabled VMs.

> **Note**
>
> For specialized Windows images, the page file location must be set to C: drive in order for Azure to successfully configure your guest OS for hibernation. If you're creating an Image version from an existing VM, you should first move the page file to the OS disk and then use the VM as the source for the Image version.

Portal

To create an image definition with the hibernation property enabled, select the checkmark for 'Enable hibernation'.



Deploy hibernation enabled VMs from an OS disk

VMs created from OS disks can also be enabled for hibernation. Ensure that the OS version associated with your OS disk supports hibernation on Azure. Refer to the list of supported OS versions.

To create VMs with hibernation enabled using OS disks, ensure that the OS disk has the hibernation property enabled. Refer to API example to enable this property on OS disks. Once the hibernation property is enabled on the OS disk, you can create hibernation enabled VMs using that OS disk.

```
PATCH https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/disks/myDisk?
api-version=2021-12-01

{
  "properties": {
    "supportsHibernation": true
  }
}
```

Troubleshooting

Refer to the Hibernate troubleshooting guide and the Windows VM hibernation troubleshooting guide for more information.

FAQs

Refer to the Hibernate FAQs for more information.

Next steps

- Learn more about Azure billing
- Look into Azure VM Sizes

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshooting hibernation in Azure

Article • 08/22/2024

Hibernating a virtual machine allows you to persist the VM state to the OS disk. This article describes how to troubleshoot issues with the hibernation feature, issues creating hibernation enabled VMs, and issues with hibernating a VM.

For information specific to Linux VMs, check out the Linux VM hibernation troubleshooting guide.

For information specific to Windows VMs, check out the Windows VM hibernation troubleshooting guide.

Unable to create a VM with hibernation enabled

If you're unable to create a VM with hibernation enabled, ensure that you're using a VM size, OS version that supports Hibernation. Refer to the supported VM sizes, OS versions section in the user guide and the limitations section for more details. Here are some common error codes that you might observe:

Expand table

ResultCode	Error Message	Action
OperationNotAllowed	The referenced os disk should support hibernation for a VM with hibernation capability.	Validate that the OS disk has hibernation support enabled.
OperationNotAllowed	The referenced platform image should support hibernation for a VM with hibernation capability.	Use a platform image that supports hibernation.
OperationNotAllowed	The referenced shared gallery image should support hibernation for a VM with hibernation capability.	Validate that the Shared Gallery Image Definition has hibernation support enabled
OperationNotAllowed	Hibernation capability isn't supported for Spot VMs.	
OperationNotAllowed	User VM Image isn't supported for a VM with Hibernation capability.	Use a platform image or Shared Gallery Image if you want to use the hibernation feature

ResultCode	Error Message	Action
OperationNotAllowed	Referencing a Dedicated Host isn't supported for a VM with Hibernation capability.	
OperationNotAllowed	Referencing a Capacity Reservation Group isn't supported for a VM with Hibernation capability.	
OperationNotAllowed	Hibernation can't be enabled on Virtual Machine since the OS Disk Size ({0} bytes) should at least be greater than the VM memory ({1} bytes).	Ensure the OS disk has enough space to be able to persist the RAM contents once the VM is hibernated
OperationNotAllowed	Hibernation can't be enabled on Virtual Machines created in an Availability Set.	Hibernation is only supported for standalone VMs & Virtual Machine Scale Sets Flex VMs

Unable to hibernate a VM

If you're unable to hibernate a VM, first check whether hibernation is enabled on the VM. For example, using the GET VM API, you can check if hibernation is enabled on the VM

```
"properties": {
    "vmId": "XXX",
    "hardwareProfile": {
        "vmSize": "Standard_D4s_v5"
    },
    "additionalCapabilities": {
        "hibernationEnabled": true
    },
```

If hibernation is enabled on the VM, check if hibernation is successfully enabled in the guest OS.

For Linux guests, check out the Linux VM hibernation troubleshooting guide.

For Windows guests, check out the Windows VM hibernation troubleshooting guide.

Common error codes

Expand table

ResultCode	errorDetails	Action
InternalOperationError	The fabric operation failed.	This is usually a transient issue. Retry the Hibernate operation after 5mins.
OperationNotAllowed	Operation 'HibernateAndDeallocate' isn't allowed on VM 'Z0000ZYH000' since VM has extension 'AzureHibernateExtension' in failed state	Customer issue. Confirm that VM creation with hibernation enabled succeeded, and that the extension is in a healthy state
OperationNotAllowed	The Hibernate-Deallocate Operation can only be triggered on a VM that is successfully provisioned and is running.	Customer error. Ensure that the VM is successfully running before attempting to Hibernate-Deallocate the VM.
OperationNotAllowed	The Hibernate-Deallocate Operation can only be triggered on a VM that is enabled for hibernation. Enable the property additionalCapabilities.hibernationEnabled during VM creation, or after stopping and deallocating the VM.	Customer error.
VMHibernateFailed	Hibernating the VM 'hiber_vm_res_5' failed due to an internal error. Retry later.	Retry after 5mins. If it continues to fail after multiple retries, check if the guest is correctly configured to support hibernation or contact Azure support.

ResultCode	errorDetails	Action
VMHibernateNotSupported	The VM 'Z0000ZYJ000' doesn't support hibernation. Ensure that the VM is correctly configured to support hibernation.	Hibernating a VM immediately after boot isn't supported. Retry hibernating the VM after a few minutes.

Unable to resume a VM

Starting a hibernated VM is similar to starting a stopped VM. In addition to commonly seen issues while starting VMs, certain issues are specific to starting a hibernated VM.

Expand table

ResultCode	errorDetails
OverconstrainedResumeFromHibernatedStateAllocationRequest	Allocation failed. VM(s) with the following constraints can't be allocated, because the condition is too restrictive. Remove some constraints and try again. Constraints applied are: Networking Constraints (such as Accelerated Networking or IPv6), Resuming from hibernated state (retry starting the VM after some time or alternatively stop-deallocate the VM and try starting the VM again).
AllocationFailed	VM allocation failed from hibernated state due to insufficient capacity. Try again later or alternatively stop-deallocate the VM and try starting the VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshooting hibernation on Linux VMs

Article • 08/22/2024

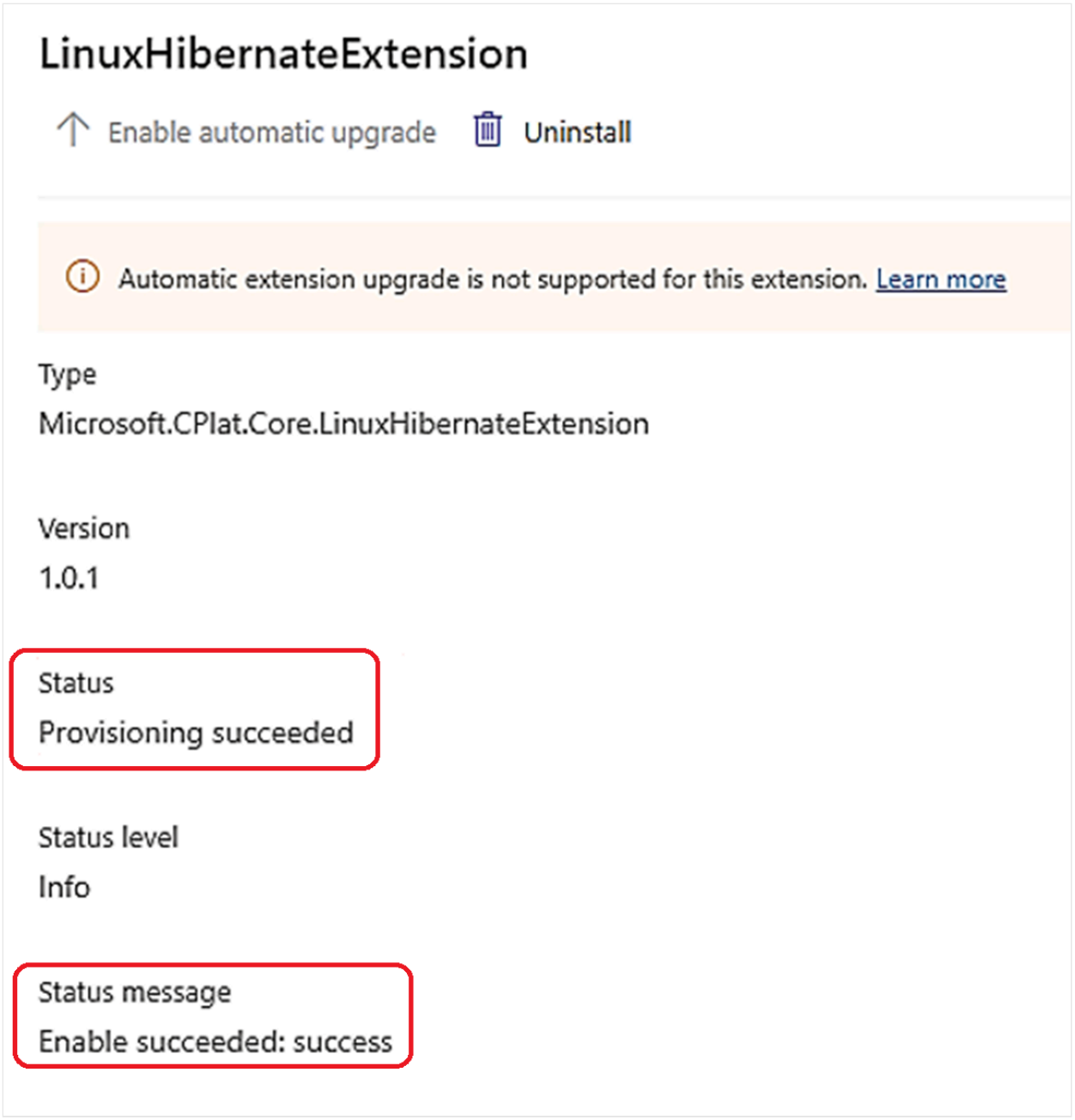
Hibernating a virtual machine allows you to persist the VM state to the OS disk. This article describes how to troubleshoot issues with the hibernation feature on Linux, issues creating hibernation enabled Linux VMs, and issues with hibernating a Linux VM.

To view the general troubleshooting guide for hibernation, check out Troubleshoot hibernation in Azure.

Unable to hibernate a Linux VM

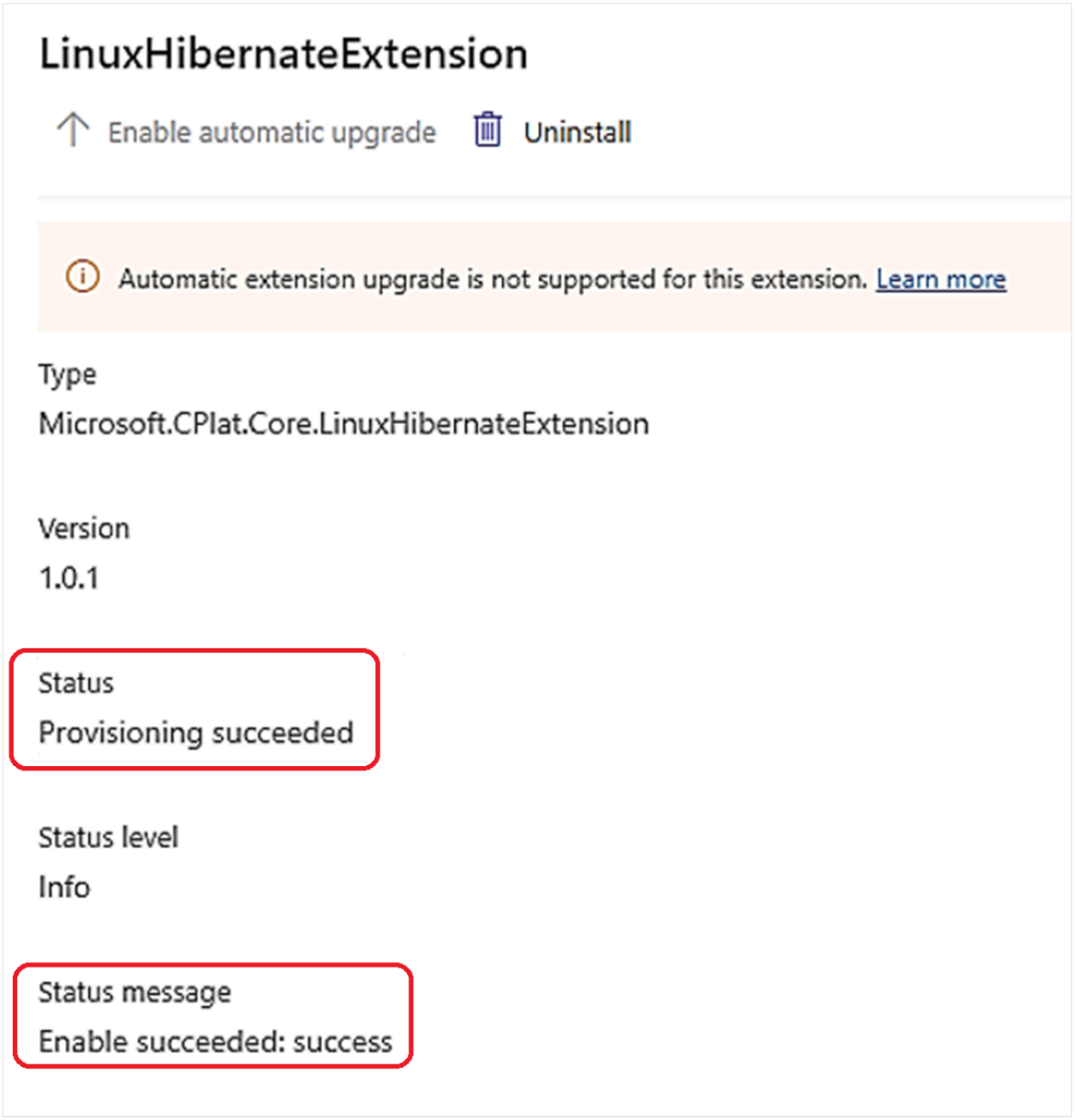
If you're unable to hibernate a VM, first check whether hibernation is enabled on the VM.

If hibernation is enabled on the VM, check if hibernation is successfully enabled in the guest OS. You can check the extension status if you used the extension to enable hibernation in the guest OS.



Guest Linux VMs unable to hibernate

You can check the extension status if you used the extension to enable hibernation in the guest OS.



If you used the hibernation-setup-tool to configure the guest for hibernation, you can check if the tool executed successfully through this command:

```
systemctl status hibernation-setup-tool
```

A successful status should return "Inactive (dead)", and the log messages should say "Swap file for VM hibernation set up successfully"

Example:

```
azureuser@:~$ systemctl status hibernation-setup-tool
● hibernation-setup-tool.service - Hibernation Setup Tool
```

```
   Loaded: loaded (/lib/systemd/system/hibernation-setup-tool.service;
enabled; vendor preset: enabled)
   Active: inactive (dead) since Wed 2021-08-25 22:44:29 UTC; 17min ago
  Process: 1131 ExecStart=/usr/sbin/hibernation-setup-tool (code=exited,
status=0/SUCCESS)
 Main PID: 1131 (code=exited, status=0/SUCCESS)

linuxhib2 hibernation-setup-tool[1131]: INFO: update-grub2 finished
successfully.
linuxhib2 hibernation-setup-tool[1131]: INFO: udev rule to hibernate with
systemd set up in /etc/udev/rules.d/99-vm-hibernation.rules.  Telling udev
about it.
...
...
linuxhib2 hibernation-setup-tool[1131]: INFO: systemctl finished
successfully.
linuxhib2 hibernation-setup-tool[1131]: INFO: Swap file for VM hibernation
set up successfully
```

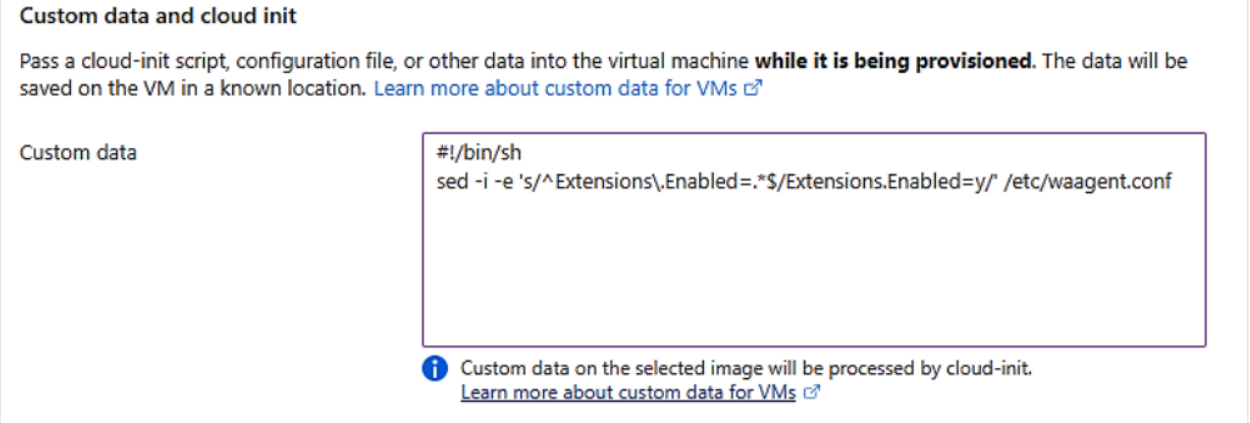
If the guest OS isn't configured for hibernation, take the appropriate action to resolve the issue. For example, if the guest failed to configure hibernation due to insufficient space, resize the OS disk to resolve the issue.

Azure extensions disabled on Debian images

Azure extensions are currently disabled by default for Debian images (more details here: https://lists.debian.org/debian-cloud/2023/07/msg00037.html). If you wish to enable hibernation for Debian based VMs through the LinuxHibernationExtension, then you can re-enable support for VM extensions via cloud-init custom data:

Bash

```
#!/bin/sh
sed -i -e 's/^Extensions\.Enabled =.* $/Extensions.Enabled=y/"
/etc/waagent.conf
```



Alternatively, you can enable hibernation on the guest by installing the hibernation-setup-tool on your Linux VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshooting hibernation on Windows VMs

Article • 08/22/2024

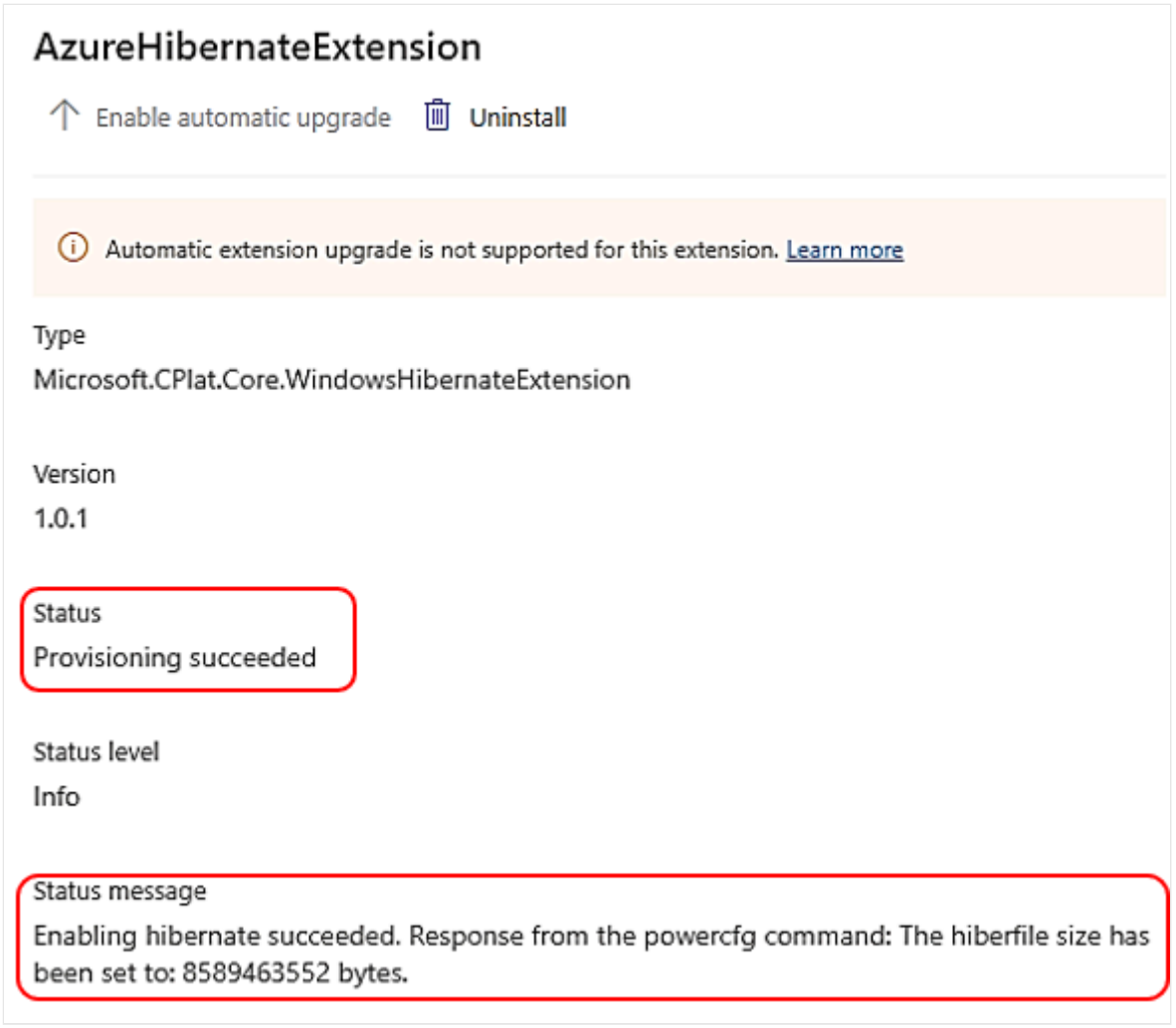
Hibernating a virtual machine allows you to persist the VM state to the OS disk. This article describes how to troubleshoot issues with the hibernation feature in Windows, issues creating hibernation enabled Windows VMs, and issues with hibernating a Windows VM.

To view the general troubleshooting guide for hibernation, check out Troubleshoot hibernation in Azure.

Unable to hibernate a Windows VM

If you're unable to hibernate a VM, first check whether hibernation is enabled on the VM.

If hibernation is enabled on the VM, check if hibernation is successfully enabled in the guest OS. You can check the status of the Hibernation extension to see if the extension was able to successfully configure the guest OS for hibernation.



The VM instance view would have the final output of the extension:

```
"extensions": [
    {
      "name": "AzureHibernateExtension",
      "type": "Microsoft.CPlat.Core.WindowsHibernateExtension",
      "typeHandlerVersion": "1.0.2",
      "statuses": [
        {
          "code": "ProvisioningState/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "Enabling hibernate succeeded. Response from the
powercfg command: \tThe hiberfile size has been set to: 17178693632
bytes.\r\n"
        }
      ]
    },
```

Additionally, confirm that hibernate is enabled as a sleep state inside the guest. The expected output for the guest should look like this.

```
C:\Users\vmadmin>powercfg /a
    The following sleep states are available on this system:
        Hibernate
        Fast Startup

    The following sleep states are not available on this system:
        Standby (S1)
            The system firmware does not support this standby state.

        Standby (S2)
            The system firmware does not support this standby state.

        Standby (S3)
            The system firmware does not support this standby state.

        Standby (S0 Low Power Idle)
            The system firmware does not support this standby state.

        Hybrid Sleep
            Standby (S3) isn't available.
```

If Hibernate isn't listed as a supported sleep state, there should be a reason associated with it, which should help determine why hibernate isn't supported. This occurs if guest hibernate isn't configured for the VM.

```
C:\Users\vmadmin>powercfg /a
    The following sleep states are not available on this system:
        Standby (S1)
            The system firmware does not support this standby state.

        Standby (S2)
            The system firmware does not support this standby state.

        Standby (S3)
            The system firmware does not support this standby state.

        Hibernate
            Hibernation hasn't been enabled.

        Standby (S0 Low Power Idle)
            The system firmware does not support this standby state.

        Hybrid Sleep
        Standby (S3) is not available.
            Hibernation is not available.

        Fast Startup
```

```
            Hibernation is not available.
```

If the extension or the guest sleep state reports an error, you'd need to update the guest configurations as per the error descriptions to resolve the issue. After fixing all the issues, you can validate that hibernation has been enabled successfully inside the guest by running the 'powercfg /a' command - which should return Hibernate as one of the sleep states. Also validate that the AzureHibernateExtension returns to a Succeeded state. If the extension is still in a failed state, then update the extension state by triggering reapply VM API

> **Note**
>
> If the extension remains in a failed state, you can't hibernate the VM.

Commonly seen issues where the extension fails.

Expand table

Issue	Action
Page file is in temp disk. Move it to OS disk to enable hibernation.	Move page file to the C: drive and trigger reapply on the VM to rerun the extension
Windows failed to configure hibernation due to insufficient space for the hiberfile	Ensure that C: drive has sufficient space. You can try expanding your OS disk, your C: partition size to overcome this issue. Once you have sufficient space, trigger the Reapply operation so that the extension can retry enabling hibernation in the guest and succeeds.
Extension error message: "A device attached to the system isn't functioning"	Ensure that C: drive has sufficient space. You can try expanding your OS disk, your C: partition size to overcome this issue. Once you have sufficient space, trigger the Reapply operation so that the extension can retry enabling hibernation in the guest and succeeds.
Hibernation is no longer supported after Virtualization Based Security (VBS) was enabled inside the guest	Enable Virtualization in the guest to get VBS capabilities along with the ability to hibernate the guest. Enable virtualization in the guest OS.

Issue	Action
Enabling hibernate failed. Response from the powercfg command. Exit Code: 1. Error message: Hibernation failed with the following error: The request isn't supported. The following items are preventing hibernation on this system. The current Device Guard configuration disables hibernation. An internal system component disabled hibernation. Hypervisor	Enable Virtualization in the guest to get VBS capabilities along with the ability to hibernate the guest. To enable virtualization in the guest, refer to this document

Guest Windows VMs unable to hibernate

If a hibernate operation succeeds, the following events are seen in the guest:

```
Guest responds to the hibernate operation (note that the following event is
logged on the guest on resume)

    Log Name:      System
    Source:        Kernel-Power
    Event ID:      42
    Level:         Information
    Description:
    The system is entering sleep
```

If the guest fails to hibernate, then all or some of these events are missing. Commonly seen issues:

Expand table

Issue	Action
Guest fails to hibernate because Hyper-V Guest Shutdown Service is disabled. You can check this by running *sc query vmicshutdown*.	Ensure that Hyper-V Guest Shutdown Service isn't disabled. Enabling this service should resolve the issue.
Guest fails to hibernate because Power Service is disabled. You can check this by running *sc query power*.	Ensure that Power Service isn't disabled. Enabling this service should resolve the issue.
Guest fails to hibernate because HVCI (Memory integrity) is enabled.	If Memory Integrity is enabled in the guest and you're trying to hibernate the VM, then ensure your guest is running the minimum OS build required to support

Issue	Action
	hibernation with Memory Integrity. Win 11 22H2 – Minimum OS Build - 22621.2134 Win 11 21H1 - Minimum OS Build - 22000.2295 Win 10 22H2 - Minimum OS Build - 19045.3324

Logs needed for troubleshooting:

If you encounter an issue outside of these known scenarios, the following logs can help Azure troubleshoot the issue:

- Relevant event logs on the guest: Microsoft-Windows-Kernel-Power, Microsoft-Windows-Kernel-General, Microsoft-Windows-Kernel-Boot.
- During a bug check, a guest crash dump is helpful.

Unable to resume a Windows VM

When you start a VM from a hibernated state, you can use the VM instance view to get more details on whether the guest successfully resumed from its previous hibernated state or if it failed to resume and instead did a cold boot.

VM instance view output when the guest successfully resumes:


```
{
  "computerName": "myVM",
  "osName": "Windows 11 Enterprise",
  "osVersion": "10.0.22000.1817",
  "vmAgent": {
    "vmAgentVersion": "2.7.41491.1083",
    "statuses": [
      {
        "code": "ProvisioningState/succeeded",
        "level": "Info",
        "displayStatus": "Ready",
        "message": "GuestAgent is running and processing the extensions.",
        "time": "2023-04-25T04:41:17.296+00:00"
      }
    ],
    "extensionHandlers": [
      {
        "type": "Microsoft.CPlat.Core.RunCommandWindows",
        "typeHandlerVersion": "1.1.15",
        "status": {
          "code": "ProvisioningState/succeeded",
          "level": "Info",
```

```
          "displayStatus": "Ready"
        }
      },
      {
        "type": "Microsoft.CPlat.Core.WindowsHibernateExtension",
        "typeHandlerVersion": "1.0.3",
        "status": {
          "code": "ProvisioningState/succeeded",
          "level": "Info",
          "displayStatus": "Ready"
        }
      }
    ]
  },
  "extensions": [
    {
      "name": "AzureHibernateExtension",
      "type": "Microsoft.CPlat.Core.WindowsHibernateExtension",
      "typeHandlerVersion": "1.0.3",
      "substatuses": [
        {
          "code": "ComponentStatus/VMBootState/Resume/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "Last guest resume was successful."
        }
      ],
      "statuses": [
        {
          "code": "ProvisioningState/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "Enabling hibernate succeeded. Response from the
powercfg command: \tThe hiberfile size has been set to: XX bytes.\r\n"
        }
      ]
    }
  ],
  "statuses": [
    {
      "code": "ProvisioningState/succeeded",
      "level": "Info",
      "displayStatus": "Provisioning succeeded",
      "time": "2023-04-25T04:41:17.8996086+00:00"
    },
    {
      "code": "PowerState/running",
      "level": "Info",
      "displayStatus": "VM running"
    }
  ]
}
```

If the Windows guest fails to resume from its previous state and cold boots, then the VM instance view response is:


```
  "extensions": [
    {
      "name": "AzureHibernateExtension",
      "type": "Microsoft.CPlat.Core.WindowsHibernateExtension",
      "typeHandlerVersion": "1.0.3",
      "substatuses": [
        {
          "code": "ComponentStatus/VMBootState/Start/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "VM booted."
        }
      ],
      "statuses": [
        {
          "code": "ProvisioningState/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "Enabling hibernate succeeded. Response from the
powercfg command: \tThe hiberfile size has been set to: XX bytes.\r\n"
        }
      ]
    }
  ],
  "statuses": [
    {
      "code": "ProvisioningState/succeeded",
      "level": "Info",
      "displayStatus": "Provisioning succeeded",
      "time": "2023-04-19T17:18:18.7774088+00:00"
    },
    {
      "code": "PowerState/running",
      "level": "Info",
      "displayStatus": "VM running"
    }
  ]
}
```


Windows guest events while resuming

If a guest successfully resumes, the following guest events are available:

```
Log Name:      System
    Source:        Kernel-Power
    Event ID:      107
    Level:         Information
    Description:
    The system has resumed from sleep.
```

If the guest fails to resume, all or some of these events are missing. To troubleshoot why the guest failed to resume, the following logs are needed:

- Event logs on the guest: Microsoft-Windows-Kernel-Power, Microsoft-Windows-Kernel-General, Microsoft-Windows-Kernel-Boot.
- On bugcheck, a guest crash dump is needed.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

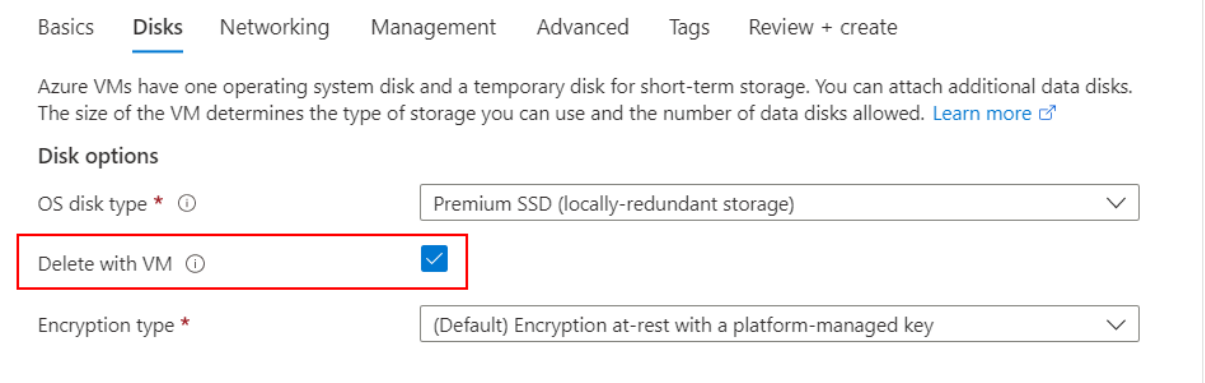
Delete a VM and attached resources

Depending on how you delete a VM, it may only delete the VM resource, not the networking and disk resources. You can change the default settings for what other resources are deleted when you delete a VM.

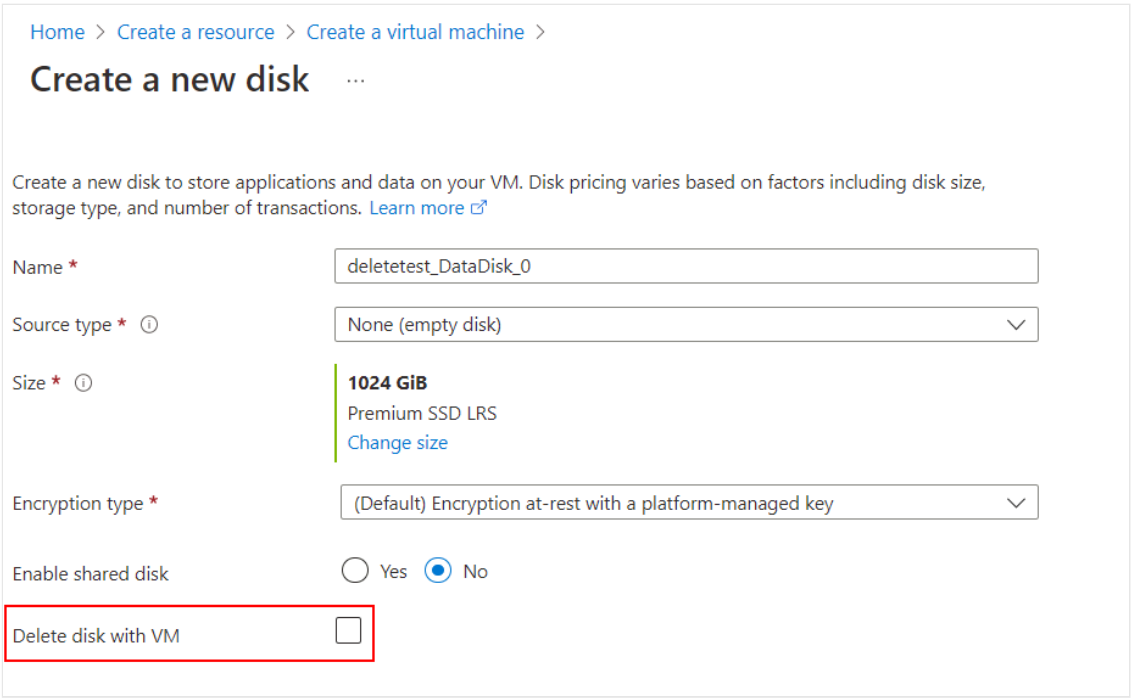
Set delete options when creating a VM

Portal

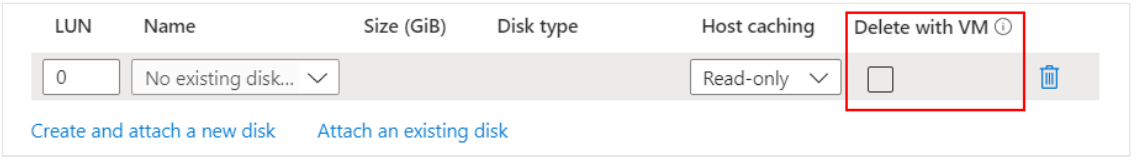
1. Open the portal.
2. Select + **Create a resource**.
3. On the **Create a resource** page, under **Virtual machines**, select **Create**.
4. Make your choices on the **Basics**, then select **Next : Disks >** to open the **Disks** tab.
5. Under **Disk options**, by default the OS disk is set to **Delete with VM**. If you don't want to delete the OS disk, clear the checkbox. If you're using an existing OS disk, the default is to detach the OS disk when the VM is deleted.



6. Under **Data disks**, you can either attach an existing data disk or create a new disk and attach it to the VM.
    - If you choose **Create and attach a new disk**, the **Create a new disk** page opens and you can select whether to delete the disk when you delete the VM.

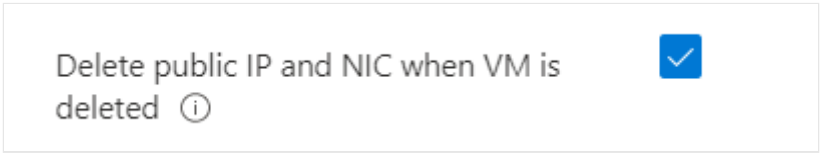


- If you choose to **Attach an existing disk**, can choose the disk, LUN, and whether you want to delete the data disk when you delete the VM.



7. When you're done adding your disk information, select **Next : Networking** > to open the **Networking** tab.

8. Towards the bottom of the page, select **Delete public IP and NIC when VM is deleted**.



9. When you're done making selections, select **Review + create**.

10. You can verify which resources you have chosen to delete when you delete the VM.

11. When you're satisfied with your selections, and validation passes, select **Create** to deploy the VM.

Update the delete behavior on an existing VM

You can change the behavior when you delete a VM.

CLI

The following example uses the Azure CLI to set the delete option to `detach` so you can reuse the disk.

Make sure to login to correct subscription using `az account set --subscription "SubscriptionIDName"`

Azure CLI

```
az resource update --resource-group myResourceGroup --name myVM --resource-type
virtualMachines --namespace Microsoft.Compute --set
properties.storageProfile.osDisk.deleteOption=detach
```

Force Delete for VMs

Force delete allows you to forcefully delete your virtual machine, reducing delete latency and immediately freeing up attached resources. For VMs that don't require graceful shutdown, Force Delete will delete the VM as fast as possible while relieving the logical resources from the VM, bypassing the graceful shutdown and some of the cleanup operations. Force Delete won't immediately free the MAC address associated with a VM, as this is a physical resource that may take up to 10 min to free. If you need to immediately reuse the MAC address on a new VM, Force Delete isn't recommended. Force delete should only be used when you aren't intending to reuse virtual hard disks. You can use force delete through Portal, CLI, PowerShell, and REST API.

Portal

When you go to delete an existing VM, you'll find an option to apply force delete in the delete pane.

1. Open the portal.
2. Navigate to your virtual machine.
3. On the **Overview** page, select **Delete**.
4. In the **Delete virtual machine** pane, select the checkbox for **Apply force delete**.
5. Select **Ok**.

Force Delete for scale sets

Force delete allows you to forcefully delete your Virtual Machine Scale Set, reducing delete latency and immediately freeing up attached resources. Force Delete won't immediately free the MAC address associated with a VM, as this is a physical resource that may take up to 10 min to free. If you need to immediately reuse the MAC address on a new VM, Force Delete is not recommended. Force delete should only be used when you are not intending to reuse virtual hard disks. You can use force delete through Portal, CLI, PowerShell, and REST API.

Portal

When you go to delete an existing scale set, you'll find an option to apply force delete in the delete pane.

1. Open the portal.
2. Navigate to your Virtual Machine Scale Set.
3. On the **Overview** page, select **Delete**.
4. In the **Delete Virtual Machine Scale Set** pane, select the checkbox for **Apply force delete**.
5. Select **Ok**.

FAQ

Q: Does this feature work with shared disks?

A: For shared disks, you can't set the 'deleteOption' property to 'Delete'. You can leave it blank or set it to 'Detach'

Q: Which Azure resources support this feature?

A: This feature is supported on all managed disk types used as OS disks and Data disks, NICs, and Public IPs

Q: What happens to the data on the disks when a VM is deleted?

A: When a VM is deleted, the data on the disks is not automatically deleted. The OS disk and any attached data disks remain in your resource group and continue to incur charges. You need to manually delete these disks if you no longer need them.

Q: What happens to VM backups when I delete the VM?

A: Azure Backup retains VM backups according to the backup policy, even after the VM is deleted. Retained backups continue to incur storage charges until they are deleted. To avoid unnecessary costs, review your backup retention and delete backups you no longer need.

To manage or delete VM backups, go to the Azure portal, navigate to your **Recovery Services** vault, select **Backup items**, then select **Azure Virtual Machine**, and choose the VM whose backups you want to manage. From there, you can stop backup and delete backup data as needed.

Q: Can I use this feature on disks and NICs that aren't associated with a VM?

A: No, this feature is only available on disks and NICs associated with a VM.

Q: How does this feature work with Flexible Virtual Machine Scale Set?

A: For Flexible Virtual Machine Scale Set the disks, NICs, and PublicIPs have `deleteOption` set to `Delete` by default so these resources are automatically cleaned up when the VMs are deleted.

For data disks that were explicitly created and attached to the VMs, you can modify this property to 'Detach' instead of 'Delete' if you want the disks to persist after the VM is deleted.

Q: Do Spot VMs support this feature?

A: Yes, you can use this feature for Spot VMs just the way you would for on-demand VMs.

Q: How do I persist the disks, NIC, and Public IPs associated with a VM?

A: By default, disks, NICs, and Public IPs associated with a VM are persisted when the VM is deleted. If you configure these resources to be automatically deleted, you can update the settings so that the resources remain after the VM is deleted. To keep these resources, set the `deleteOption` property to `Detach`.

Next steps

To learn more about basic VM management, see Tutorial: Create and Manage Linux VMs with the Azure CLI.

Last updated on 02/06/2026

Connect to a Linux VM

09/11/2025

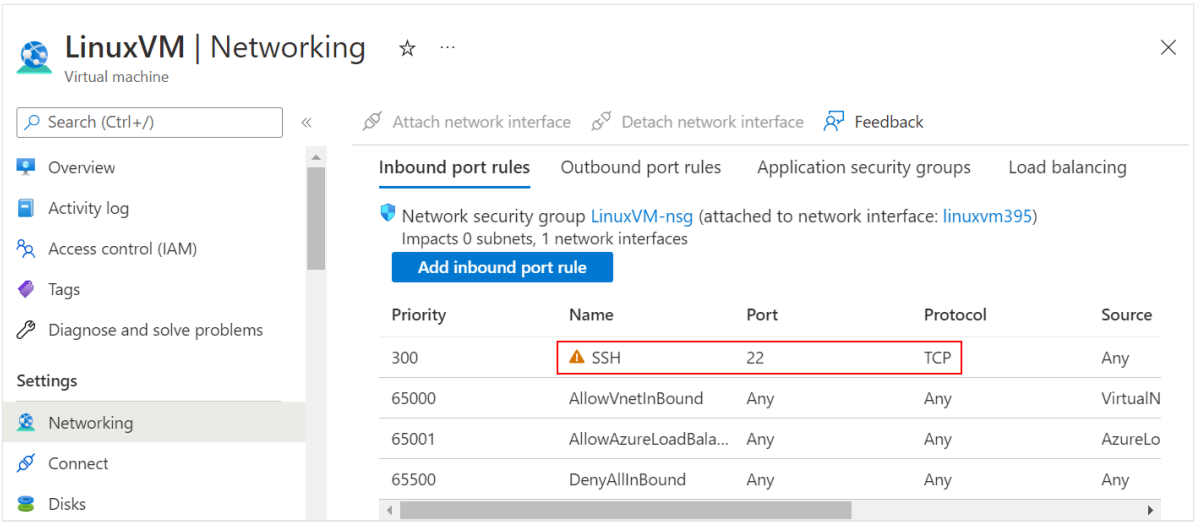
When hosting a Linux virtual machine on Azure, the most common method for accessing that VM is through the Secure Shell Protocol (SSH). Any standard SSH client commonly found in Linux and Windows allows you to connect. You can also use Azure Cloud Shell from any browser.

This document describes how to connect, via SSH, to a VM that has a public IP. If you need to connect to a VM without a public IP, see Azure Bastion Service.

Prerequisites

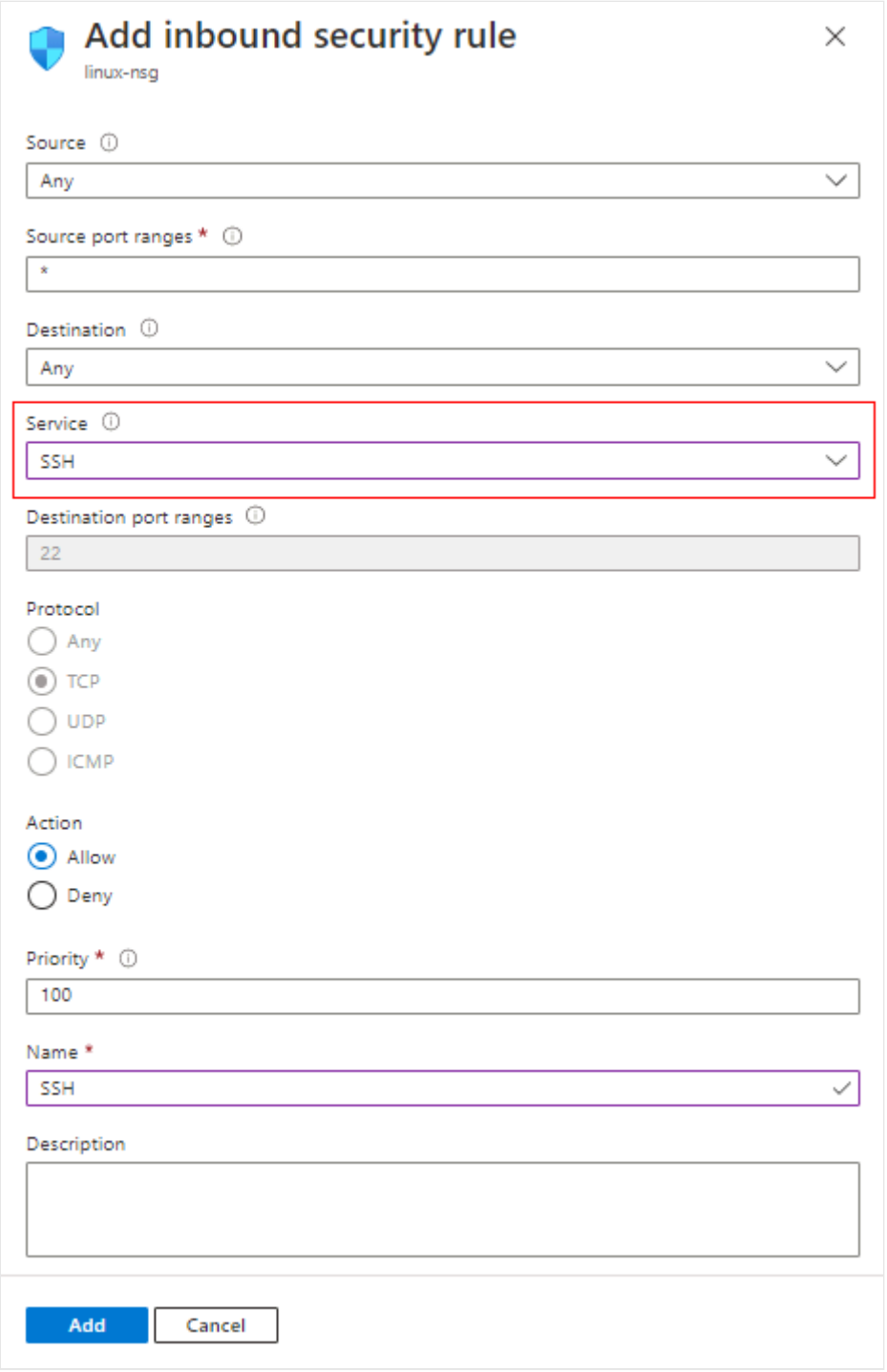
- You need an SSH key pair. If you don't already have one, Azure creates a key pair during the deployment process. If you need help with creating one manually, see Create and use an SSH public-private key pair for Linux VMs in Azure.
- You need an existing Network Security Group (NSG). Most VMs have an NSG by default, but if you don't already have one you can create one and attach it manually. For more information, see Create, change, or delete a network security group.
- To connect to a Linux VM, you need the appropriate port open. Typically SSH uses port 22. The following instructions assume port 22 but the process is the same for other port numbers. You can validate an appropriate port is open for SSH using the troubleshooter or by checking manually in your VM settings. To check if port 22 is open:

    1. On the page for the VM, select **Networking** from the left menu.
    2. On the **Networking** page, check to see if there's a rule that allows TCP on port 22 from the IP address of the computer you're using to connect to the VM. If the rule exists, you can move to the next section.



3. If there isn't a rule, add one by selecting **Add inbound port rule**.

4. For **Service**, select **SSH** from the dropdown.



5. Edit **Priority** and **Source** if necessary

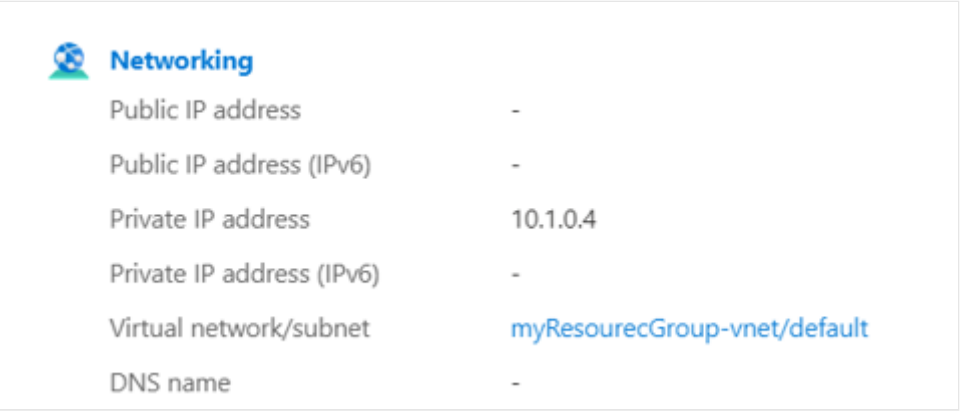
6. For **Name**, type *SSH*.

7. When you're done, select **Add**.

8. You should now have an SSH rule in the table of inbound port rules.

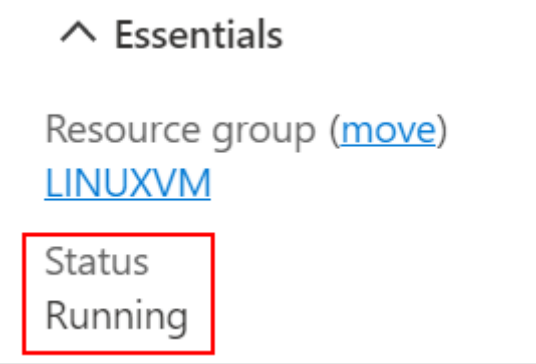
- Your VM must have a public IP address. To check if your VM has a public IP address, select **Overview** from the left menu and look at the **Networking** section. If you see an IP address next to **Public IP address**, then your VM has a public IP

  If your VM doesn't have a public IP Address, it looks like this:



To learn more about adding a public IP address to an existing VM, see Associate a public IP address to a virtual machine

- Verify your VM is running. On the Overview tab, in the **Essentials** section, verify the status of the VM is **Running**. To start the VM, select **Start** at the top of the page.



If you're having trouble connecting, you can also use portal:

1. Go to the Azure portal to connect to a VM. Search for and select **Virtual machines**.
2. Select the virtual machine from the list.
3. Select **Connect** from the left menu.
4. Select the option that fits with your preferred way of connecting. The most common connection method is Native SSH. The portal helps walk you through all prerequisites for a given connection method, and will run a VM Access check with your configured settings.

Connect to the VM

Once the above prerequisites are met, you're ready to connect to your VM. Open your SSH client of choice. The SSH client command is typically included in Linux, macOS, and Windows. If you're using Windows 7 or older, where Win32 OpenSSH isn't included by default, consider installing WSL or using Azure Cloud Shell from the browser.

> ⓘ **Note**

The following examples assume the SSH key is in the key.pem format. If you used CLI or Azure PowerShell to download your keys, they may be in the id_rsa or ED25519 format.

WSL, macOS, or native Linux client

SSH with a new key pair

1. Ensure your public and private keys are in the correct directory. The directory is usually `~/.ssh`.

   If you generated keys manually or generated them with the CLI, then the keys are probably already there. However, if you downloaded them in pem format from the Azure portal, you may need to move them to the right location. Moving the keys is done with the following syntax: `mv PRIVATE_KEY_SOURCE PRIVATE_KEY_DESTINATION`

   For example, if the key is in the `Downloads` folder, and `myKey.pem` is the name of your SSH key, type:

   ```
   Bash

   mv /Downloads/myKey.pem ~/.ssh
   ```

   > **Note**
   >
   > If you're using WSL, local files are found in the `mnt/c/` directory. Accordingly, the path to the downloads folder and SSH key would be `/mnt/c/Users/{USERNAME}/Downloads/myKey.pem`

2. Ensure you have read-only access to the private key by running

   ```
   Bash

   chmod 400 ~/.ssh/myKey.pem
   ```

3. Run the SSH command with the following syntax: `ssh -i PATH_TO_PRIVATE_KEY USERNAME@EXTERNAL_IP`

   For example, if your `azureuser` is the username you created and `20.51.230.13` is the public IP address of your VM, type:

```
Bash

ssh -i ~/.ssh/myKey.pem azureuser@20.51.230.13
```

4. Validate the returned fingerprint.

   If you have never connected to this VM before, you're asked to verify the hosts fingerprint. It's tempting to accept the fingerprint presented, but that exposes you to a potential person in the middle attack. You should always validate the hosts fingerprint. You only need to do this the first time you connect from a client. To get the host fingerprint via the portal, use the Run Command feature to execute the command:

```
Bash

ssh-keygen -lf /etc/ssh/ssh_host_ecdsa_key.pub | awk '{print $2}'
```

5. Success! You should now be connected to your VM. If you're unable to connect, see Troubleshoot SSH connections.

SSH With existing public key

1. Run the following command in your SSH client. In this example, *20.51.230.13* is the public IP Address of your VM and *azureuser* is the username you created when you created the VM.

```
Bash

ssh azureuser@20.51.230.13
```

2. Validate the returned fingerprint.

   If you have never connected to the desired VM from your current SSH client before you're asked to verify the host's fingerprint. While the default option is to accept the fingerprint presented, you're exposed to a possible "person in the middle attack". You should always validate the host's fingerprint, which only needs to be done the first time your client connects. To obtain the host fingerprint via the portal, use the Run Command feature to execute the command:

```
Bash

ssh-keygen -lf /etc/ssh/ssh_host_ecdsa_key.pub | awk '{print $2}'
```

3. Success! You should now be connected to your VM. If you're unable to connect, see our troubleshooting guide.

Password authentication

> ⚠ **Warning**
>
> This type of authentication method is not as secure as an SSH key pair and is not recommended.

1. Run the following command in your SSH client. In this example, *20.51.230.13* is the public IP Address of your VM and *azureuser* is the username you created when you created the VM.

   ```
   Bash

   ssh azureuser@20.51.230.13
   ```

   If you forgot your password or username see Reset Access to an Azure VM

2. Validate the returned fingerprint.

   If you have never connected to the desired VM from your current SSH client before you're asked to verify the host's fingerprint. While the default option is to accept the fingerprint presented, you're exposed to a possible "person in the middle attack". You should always validate the host's fingerprint, which only needs to be done the first time your client connects. To obtain the host fingerprint via the portal, use the Run Command feature to execute the command:

   ```
   Bash

   ssh-keygen -lf /etc/ssh/ssh_host_ecdsa_key.pub | awk '{print $2}'
   ```

3. Success! You should now be connected to your VM. If you're unable to connect, see Troubleshoot SSH connections.

Next steps

Learn how to transfer files to an existing VM, see Use SCP to move files to and from a VM.

Detailed steps: Create and manage SSH keys for authentication to a Linux VM in Azure

Article • 10/17/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

With a secure shell (SSH) key pair, you can create a Linux virtual machine that uses SSH keys for authentication. This article shows you how to create and use an SSH RSA public-private key file pair for SSH client connections.

If you want quick commands rather than a more in-depth explanation of SSH keys, see How to create an SSH public-private key pair for Linux VMs in Azure.

To create SSH keys and use them to connect to a Linux VM from a **Windows** computer, see How to use SSH keys with Windows on Azure. You can also use the Azure portal to create and manage SSH keys for creating VMs in the portal.

Overview of SSH and keys

SSH is an encrypted connection protocol that provides secure sign-ins over unsecured connections. Although SSH provides an encrypted connection, using passwords with SSH connections still leaves the VM vulnerable to brute-force attacks. We recommend connecting to a VM over SSH using a public-private key pair, also known as *SSH keys*.

- The *public key* is placed on your VM.
- The *private key* remains on your local system. Protect this private key. Do not share it.

When you use an SSH client to connect to your VM (which has the public key), the remote VM tests the client to make sure it has the correct private key. If the client has the private key, it's granted access to the VM.

Depending on your organization's security policies, you can reuse a single public-private key pair to access multiple Azure VMs and services. You do not need a separate pair of keys for each VM or service you wish to access.

Your public key can be shared with anyone, but only you (or your local security infrastructure) should have access to your private key.

Supported SSH key formats

Azure currently supports following key types:

- SSH protocol 2 (SSH-2) RSA (Rivest, Shamir, Adleman) with a minimum length of 2048 bits
- ED25519 Keys with a fixed length of 256 bits

Other key formats such as Elliptic-curve Diffie–Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA) are currently not supported.

SSH keys use and benefits

When you create an Azure VM by specifying the public key, Azure copies the public key (in the `.pub` format) to the `~/.ssh/authorized_keys` folder on the VM. SSH keys in `~/.ssh/authorized_keys` ensure that connecting clients present the corresponding private key during an SSH connection. In an Azure Linux VM that uses SSH keys for authentication, Azure disables the SSH server's password authentication system and only allows for SSH key authentication. By creating an Azure Linux VM with SSH keys, you can help secure the VM deployment and save yourself the typical post-deployment configuration step of disabling passwords in the `sshd_config` file.

If you do not wish to use SSH keys, you can set up your Linux VM to use password authentication. If your VM is not exposed to the Internet, using passwords may be sufficient. However, you still need to manage your passwords for each Linux VM and maintain healthy password policies and practices, such as minimum password length and regular system updates.

Generate keys with ssh-keygen

To create the keys, a preferred command is `ssh-keygen`, which is available with OpenSSH utilities in the Azure Cloud Shell, a macOS or Linux host, and Windows (10 & 11). `ssh-keygen` asks a series of questions and then writes a private key and a matching public key.

SSH keys are by default kept in the `~/.ssh` directory. If you do not have a `~/.ssh` directory, the `ssh-keygen` command creates it for you with the correct permissions. An SSH key is created as a resource and stored in Azure for later use.

> ⓘ **Note**

You can also create keys with the Azure CLI with the az sshkey create command, as described in Generate and store SSH keys.

Basic example

The following `ssh-keygen` command generates 4096-bit SSH RSA public and private key files by default in the `~/.ssh` directory. If an existing SSH key pair is found in the current location, those files are overwritten.

Bash

```
ssh-keygen -m PEM -t rsa -b 4096
```

The following `ssh-keygen` command generates 256-bit ED25519 public and private key files by default in the `~/.ssh` directory. If an existing SSH key pair is found in the current location, those files are overwritten.

Bash

```
ssh-keygen -m PEM -t ed25519
```

Detailed example

The following example shows additional command options to create an SSH RSA key pair. If an SSH key pair exists in the current location, those files are overwritten.

Bash

```
ssh-keygen \
    -m PEM \
    -t rsa \
    -b 4096 \
    -C "azureuser@myserver" \
    -f ~/.ssh/mykeys/myrsaprivatekey \
    -N mypassphrase
```

The following example shows additional command options to create an SSH ED25519 key pair. If an SSH key pair exists in the current location, those files are overwritten.

Bash

```
ssh-keygen \
    -m PEM \
```

```
    -t ed25519 \
    -C "azureuser@myserver" \
    -f ~/.ssh/mykeys/myedprivatekey \
    -N mypassphrase
```

Command explained

`ssh-keygen` = the program used to create the keys

`-m PEM` = format the key as PEM

`-t rsa` = type of key to create, in this case in the RSA format

`-b 4096` = the number of bits in the key, in this case 4096

`-C "azureuser@myserver"` = a comment appended to the end of the public key file to easily identify it. Normally an email address is used as the comment, but use whatever works best for your infrastructure.

`-f ~/.ssh/mykeys/myprivatekey` = the filename of the private key file, if you choose not to use the default name. A corresponding public key file appended with `.pub` is generated in the same directory. The directory must exist.

`-N mypassphrase` = an additional passphrase used to access the private key file.

Example of ssh-keygen (RSA)

Bash

```
ssh-keygen -t rsa -m PEM -b 4096 -C "azureuser@myserver"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/azureuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/azureuser/.ssh/id_rsa.
Your public key has been saved in /home/azureuser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:vFfHHrpSGQBd/oNdvNiX0sG9Vh+wROlZBktNZw9AUjA azureuser@myserver
The key's randomart image is:
+---[RSA 4096]----+
|        .oE=*B*+ |
|          o+o.*++|
|           .oo++*|
|       .    .B+.O|
|        S   o=BO.|
|         . .o++o |
|        . ... .  |
|         ..  .   |
```

```
|           ..    |
+----[SHA256]-----+
```

Example of ssh-keygen (ED25519)

Bash

```
ssh-keygen -t ed25519 -m PEM -C "azureuser@myserver"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/azureuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/azureuser/.ssh/id_ed25519.
Your public key has been saved in /home/azureuser/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:vFfHHrpSGQBd/oNdvNiX0sG9Vh+wROlZBktNZw9AUjA azureuser@myserver
The key's randomart image is:
+---[ED25519 256]----+
|                    |
|..  .               |
|o+.o       .        |
|*=o o   o + +       |
|*+o+   oSB + o      |
|**++o.+oo = .       |
|=+*..*.o E          |
|..  o o..           |
|     .o.            |
+----[SHA256]-----+
```

Saved key files

`Enter file in which to save the key (/home/azureuser/.ssh/id_rsa): ~/.ssh/id_rsa`

or

`Enter file in which to save the key (/home/azureuser/.ssh/id_ed25519): ~/.ssh/id_ed25519`

The default key pair names for RSA and ED25519 are `id_rsa` and `id_ed25519` respectively; some tools might expect the `id_rsa` or `id_ed25519` private key file name, so having one is a good idea. The directory `~/.ssh/` is the default location for SSH key pairs and the SSH config file. If not specified with a full path, `ssh-keygen` creates the keys in the current working directory, not the default `~/.ssh`.

List of the `~/.ssh` directory

To view existing files in the `~/.ssh` directory, run the following command. If no files are found in the directory or the directory itself is missing, make sure that all previous commands were successfully run. You may require root access to modify files in this directory on certain Linux distributions.

RSA Key pair:

```Bash
ls -al ~/.ssh
-rw------- 1 azureuser staff  1675 Aug 25 18:04 id_rsa
-rw-r--r-- 1 azureuser staff   410 Aug 25 18:04 id_rsa.pub
```

ED25519 Key pair:

```Bash
ls -al ~/.ssh
-rw------- 1 azureuser staff  1675 Aug 25 18:04 id_ed25519
-rw-r--r-- 1 azureuser staff   410 Aug 25 18:04 id_ed25519.pub
```

Key passphrase

`Enter passphrase (empty for no passphrase):`

It is *strongly* recommended to add a passphrase to your private key. Without a passphrase to protect the key file, anyone with the file can use it to sign in to any server that has the corresponding public key. Adding a passphrase offers more protection in case someone is able to gain access to your private key file, giving you time to change the keys.

Generate keys automatically during deployment

If you use the Azure CLI to create your VM, you can optionally generate both public and private SSH key files by running the az vm create command with the `--generate-ssh-keys` option. This command would default to key type of RSA, in order to generate ED25519 keys you can pass in additional flag `--ssh-key-type` command. The keys are stored in the ~/.ssh directory. Note that this command option does not overwrite keys if they already exist in that location, such as with some pre-configured Compute Gallery images.

Basic example

Create a simple Ubuntu Linux VM along with Ed25519 SSH key pair.

Azure

```
az vm create -n MyVm -g MyResourceGroup --image Ubuntu2204 --generate-ssh-keys --ssh-key-type ed25519
```

Provide SSH public key when deploying a VM

To create a Linux VM that uses SSH keys for authentication, provide your SSH public key when creating the VM using the Azure portal, CLI, Resource Manager templates, or other methods. When using the portal, you enter the public key itself. If you use the Azure CLI to create your VM with an existing public key, specify the value or location of this public key by running the az vm create command with the `--ssh-key-value` option.

If you're not familiar with the format of an SSH public key, you can see your public key by running `cat` as follows, replacing `~/.ssh/id_rsa.pub` with your own public key file location:

RSA key pair

Bash

```
cat ~/.ssh/id_rsa.pub
```

Output is similar to the following (redacted example below):

```
ssh-rsa
XXXXXXXXXXc2EAAAADAXABAAABAXC5Am7+fGZ+5zXBGgXS6GUvmsXCLGc7tX7/rViXk3+eShZzaX
nt75gUmT1I2f75zFn2hlAIDGKWf4g12KWcZxy81TniUOTjUsVlwPymXUXxESL/UfJKfbdstBhTOd
y5EG9rYWA0K43SJmwPhH28BpoLfXXXXXG+/ilsXXXXXKgRLiJ2W19MzXHp8z3Lxw7r9wx3HaVlP4
XiFv9U4hGcp8RMI1MP1nNesFlOBpG4pV2bJRBTXNXeY4l6F8WZ3C4kuf8XxOo08mXaTpvZ3T1841
altmNTZCcPkXuMrBjYSJbA8npoXAXNwiivyoe3X2KMXXXXXdXXXXXXXXXXCXXXXX/
azureuser@myserver
```

If you copy and paste the contents of the public key file into the Azure portal or a Resource Manager template, make sure you don't copy any additional whitespace or introduce additional line breaks. For example, if you use macOS, you can pipe the public

key file (by default, `~/.ssh/id_rsa.pub`) to **pbcopy** to copy the contents (there are other Linux programs that do the same thing, such as `xclip`).

If you prefer to use a public key that is in a multiline format, you can generate an RFC4716 formatted key in a 'pem' container from the public key you previously created.

To create an RFC4716 formatted key from an existing SSH public key:

Bash

```
ssh-keygen \
-f ~/.ssh/id_rsa.pub \
-e \
-m RFC4716 > ~/.ssh/id_ssh2.pem
```

ED25519 key pair

Bash

```
cat ~/.ssh/id_ed25519.pub
```

Output is similar to the following (redacted example below):

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIP6I5JuhGq3RidMNpxrplIQwEfc4Rh7UyV8JYYH2U2xA
azureuser@myserver
```

If you copy and paste the contents of the public key file into the Azure portal or a Resource Manager template, make sure you don't copy any additional whitespace or introduce additional line breaks. For example, if you use macOS, you can pipe the public key file (by default, `~/.ssh/id_ed25519.pub`) to **pbcopy** to copy the contents (there are other Linux programs that do the same thing, such as `xclip`).

If you prefer to use a public key that is in a multiline format, you can generate an RFC4716 formatted key in a 'pem' container from the public key you previously created.

To create a PEM formatted key from an existing SSH public key:

Bash

```
ssh-keygen \
-f ~/.ssh/id_ed25519.pub \
```

```
-e \
-m RFC4716 > ~/.ssh/id_edssh.pem
```

SSH to your VM with an SSH client

With the public key deployed on your Azure VM, and the private key on your local system, SSH to your VM using the IP address or DNS name of your VM. Replace *azureuser* and *myvm.westus.cloudapp.azure.com* in the following command with the administrator user name and the fully qualified domain name (or IP address):

Bash

```
ssh azureuser@myvm.westus.cloudapp.azure.com
```

If you provided a passphrase when you created your key pair, enter the passphrase when prompted during the sign-in process. (The server is added to your `~/.ssh/known_hosts` folder, and you won't be asked to connect again until the public key on your Azure VM changes or the server name is removed from `~/.ssh/known_hosts`.)

If the VM is using the just-in-time access policy, you need to request access before you can connect to the VM. For more information about the just-in-time policy, see Manage virtual machine access using the just in time policy.

Use ssh-agent to store your private key passphrase

To avoid typing your private key file passphrase with every SSH sign-in, you can use `ssh-agent` to cache your private key file passphrase on your local system. If you are using a Mac, the macOS Keychain securely stores the private key passphrase when you invoke `ssh-agent`.

Verify and use `ssh-agent` and `ssh-add` to inform the SSH system about the key files so that you do not need to use the passphrase interactively.

Bash

```
eval "$(ssh-agent -s)"
```

Now add the private key to `ssh-agent` using the command `ssh-add`.

Bash

```
ssh-add ~/.ssh/id_rsa
```

or

```Bash
ssh-add ~/.ssh/id_ed25519
```

The private key passphrase is now stored in `ssh-agent`.

Use ssh-copy-id to copy the key to an existing VM

If you have already created a VM, you can add a new SSH public key to your Linux VM using `ssh-copy-id`.

```Bash
ssh-copy-id -i ~/.ssh/id_rsa.pub azureuser@myserver
```

Create and configure an SSH config file

You can create and configure an SSH config file (`~/.ssh/config`) to speed up log-ins and to optimize your SSH client behavior.

The following example shows a simple configuration that you can use to quickly sign in as a user to a specific VM using the default SSH private key.

Create the file.

```Bash
touch ~/.ssh/config
```

Edit the file to add the new SSH configuration

```Bash
vim ~/.ssh/config
```

Add configuration settings appropriate for your host VM. In this example, the VM name (Host) is *myvm*, the account name (User) is *azureuser* and the IP Address or FQDN (Hostname) is 192.168.0.255.

Bash

```
# Azure Keys
Host myvm
  Hostname 192.168.0.255
  User azureuser
# ./Azure Keys
```

You can add configurations for additional hosts to enable each to use its own dedicated key pair. See SSH config file for more advanced configuration options.

Now that you have an SSH key pair and a configured SSH config file, you are able to remotely access your Linux VM quickly and securely. When you run the following command, SSH locates and loads any settings from the `Host myvm` block in the SSH config file.

Bash

```
ssh myvm
```

The first time you sign in to a server using an SSH key, the command prompts you for the passphrase for that key file.

Next steps

Next up is to create Azure Linux VMs using the new SSH public key. Azure VMs that are created with an SSH public key as the sign-in are better secured than VMs created with the default sign-in method, passwords.

- Create a Linux virtual machine with the Azure portal
- Create a Linux virtual machine with the Azure CLI
- Create a Linux VM using an Azure template

Feedback

Was this page helpful? Yes No

Generate and store SSH keys in the Azure portal

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

If you frequently use the portal to deploy Linux VMs, you can simplify using SSH keys by integrating them into Azure. There are several ways to create SSH keys for use with Azure.

- You can create SSH keys when you first create a VM. Your keys aren't tied to a specific VM and you can use them in future applications.
- You can create SSH keys in the Azure portal separate from a VM. You can use them with both new and old VMs.
- You can create SSH keys externally and upload them for use in Azure.

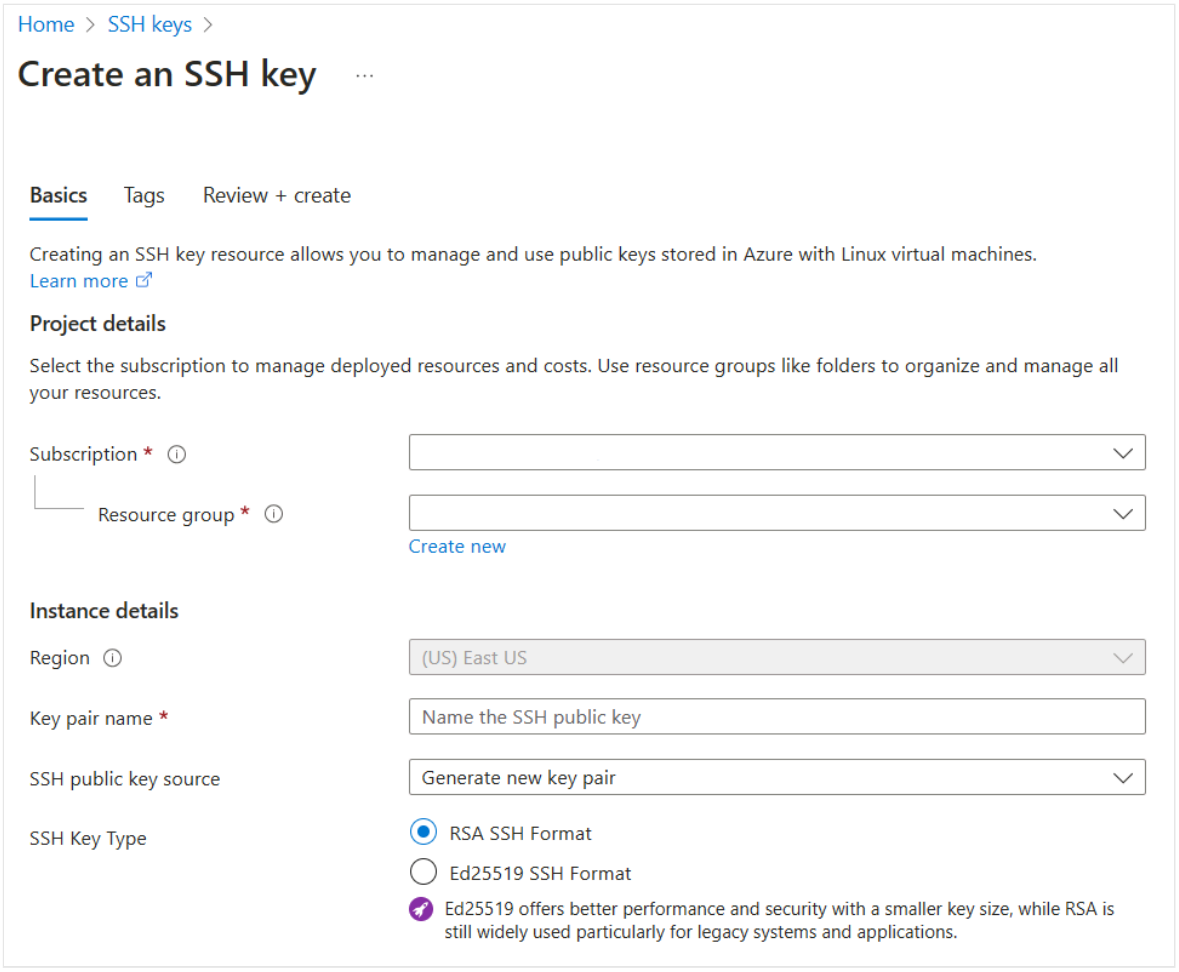
You can reuse your stored keys in various of applications to fit your organization's needs.

For more detailed information about creating and using SSH keys with Linux VMs, see Use SSH keys to connect to Linux VMs.

> **Note**
>
> ED25519 SSH key support for Linux VMs is now in preview in all regions including sovereign clouds.

Generate new keys

1. Open the Azure portal.
2. At the top of the page, type *SSH* to search. Under **Marketplace**, select **SSH keys**.
3. On the **SSH Key** page, select **Create**.



4. In **Resource group** select **Create new** to create a new resource group to store your keys. Type a name for your resource group and select **OK**.

5. In **Region** select a region to store your keys. You can use the keys in any region, this option is just the region where you store them.

6. Type a name for your key in **Key pair name**.

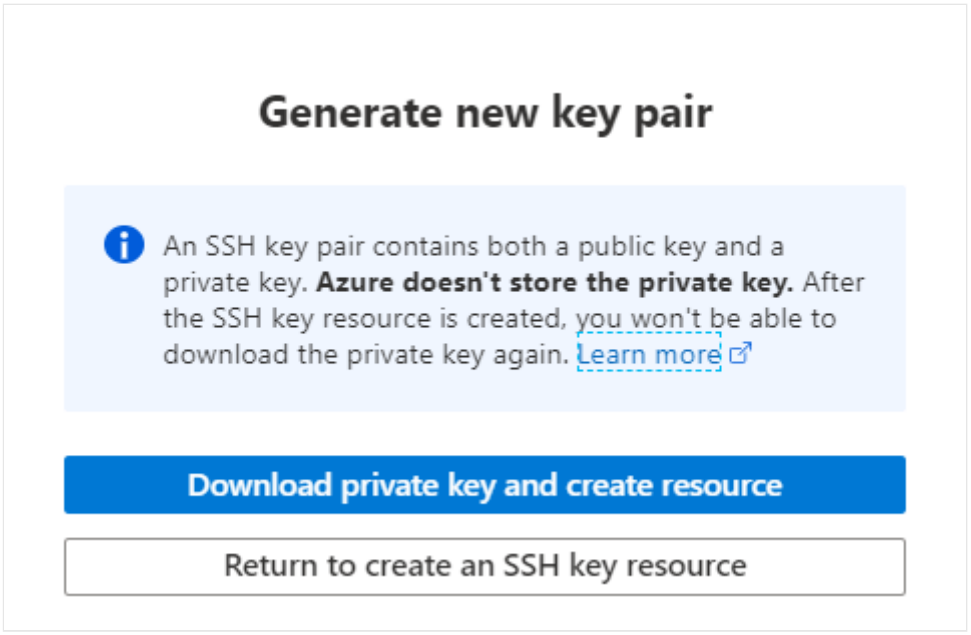
7. In **SSH public key source**, select **Generate public key source**.

8. In **SSH Key Type**, select either **RSA SSH Format** or **Ed25519 SSH Format** [Preview]

9. When you're done, select **Review + create**.

10. After it passes validation, select **Create**.

11. You'll get a pop-up window to, select **Download private key and create resource** that downloads the SSH key as a .pem file.



12. Once you've downloaded the .pem file, you might want to move it somewhere on your computer where it's easy to point to from your SSH client.

Connect to the VM

On your local computer, open a PowerShell prompt and type:

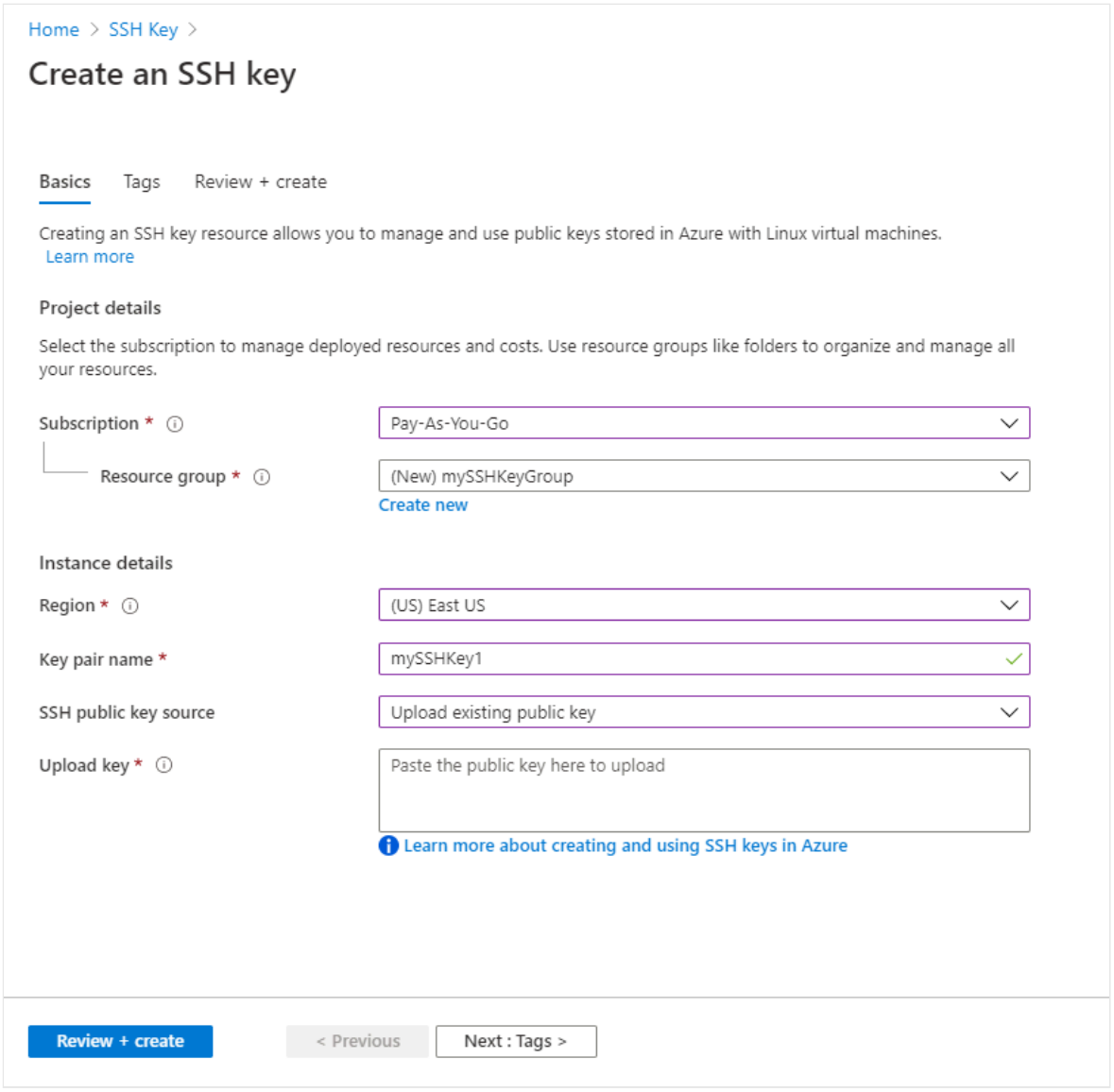
```
PowerShell
ssh -i <path to the .pem file> username@<ipaddress of the VM>
```

For example, type: `ssh -i /Downloads/mySSHKey.pem azureuser@123.45.67.890` and replace the example IP address at the end of the command with your VM's public IP address.

Upload an SSH key

You can also upload a public SSH key to store in Azure. For information about how to create an SSH key pair, see Use SSH keys to connect to Linux VMs.

1. Open the Azure portal.
2. At the top of the page, type *SSH* to search. Under **Marketplace*, select **SSH keys**.
3. On the **SSH Key** page, select **Create**.



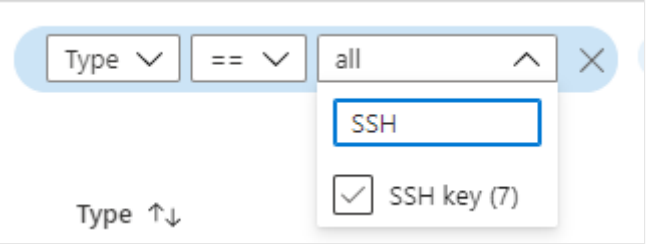
4. In **Resource group** select **Create new** to create a new resource group to store your keys. Type a name for your resource group and select **OK**.
5. In **Region** select a region to store your keys. You can use the keys in any region, this option is just the region where they're stored.
6. Type a name for your key in **Key pair name**.
7. In **SSH public key source**, select **Upload existing public key**.
8. Paste the full contents of the public key into **Upload key** and then select **Review + create**.
9. After validation completes, select **Create**.

Once you upload the key, you can choose to use it when you create a VM.

List keys

Azure stores your SSH keys created in the portal as resources, so you can filter your resources view to see all of them.

1. In the portal, select **All resource**.
2. In the filters, select **Type**, unselect the **Select all** option to clear the list.
3. Type **SSH** in the filter and select **SSH key**.



Get the public key

If you need your public key, you can easily copy it from the portal page for the key. Just list your keys (using the process in the last section) then select a key from the list. The page for your key opens and you can click the **Copy to clipboard** icon next to the key to copy it.

Next steps

To learn more about using SSH keys with Azure VMs, see Use SSH keys to connect to Linux VMs.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Generate and store SSH keys with the Azure CLI

Article • 10/17/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can create SSH keys before creating a VM and store them in Azure. Each newly created SSH key is also stored locally.

If you have existing SSH keys, you can upload and store them in Azure for reuse.

For more information, see Detailed steps: Create and manage SSH keys for authentication to a Linux VM in Azure.

For more information on how to create and use SSH keys with Linux VMs, see Use SSH keys to connect to Linux VMs.

Generate new keys

1. After you sign in, use the az sshkey create command to create the new SSH key:

   Azure CLI

   ```
   az sshkey create --name "mySSHKey" --resource-group "myResourceGroup"
   ```

   > **Note**
   >
   > This command would default to key type of RSA, in order to generate ED25519 keys you can pass in the optional flag `--encryption-type Ed25519`.

2. The resulting output lists the new key files' paths:

   Azure CLI

   ```
   Private key is saved to "/home/user/.ssh/7777777777_9999999".
   Public key is saved to "/home/user/.ssh/7777777777_9999999.pub".
   ```

3. Change the permissions for the private key file for privacy:

   Azure CLI

```
chmod 600 /home/user/.ssh/7777777777_9999999
```

Connect to the VM

On your local computer, open a Bash prompt:

Azure CLI

```
ssh -identity_file <path to the private key file> username@<ipaddress of the VM>
```

For example, enter: `ssh -i /home/user/.ssh/mySSHKey azureuser@123.45.67.890`

Upload an SSH key

You can upload a public SSH key to store in Azure.

Use the az sshkey create command to upload an SSH public key by specifying its file:

Azure CLI

```
az sshkey create --name "mySSHKey" --public-key "@/home/user/.ssh/7777777777_9999999.pub" --resource-group "myResourceGroup"
```

List keys

Use the az sshkey list command to list all public SSH keys, optionally specifying a resource group:

Azure CLI

```
az sshkey list --resource-group "myResourceGroup"
```

Get the public key

Use the az sshkey show command to show the values of a public SSH key:

Azure CLI

```
az sshkey show --name "mySSHKey" --resource-group "myResourceGroup"
```

Next steps

To learn more about how to use SSH keys with Azure VMs, see Use SSH keys to connect to Linux VMs.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quick steps: Create and use an SSH public-private key pair for Linux VMs in Azure

Article • 10/16/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

With a secure shell (SSH) key pair, you can create virtual machines (VMs) in Azure that use SSH keys for authentication. This article shows you how to quickly generate and use an SSH public-private key file pair for Linux VMs. You can complete these steps with the Azure Cloud Shell, a macOS, or a Linux host.

For help with troubleshooting issues with SSH, see Troubleshoot SSH connections to an Azure Linux VM that fails, errors out, or is refused.

> ⓘ **Note**
>
> VMs created using SSH keys are by default configured with passwords disabled, which greatly increases the difficulty of brute-force guessing attacks.

For more background and examples, see Detailed steps to create SSH key pairs.

For additional ways to generate and use SSH keys on a Windows computer, see How to use SSH keys with Windows on Azure.

Supported SSH key formats

Azure currently supports following key types:

- SSH protocol 2 (SSH-2) RSA (Rivest, Shamir, Adleman) with a minimum length of 2048 bits
- ED25519 Keys with a fixed length of 256 bits

Other key formats such as Elliptic-curve Diffie–Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA) are currently not supported.

Create an SSH key pair

Use the `ssh-keygen` command to generate SSH public and private key files. By default, these files are created in the ~/.ssh directory. You can specify a different location, and an optional password (*passphrase*) to access the private key file. If an SSH key pair with the same name exists in the given location, those files are overwritten.

The following command creates an SSH key pair using RSA encryption and a bit length of 4096:

Bash

```
ssh-keygen -m PEM -t rsa -b 4096 -f ~/.ssh/id_rsa.pem
```

The following command creates an SSH key pair using ED25519 encryption with a fixed length of 256 bits:

Bash

```
ssh-keygen -m PEM -t ed25519 -f ~/.ssh/id_ed25519.pem
```

> **Note**
>
> You can also create key pairs with the Azure CLI with the az sshkey create command, as described in Generate and store SSH keys.

If you use the Azure CLI to create your VM with the az vm create command, you can optionally generate SSH public and private key files using the `--generate-ssh-keys` option. The key files are stored in the ~/.ssh directory unless specified otherwise with the `--ssh-dest-key-path` option. If an ssh key pair already exists and the `--generate-ssh-keys` option is used, a new key pair won't be generated but instead the existing key pair will be used. In the following command, replace *VMname*, *RGname* and *UbuntuLTS* with your own values:

Azure CLI

```
az vm create --name VMname --resource-group RGname --image Ubuntu2204 --
generate-ssh-keys
```

> **Note**
>
> az sshkey create command deafults to RSA encryption and cannot be use to generate ED25519 key pairs, however you can create a ED25519 key pair using ssh-

keygen as described above and then use that public key to create a VM.

Provide an SSH public key when deploying a VM

To create a Linux VM that uses SSH keys for authentication, specify your SSH public key when creating the VM using the Azure portal, Azure CLI, Azure Resource Manager templates, or other methods:

- Create a Linux virtual machine with the Azure portal
- Create a Linux virtual machine with the Azure CLI
- Create a Linux VM using an Azure template

If you're not familiar with the format of an SSH public key, you can display your public key with the following `cat` command, replacing `~/.ssh/id_rsa.pub` with the path and filename of your own public key file if needed:

RSA key pair

Bash

```
cat ~/.ssh/id_rsa.pub
```

A typical RSA public key value looks like this example:

Output

```
ssh-rsa
AAAAB3NzaC1yc2EAABADAQABAAACAQC1/KanayNr+Q7ogR5mKnGpKWRBQU7F3Jjhn7utdf7Z2iUF
ykaYx+MInSnT3XdnBRS8KhC0IP8ptbngIaNOWd6zM8hB6UrcRTlTpwk/SuGMw1Vb40xlEFphBkVE
UgBolOoANIEXriAMvlDMZsgvnMFiQ12tD/u14cxy1WNEMAftey/vX3Fgp2vEq4zHXEliY/sFZLJU
JzcRUI0MOfHXAuCjg/qyqqbIuTDFyfg8k0JTtyGFEMQhbXKcuP2yGx1uw0ice62LRzr8w0mszftX
yMik1PnshRXbmE2xgINYg5xo/ra3mq2imwtOKJpfdtFoMiKhJmSNHBSkK7vFTeYgg0v2cQ2+vL38
lcIFX4Oh+QCzvNF/AXoDVlQtVtSqfQxRVG79Zqio5p12gHFktlfV7reCBvVIhyxc2LlYUkrq4DHz
kxNY5c9OGSHXSle9YsO3F1J5ip18f6gPq4xFmo6dVoJodZm9N0YMKCkZ4k1qJDESsJBk2ujDPmQQ
eMjJX3FnDXYYB182ZCGQzXfzlPDC29cWVgDZEXNHuYrOLmJTmYtLZ4WkdUhLLlt5XsdoKWqlWpbe
gyYtGZgeZNRtOOdN6ybOPJqmYFd2qRtb4sYPniGJDOGhx4VodXAjT09omhQJpE6wlZbRWDvKC55R
2d/CSPHJscEiuudb+1SG2uA/oik/WQ== username@domainname
```

ED25519 key pair

Bash

```
cat ~/.ssh/id_ed25519.pub
```

A typical ED25519 public key value looks like this example:

Output

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAILRjWGWLeiUQ3U9fNnCsNpXIyACpD/Jbm09OZGsz3DIM
username@domainname
```

If you copy and paste the contents of the public key file to use in the Azure portal or a Resource Manager template, make sure you don't copy any trailing whitespace. To copy a public key in macOS, you can pipe the public key file to `pbcopy`. Similarly in Linux, you can pipe the public key file to programs such as `xclip`.

The public key that you place on your Linux VM in Azure is by default stored under `~/.ssh/` directory, unless you specified a different location when you created the key pair. To use the Azure CLI 2.0 to create your VM with an existing public key, specify the value and optionally the location of this public key using the az vm create command with the `--ssh-key-values` option. In the following command, replace *myVM*, *myResourceGroup*, *UbuntuLTS*, *azureuser*, and *mysshkey.pub* with your own values:

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --ssh-key-values mysshkey.pub
```

If you want to use multiple SSH keys with your VM, you can enter them in a comma-separated list, like this `--ssh-key-values sshkey-desktop.pub, sshkey-laptop.pub`.

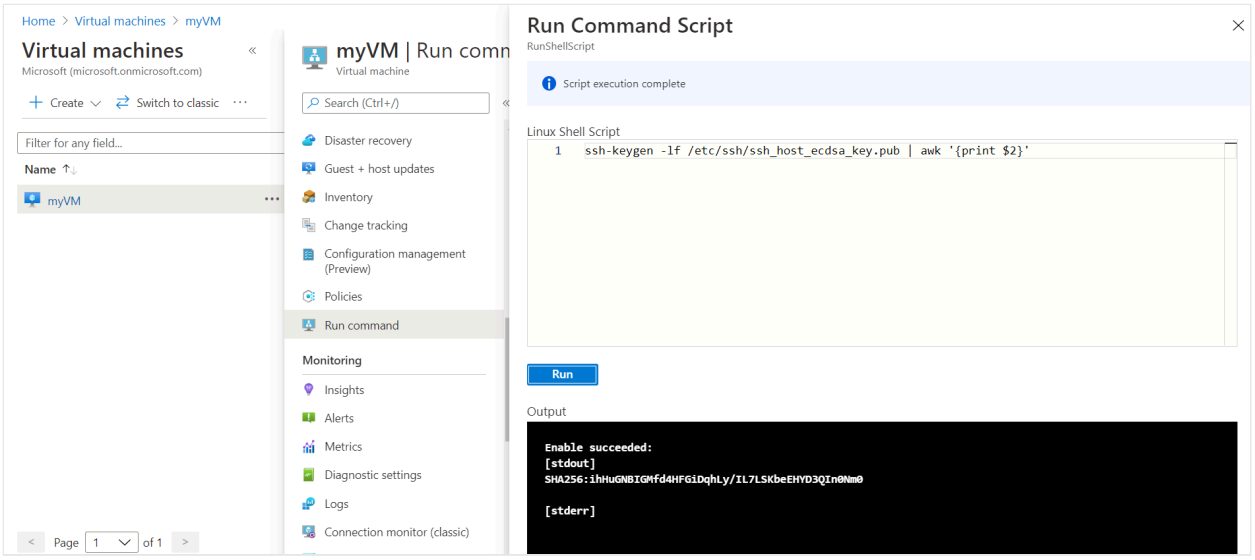
SSH into your VM

With the public key deployed on your Azure VM, and the private key on your local system, SSH into your VM using the IP address or DNS name of your VM. In the following command, replace *azureuser* and *myvm.westus.cloudapp.azure.com* with the administrator user name and the fully qualified domain name (or IP address):

Bash

```
ssh azureuser@myvm.westus.cloudapp.azure.com
```

If you're connecting to this VM for the first time, you'll be asked to verify the host's fingerprint. It's tempting to accept the fingerprint that's presented, but that approach exposes you to a possible person-in-the-middle attack. You should always validate the host's fingerprint. You need to do this only the first time you connect from a client. To obtain the host fingerprint via the portal, use the Run Command feature to execute the command `ssh-keygen -lf /etc/ssh/ssh_host_ecdsa_key.pub | awk '{print $2}'`.



To run the command using CLI, use az vm run-command invoke.

If you specified a passphrase when you created your key pair, enter that passphrase when prompted during the sign-in process. The VM is added to your ~/.ssh/known_hosts file, and you won't be asked to connect again until either the public key on your Azure VM changes or the server name is removed from ~/.ssh/known_hosts.

If the VM is using the just-in-time access policy, you need to request access before you can connect to the VM. For more information about the just-in-time policy, see Manage virtual machine access using the just in time policy.

Next steps

- For more information on working with SSH key pairs, see Detailed steps to create and manage SSH key pairs.
- If you have difficulties with SSH connections to Azure VMs, see Troubleshoot SSH connections to an Azure Linux VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to use SSH keys with Windows on Azure

Article • 10/16/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article is for Windows users who want to create and use *secure shell* (SSH) keys to connect to Linux virtual machines (VMs) in Azure. You can also generate and store SSH keys in the Azure portal to use when creating VMs in the portal.

To use SSH keys from a Linux or macOS client, see the quick steps. For a more detailed overview of SSH, see Detailed steps: Create and manage SSH keys for authentication to a Linux VM in Azure.

Overview of SSH and keys

SSH is an encrypted connection protocol that allows secure sign-ins over unsecured connections. SSH is the default connection protocol for Linux VMs hosted in Azure. Although SSH itself provides an encrypted connection, using passwords with SSH still leaves the VM vulnerable to brute-force attacks. We recommend connecting to a VM over SSH using a public-private key pair, also known as *SSH keys*.

The public-private key pair is like the lock on your front door. The lock is exposed to the **public**, anyone with the right key can open the door. The key is **private**, and only given to people you trust because it can be used to unlock the door.

- The *public key* is placed on your Linux VM when you create the VM.
- The *private key* remains on your local system. Protect this private key. Don't share it.

When you connect to your Linux VM, the VM tests the SSH client to make sure it has the correct private key. If the client has the private key, it's granted access to the VM.

Depending on your organization's security policies, you can reuse a single key pair to access multiple Azure VMs and services. You don't need a separate pair of keys for each VM.

Your public key can be shared with anyone, but only you (or your local security infrastructure) should have access to your private key.

Supported SSH key formats

Azure currently supports following key types:

- SSH protocol 2 (SSH-2) RSA (Rivest, Shamir, Adleman) with a minimum length of 2048 bits
- ED25519 Keys with a fixed length of 256 bits

Other key formats such as Elliptic-curve Diffie–Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA) are currently not supported.

SSH clients

Recent versions of Windows 10 include OpenSSH client commands to create and use SSH keys and make SSH connections from PowerShell or a command prompt.

You can also use Bash in the Azure Cloud Shell to connect to your VM. You can use Cloud Shell in a web browser, from the Azure portal, or as a terminal in Visual Studio Code using the Azure Account extension.

You can also install the Windows Subsystem for Linux to connect to your VM over SSH and use other native Linux tools within a Bash shell.

Create an SSH key pair

The easiest way to create and manage your SSH keys is to use the portal to create and store them for reuse.

You can also create key pairs with the Azure CLI with the az sshkey create command, as described in Generate and store SSH keys.

To create an SSH key pair on your local computer using the `ssh-keygen` command from PowerShell or a command prompt, use the following command:

PowerShell

```
ssh-keygen -m PEM -t rsa -b 2048
```

Enter a filename, or use the default shown in parenthesis (for example `C:\Users\username/.ssh/id_rsa`). Enter a passphrase for the file, or leave the passphrase blank if you don't want to use a passphrase.

Create a VM using your key

To create a Linux VM that uses SSH keys for authentication, provide your SSH public key when creating the VM.

Using the Azure CLI, you specify the path and filename for the public key using `az vm create` and the `--ssh-key-value` parameter.

Azure CLI

```
az vm create \
   --resource-group myResourceGroup \
   --name myVM \
   --image Ubuntu2204\
   --admin-username azureuser \
   --ssh-key-value ~/.ssh/id_rsa.pub
```

With PowerShell, use `New-AzVM` and add the SSH key to the VM configuration using`. For an example, see Quickstart: Create a Linux virtual machine in Azure with PowerShell.

If you do many deployments using the portal, you might want to upload your public key to Azure, where it can be easily selected when creating a VM from the portal. For more information, see Upload an SSH key.

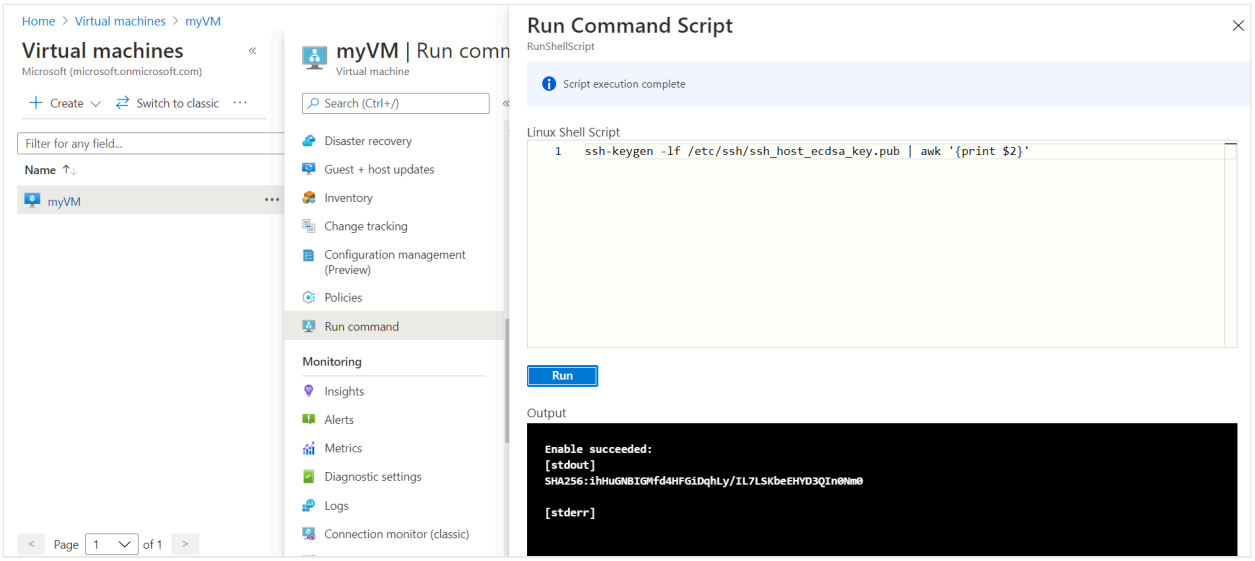
Connect to your VM

With the public key deployed on your Azure VM, and the private key on your local system, SSH to your VM using the IP address or DNS name of your VM. Replace *azureuser* and *10.111.12.123* in the following command with the administrator user name, the IP address (or fully qualified domain name), and the path to your private key:

Bash

```
ssh -i ~/.ssh/id_rsa azureuser@10.111.12.123
```

If you've never connected to this VM before you'll be asked to verify the hosts fingerprint. It's tempting to accept the fingerprint presented, however, this exposes you to a possible person in the middle attack. You should always validate the hosts fingerprint. You only need to do this on the first time you connect from a client. To obtain the host fingerprint via the portal, use the Run Command with the following command: `ssh-keygen -lf /etc/ssh/ssh_host_ecdsa_key.pub | awk '{print $2}'`.



To run the command using CLI, use the az vm run-command invoke command.

If you configured a passphrase when you created your key pair, enter the passphrase when prompted.

If the VM is using the just-in-time access policy, you need to request access before you can connect to the VM. For more information about the just-in-time policy, see Manage virtual machine access using the just in time policy.

Next steps

- For information about SSH keys in the Azure portal to use when creating VMs, see Generate and store SSH keys in the Azure portal.
- For information about SSH keys in the Azure CLI to use when creating VMs, see Generate and store SSH keys with the Azure CLI.
- For detailed steps, options, and advanced examples of working with SSH keys, see Detailed steps to create SSH key pairs.
- If you have difficulty using SSH to connect to your Linux VMs, see Troubleshoot SSH connections to an Azure Linux VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Install and configure xrdp to use Remote Desktop with Ubuntu

10/02/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When new to Linux, or for quick troubleshooting scenarios, the use of remote desktop may be easier than Secure Shell (SSH) access. This article details how to install and configure a desktop environment (xfce) and remote desktop (xrdp) for your Linux VM running Ubuntu.

> **Note**
>
> Using Remote Desktop over the internet will introduce noticeable "lag" (input latency) when compared to local desktop use. This can be influenced by multiple factors including local internet speed and distance from the datacenter where the virtual machine is hosted. This lag does not usually reflect the performance of the VM itself.

> **Important**
>
> Starting with Red Hat 9, xrdp is not supported anymore since it requires third party repositories, Red Hat has specific steps on how to enable this kind of access described here

Prerequisites

This article and steps provided were taken from testing Ubuntu 24.04 and SUSE Enterprise Linux 15.

If you need to create a VM, use one of the following methods:

- The Azure CLI
- The Azure portal

Install a desktop environment on your Linux VM

Most Linux VMs in Azure don't have a desktop environment installed by default. Linux VMs are commonly managed using SSH connections rather than a desktop environment, however there are several desktop environments that you can choose to install. Depending on your choice of

desktop environment, it consumes up to 2 GB of disk space and take up to ten minutes to both install and configure all the required packages.

The following example installs the lightweight xfce4 desktop environment on an Ubuntu VM.

First, SSH to your VM. The following example connects to the VM named *myvm.westus.cloudapp.azure.com* with the username of *azureuser*. Use your own values:

```bash
ssh azureuser@myvm.westus.cloudapp.azure.com
```

If you're using Windows and need more information on using SSH, see How to use SSH keys with Windows.

For Ubuntu 24.04

Install all the required packages using `apt` :

```bash
sudo apt update
sudo apt install xfce4 xfce4-session xrdp
sudo systemctl enable xrdp
```

For SUSE Enterprise Linux 15

Install all the required packages using `zypper` :

```bash
sudo zypper in xorg-x11-server xterm xorg-x11-xauth xrdp ssl-cert xhost xclock
sudo systemctl enable xrdp
```

On Ubuntu 24, you need to give certificate access to an xrdp user:

```bash
sudo adduser xrdp ssl-cert
```

The steps below apply to both distributions (Ubuntu 24 and SUSE Enterprise Linux 15)

Tell xrdp what desktop environment to use when you start your session. Configure xrdp to use xfce as your desktop environment as follows:

```
Bash

echo xfce4-session >~/.xsession
```

Restart the xrdp service for the changes to take effect as follows:

```
Bash

sudo systemctl restart xrdp
```

Set a local user account password

If you created a password for your user account when you created your VM, skip this step. If you only use SSH key authentication and don't have a local account password set, specify a password before you use xrdp to log in to your VM. xrdp can't accept SSH keys for authentication. The following example specifies a password for the user account *azureuser*:

```
Bash

sudo passwd azureuser
```

> **Note**
>
> Specifying a password does not update your SSHD configuration to permit password logins if it currently does not. From a security perspective, you may wish to connect to your VM with an SSH tunnel using key-based authentication and then connect to xrdp. If so, skip the following step on creating a network security group rule to allow remote desktop traffic.

Create a Network Security Group rule for Remote Desktop traffic

To allow Remote Desktop traffic to reach your Linux VM, a network security group rule needs to be created that allows TCP on port 3389 to reach your VM. For more information about network security group rules, see What is a network security group? You can also use the Azure portal to create a network security group rule.

Azure CLI

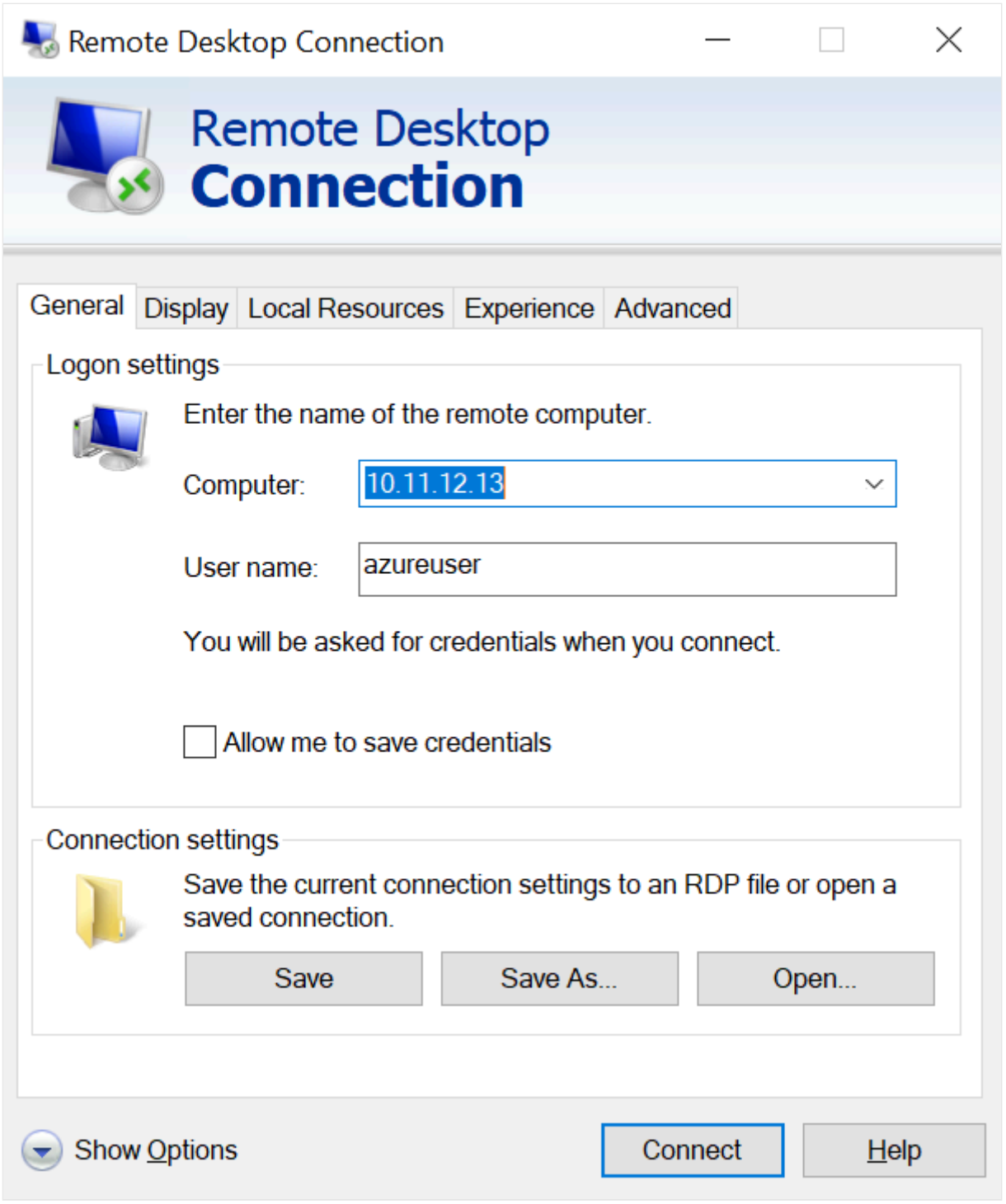
The following example creates a network security group rule with az vm open-port on port *3389*. From the Azure CLI, not the SSH session to your VM, open the following network security group rule:

Azure CLI

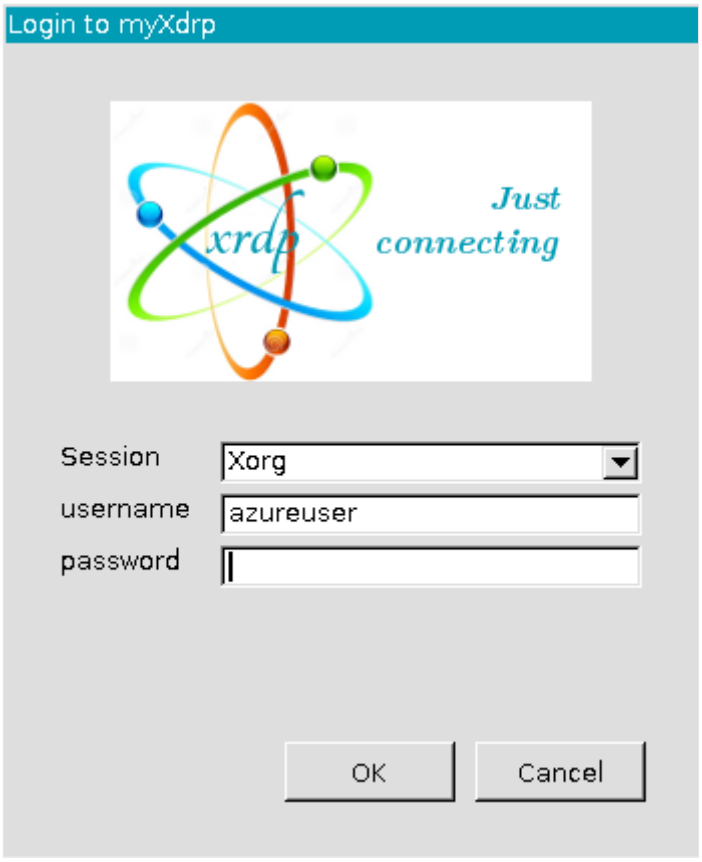
```
az vm open-port --resource-group myResourceGroup --name myVM --port 3389
```

Connect your Linux VM with a Remote Desktop client

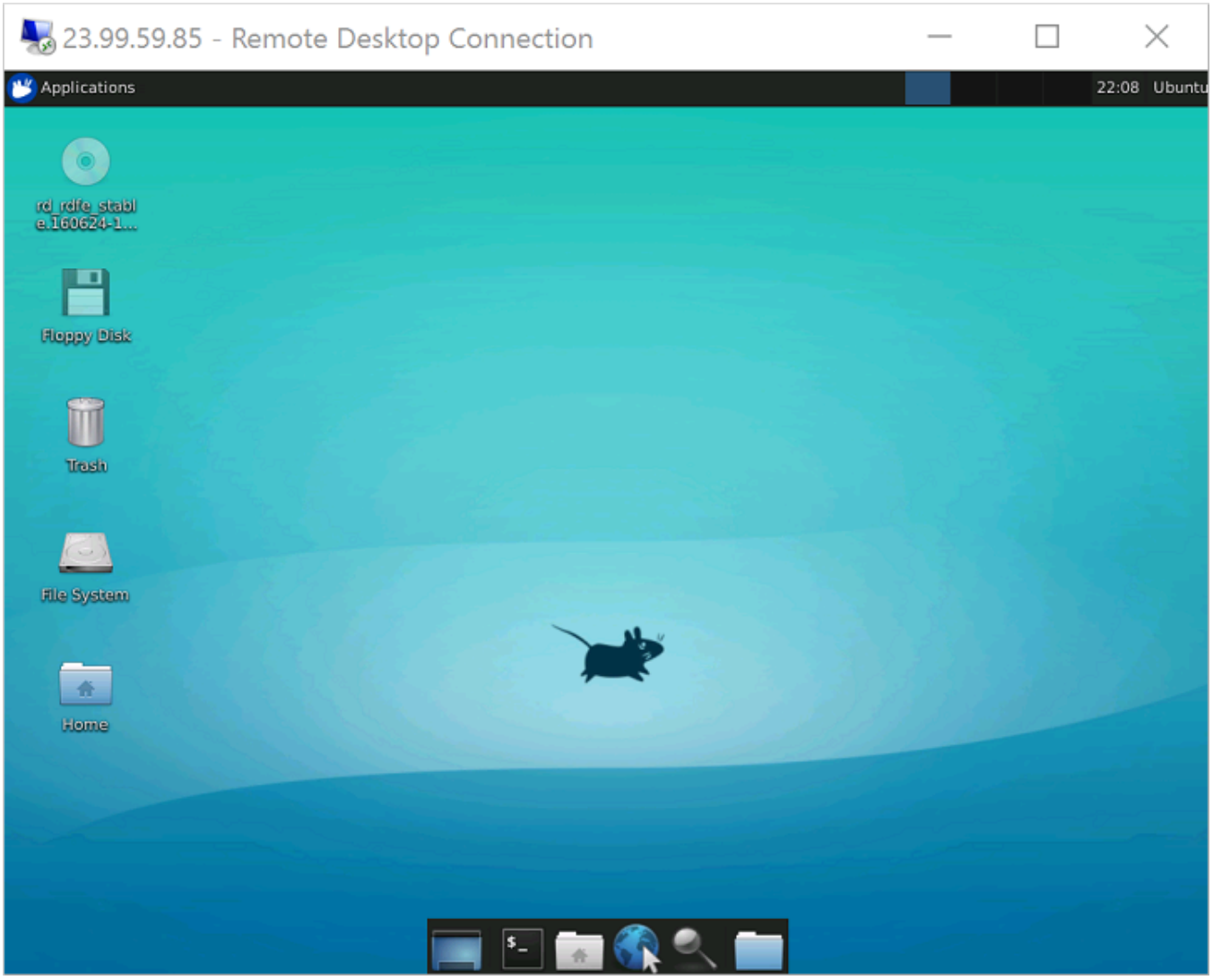
Open your local remote desktop client and connect to the IP address or DNS name of your Linux VM.



Enter the username and password for the user account on your VM as follows:



After authenticating, the xfce desktop environment will load and look similar to the following example:



If your local RDP client uses network level authentication (NLA), you may need to disable that connection setting. XRDP does not currently support NLA. You can also look at alternative RDP solutions that do support NLA, such as FreeRDP.

Troubleshoot

If you can't connect to your Linux VM using a Remote Desktop client, use `netstat` on your Linux VM to verify that your VM is listening for RDP connections as follows:

Bash

```
sudo netstat -plnt | grep rdp
```

The following example shows the VM listening on TCP port 3389 as expected:

Bash

```
tcp      0      0      127.0.0.1:3350      0.0.0.0:*      LISTEN      53192/xrdp-sesman
tcp      0      0      0.0.0.0:3389        0.0.0.0:*      LISTEN      53188/xrdp
```

If the *xrdp-sesman* service isn't listening, on an Ubuntu VM restart the service as follows:

```
Bash

sudo systemctl restart xrdp
```

Review logs in */var/log* on your Ubuntu VM for indications as to why the service may not be responding. You can also monitor the syslog during a remote desktop connection attempt to view any errors:

```
Bash

tail -f /var/log/syslog
```

If you don't receive any response in your remote desktop client and don't see any events in the system log, this behavior indicates that remote desktop traffic can't reach the VM. Review your network security group rules to ensure that you have a rule to permit TCP on port 3389. For more information, see Troubleshoot application connectivity issues.

Next steps

For more information about creating and using SSH keys with Linux VMs, see Create SSH keys for Linux VMs in Azure.

For information on using SSH from Windows, see How to use SSH keys with Windows.

How to connect using Secure Shell (SSH) and sign on to an Azure virtual machine running Windows

Article • 08/23/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

The Win32 OpenSSH project makes remote connectivity with Secure Shell ubiquitous by providing native support in Windows. The capability is provided in Windows Server version 2019 and later, and can be added to older versions of Windows using a virtual machine (VM) extension.

The examples below use variables. You can set variables in your environment as follows.

Expand table

Shell	Example
Bash/ZSH	myResourceGroup='resGroup10'
PowerShell	$myResourceGroup='resGroup10'

Enable SSH

First, you'll need to enable SSH in your Windows machine.

Deploy the SSH extension for Windows. The extension provides an automated installation of the Win32 OpenSSH solution, similar to enabling the capability in newer versions of Windows. Use the following examples to deploy the extension.

Azure CLI

Azure CLI

```
az vm extension set --resource-group $myResourceGroup --vm-name $myVM --
name WindowsOpenSSH --publisher Microsoft.Azure.OpenSSH --version 3.0
```

Open TCP port

Ensure the appropriate port (by default, TCP 22) is open to allow connectivity to the VM.

Azure CLI

Azure CLI

```
az network nsg rule create -g $myResourceGroup --nsg-name $myNSG -n
allow-SSH --priority 1000 --source-address-prefixes 208.130.28.4/32 --
destination-port-ranges 22 --protocol TCP
```

- Your VM must have a public IP address. To check if your VM has a public IP address, select **Overview** from the left menu and look at the **Networking** section. If you see an IP address next to **Public IP address**, then your VM has a public IP. To learn more about adding a public IP address to an existing VM, see Associate a public IP address to a virtual machine
- Verify your VM is running. On the Overview tab, in the essentials section, verify the status of the VM is Running. To start the VM, select **Start** at the top of the page.

Authentication

You can authenticate to Windows machines using either username and password or SSH keys. Azure doesn't support provisioning public keys to Windows machines automatically, however you can copy the key using the RunCommand extension.

Overview of SSH and keys

SSH is an encrypted connection protocol that provides secure sign-ins over unsecured connections. Although SSH provides an encrypted connection, using passwords with SSH connections still leaves the VM vulnerable to brute-force attacks. We recommend connecting to a VM over SSH using a public-private key pair, also known as *SSH keys*.

- The *public key* is placed on your VM.
- The *private key* remains on your local system. Protect this private key. Do not share it.

When you use an SSH client to connect to your VM (which has the public key), the remote VM tests the client to make sure it has the correct private key. If the client has the private key, it's granted access to the VM.

Depending on your organization's security policies, you can reuse a single public-private key pair to access multiple Azure VMs and services. You do not need a separate pair of keys for each VM or service you wish to access.

Your public key can be shared with anyone, but only you (or your local security infrastructure) should have access to your private key.

Supported SSH key formats

Azure currently supports SSH protocol 2 (SSH-2) RSA public-private key pairs with a minimum length of 2048 bits. Support for ED25519 Keys is in preview, other key formats such as ECDH and ECDSA are not supported.

Copy a public key using the RunCommand extension.

The RunCommand extension provides an easy solution to copying a public key into Windows machines and making sure the file has correct permissions.

Azure CLI

Azure CLI

```
az vm run-command invoke -g $myResourceGroup -n $myVM --command-id
RunPowerShellScript --scripts "MYPUBLICKEY | Add-Content
'C:\ProgramData\ssh\administrators_authorized_keys' -Encoding
UTF8;icacls.exe 'C:\ProgramData\ssh\administrators_authorized_keys'
/inheritance:r /grant 'Administrators:F' /grant 'SYSTEM:F'"
```

Connect using Az CLI

Connect to Windows machines using `Az SSH` commands.

Azure CLI

```
az ssh vm  -g $myResourceGroup -n $myVM --local-user $myUsername
```

It's also possible to create a network tunnel for specific TCP ports through the SSH connection. A good use case for this is Remote Desktop which defaults to port 3389.

Azure CLI

```
az ssh vm  -g $myResourceGroup -n $myVM --local-user $myUsername -- -L
3389:localhost:3389
```

Connect from Azure portal

1. Go to the Azure portal to connect to a VM. Search for and select **Virtual machines**.
2. Select the virtual machine from the list.
3. Select **Connect** from the left menu.
4. Select the option that fits with your preferred way of connecting. The portal helps walk you through the prerequisites for connecting.

Next steps

Learn how to transfer files to an existing VM, see Use SCP to move files to and from a VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Setting up WinRM access for Virtual Machines in Azure Resource Manager

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

Here are the steps you need to take to set up a VM with WinRM connectivity

1. Create a Key Vault
2. Create a self-signed certificate
3. Upload your self-signed certificate to Key Vault
4. Get the URL for your self-signed certificate in the Key Vault
5. Reference your self-signed certificates URL while creating a VM

Step 1: Create a Key Vault

You can use the below command to create the Key Vault

Azure PowerShell

```
New-AzKeyVault -VaultName "<vault-name>" -ResourceGroupName "<rg-name>" -
Location "<vault-location>" -EnabledForDeployment -
EnabledForTemplateDeployment
```

Step 2: Create a self-signed certificate

You can create a self-signed certificate using this PowerShell script

Azure PowerShell

```
$certificateName = "somename"

$thumbprint = (New-SelfSignedCertificate -DnsName $certificateName -
CertStoreLocation Cert:\CurrentUser\My -KeySpec KeyExchange).Thumbprint

$cert = (Get-ChildItem -Path cert:\CurrentUser\My\$thumbprint)

$password = Read-Host -Prompt "Please enter the certificate password." -
AsSecureString

Export-PfxCertificate -Cert $cert -FilePath ".\$certificateName.pfx" -
Password $password
```

Step 3: Upload your self-signed certificate to the Key Vault

Before uploading the certificate to the Key Vault created in step 1, it needs to be converted into a format the Microsoft.Compute resource provider will understand. The below PowerShell script will allow you to do that

Azure PowerShell

```
$fileName = "<Path to the .pfx file>"
$fileContentBytes = Get-Content $fileName -Encoding Byte
$fileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
[System.Collections.HashTable]$TableForJSON = @{
    "data"     = $fileContentEncoded;
    "dataType" = "pfx";
    "password" = "<password>";
}
[System.String]$jsonObject = $TableForJSON | ConvertTo-Json
$encoding = [System.Text.Encoding]::UTF8
$jsonEncoded =
[System.Convert]::ToBase64String($encoding.GetBytes($jsonObject))
$secret = ConvertTo-SecureString -String $jsonEncoded -AsPlainText –Force
Set-AzKeyVaultSecret -VaultName "<vault name>" -Name "<secret name>" -
SecretValue $secret
```

Step 4: Get the URL for your self-signed certificate in the Key Vault

The Microsoft.Compute resource provider needs a URL to the secret inside the Key Vault while provisioning the VM. This enables the Microsoft.Compute resource provider to download the secret and create the equivalent certificate on the VM.

> ⓘ **Note**
>
> The URL of the secret needs to include the version as well. An example URL looks like below https://contosovault.vault.azure.net:443/secrets/contososecret/01h9db0df2cd4300a20ence585a6s7ve

Templates

You can get the link to the URL in the template using the below code

JSON

```
"certificateUrl": "[reference(resourceId(resourceGroup().name,
'Microsoft.KeyVault/vaults/secrets', '<vault-name>', '<secret-name>'),
'2015-06-01').secretUriWithVersion]"
```

PowerShell

You can get this URL using the below PowerShell command

Azure PowerShell

```
$secretURL = (Get-AzKeyVaultSecret -VaultName "<vault name>" -Name "<secret
name>").Id
```

Step 5: Reference your self-signed certificates URL while creating a VM

Azure Resource Manager Templates

While creating a VM through templates, the certificate gets referenced in the secrets section and the winRM section as below:

JSON

```
"osProfile": {
      ...
      "secrets": [
        {
          "sourceVault": {
            "id": "<resource id of the Key Vault containing the secret>"
          },
          "vaultCertificates": [
            {
              "certificateUrl": "<URL for the certificate you got in Step
4>",
              "certificateStore": "<Name of the certificate store on the
VM>"
            }
          ]
        }
      ],
      "windowsConfiguration": {
        ...
        "winRM": {
          "listeners": [
```

```
            {
              "protocol": "http"
            },
            {
              "protocol": "https",
              "certificateUrl": "<URL for the certificate you got in Step
4>"
            }
          ]
        },
        ...
      }
    },
```

A sample template for the above can be found here at vm-winrm-keyvault-windows

Source code for this template can be found on GitHub

PowerShell

Azure PowerShell

```
$vm = New-AzVMConfig -VMName "<VM name>" -VMSize "<VM Size>"
$credential = Get-Credential
$secretURL = (Get-AzKeyVaultSecret -VaultName "<vault name>" -Name "<secret
name>").Id
$vm = Set-AzVMOperatingSystem -VM $vm -Windows -ComputerName "<Computer
Name>" -Credential $credential -WinRMHttp -WinRMHttps -ProvisionVMAgent -
WinRMCertificateUrl $secretURL
$sourceVaultId = (Get-AzKeyVault -ResourceGroupName "<Resource Group name>"
-VaultName "<Vault Name>").ResourceId
$CertificateStore = "My"
$vm = Add-AzVMSecret -VM $vm -SourceVaultId $sourceVaultId -CertificateStore
$CertificateStore -CertificateUrl $secretURL
```

Step 6: Connecting to the VM

Before you can connect to the VM, you'll need to make sure your machine is configured for WinRM remote management. Start PowerShell as an administrator and execute the below command to make sure you're set up.

Azure PowerShell

```
Enable-PSRemoting -Force
```

Note

You might need to make sure the WinRM service is running if the above does not work. You can do that using `Get-Service WinRM`

Once the setup is done, you can connect to the VM using the below command

Azure PowerShell

```
Enter-PSSession -ConnectionUri https://<public-ip-dns-of-the-vm>:5986 -
Credential $cred -SessionOption (New-PSSessionOption -SkipCACheck -
SkipCNCheck -SkipRevocationCheck) -Authentication Negotiate
```

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Time sync for Linux VMs in Azure

Article • 12/09/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

Time sync is important for security and event correlation. Sometimes it's used for distributed transactions implementation. Time accuracy between multiple computer systems is achieved through synchronization. Synchronization can be affected by multiple things, including reboots and network traffic between the time source and the computer fetching the time.

Azure is backed by infrastructure running Windows Server 2016. Windows Server 2016 has improved algorithms used to correct time and condition the local clock to synchronize with UTC. The Windows Server 2016 Accurate Time feature greatly improved how the VMICTimeSync service that governs VMs with the host for accurate time. Improvements include more accurate initial time on VM start or VM restore and interrupt latency correction.

> ⓘ **Note**
>
> For more information, see How the Windows Time Service Works. You can also find information specific to Windows Server 2016 on the Accurate time for Windows Server 2016 page.

Overview

Accuracy for a computer clock is gauged on how close the computer clock is to the Coordinated Universal Time (UTC) time standard. UTC is defined by a multinational sample of precise atomic clocks that can only be off by one second in 300 years. But, reading UTC directly requires specialized hardware. Instead, time servers are synced to UTC and are accessed from other computers to provide scalability and robustness. Every computer has time synchronization service running that knows what time servers to use and periodically checks if computer clock needs to be corrected and adjusts time if needed.

Azure hosts are synchronized to internal Microsoft time servers that take their time from Microsoft-owned Stratum 1 devices, with GPS antennas. Virtual machines in Azure can either depend on their host to pass the accurate time (*host time*) on to the VM or the VM can directly get time from a time server, or a combination of both.

On stand-alone hardware, the Linux OS only reads the host hardware clock on boot. After that, the clock is maintained using the interrupt timer in the Linux kernel. In this configuration, the clock will drift over time. In newer Linux distributions on Azure, VMs can use the VMICTimeSync provider, included in the Linux integration services (LIS), to query for clock updates from the host more frequently.

Virtual machine interactions with the host can also affect the clock. During memory preserving maintenance, VMs are paused for up to 30 seconds. For example, before maintenance begins the VM clock shows 10:00:00 AM and lasts 28 seconds. After the VM resumes, the clock on the VM would still show 10:00:00 AM, which would be 28 seconds off. To correct for this, the VMICTimeSync service monitors what is happening on the host and updates the time-of-day clock in Linux VMs to compensate.

Without time synchronization working, the clock on the VM would accumulate errors. When there's only one VM, the effect might not be significant unless the workload requires highly accurate timekeeping. But in most cases, we have multiple, interconnected VMs that use time to track transactions and the time needs to be consistent throughout the entire deployment. When time between VMs is different, you could see the following effects:

- Authentication will fail. Security protocols like Kerberos or certificate-dependent technology rely on time being consistent across the systems.
- It's hard to figure out what have happened in a system if logs (or other data) don't agree on time. The same event would look like it occurred at different times, making correlation difficult.
- If clock is off, the billing could be calculated incorrectly.

Configuration options

Time sync requires that a time sync service is running in the Linux VM, plus a source of accurate time information against which to synchronize. Typically ntpd or chronyd is used as the time sync service, though there are other open source time sync services that can be used as well. The source of accurate time information can be the Azure host or an external time service that is accessed over the public internet. By itself, the VMICTimeSync service doesn't provide ongoing time sync between the Azure host and a Linux VM except after pauses for host maintenance as described above.

Historically, most Azure Marketplace images with Linux have been configured in one of two ways:

- No time sync service is running by default

- ntpd is running as the time sync service, and synchronizing against an external NTP time source that is accessed over the network. For example, Ubuntu 18.04 LTS Marketplace images use **ntp.ubuntu.com**.

To confirm ntpd is synchronizing correctly, run the `ntpq -p` command.

Some Azure Marketplace images with Linux are being changed to use chronyd as the time sync service, and chronyd is configured to synchronize against the Azure host rather than an external NTP time source. The Azure host time is usually the best time source to synchronize against, as it is maintained accurately and reliably, and is accessible without the variable network delays inherent in accessing an external NTP time source over the public internet.

The VMICTimeSync is used in parallel and provides two functions:

- Immediately updates the Linux VM time-of-day clock after a host maintenance event
- Instantiates an IEEE 1588 Precision Time Protocol (PTP) hardware clock source as a /dev/ptp device that provides the accurate time-of-day from the Azure host. Chronyd can be configured to synchronize against this time source (which is the default configuration in the newest Linux images). Linux distributions with kernel version 4.11 or later (or version 3.10.0-693 or later for RHEL 7) support the /dev/ptp device. For earlier kernel versions that don't support /dev/ptp for Azure host time, only synchronization against an external time source is possible.

The default configuration can be changed. An older image that is configured to use ntpd and an external time source can be changed to use chronyd and the /dev/ptp device for Azure host time. Similarly, an image using Azure host time via a /dev/ptp device can be configured to use an external NTP time source if required by your application or workload.

Tools and resources

There are some basic commands for checking your time synchronization configuration. Documentation for Linux distribution has more details on the best way to configure time synchronization for that distribution.

Integration services

Check to see if the integration service (hv_utils) is loaded.

```
Bash
```

```
$ sudo lsmod | grep hv_utils
```

You should see something similar to this:

```
hv_utils               24418  0
hv_vmbus              397185  7
hv_balloon,hyperv_keyboard,hv_netvsc,hid_hyperv,hv_utils,hyperv_fb,hv_storvs
c
```

Check for PTP Clock Source

With newer versions of Linux, a Precision Time Protocol (PTP) clock source corresponding to the Azure host is available as part of the VMICTimeSync provider. On older versions of Red Hat Enterprise Linux 7.x, the Linux Integration Services can be downloaded and used to install the updated driver. When the PTP clock source is available, the Linux device is of the form /dev/ptp*x*.

See which PTP clock sources are available.

Bash

```
$ ls /sys/class/ptp
```

In this example, the value returned is *ptp0*, so we use that to check the clock name. To verify the device, check the clock name.

Bash

```
$ sudo cat /sys/class/ptp/ptp0/clock_name
```

This should return `hyperv`, meaning the Azure host.

In some Linux VMs you may see multiple PTP devices listed. One example is for Accelerated Networking the Mellanox mlx5 driver also creates a /dev/ptp device. Because the initialization order can be different each time Linux boots, the PTP device corresponding to the Azure host might be `/dev/ptp0` or it might be `/dev/ptp1`, which makes it difficult to configure `chronyd` with the correct clock source. To solve this problem, the most recent Linux images have a `udev` rule that creates the symlink `/dev/ptp_hyperv` to whichever `/dev/ptp` entry corresponds to the Azure host. Chrony

should always be configured to use the `/dev/ptp_hyperv` symlink instead of `/dev/ptp0` or `/dev/ptp1`.

If you are having issues with the `/dev/ptp_hyperv` device not being created, you can use the `udev` rule and steps below to configure it:

NOTE: Most Linux distributions should not need this udev rule as it has been implemented in newer versions of systemd

Create the `udev` rules file:

Bash

```
$ sudo cat > /etc/udev/rules.d/99-ptp_hyperv.rules << EOF
ACTION!="add", GOTO="ptp_hyperv"
SUBSYSTEM=="ptp", ATTR{clock_name}=="hyperv", SYMLINK += "ptp_hyperv"
LABEL="ptp_hyperv"
EOF
```

Reboot the Virtual Machine OR reload the `udev` rules with:

Bash

```
$ sudo udevadm control --reload
$ sudo udevadm trigger --subsystem-match=ptp --action=add
```

chrony

On Ubuntu 19.10 (and later versions) and Red Hat Enterprise Linux 8.x, chrony is configured to use a PTP source clock. Instead of chrony, older Linux releases use the Network Time Protocol daemon (ntpd), which doesn't support PTP sources. To enable PTP in those releases, chrony must be manually installed and configured (in chrony.conf) by using the following statement:

Bash

```
refclock PHC /dev/ptp_hyperv poll 3 dpoll -2 offset 0 stratum 2
```

If the /dev/ptp_hyperv symlink is available, use it instead of /dev/ptp0 to avoid any confusion with the /dev/ptp device created by the Mellanox mlx5 driver.

Stratum information isn't automatically conveyed from the Azure host to the Linux guest. The preceding configuration line specifies that the Azure host time source is to be treated as Stratum 2, which in turn causes the Linux guest to report itself as Stratum 3.

You can change the stratum setting in the configuration line if you want the Linux guest to report itself differently.

By default, chronyd accelerates or slows the system clock to fix any time drift. If the drift becomes too large, chrony fails to fix the drift. To overcome this, the `makestep` parameter in **/etc/chrony.conf** can be changed to force a time sync if the drift exceeds the threshold specified.

Bash

```
makestep 1.0 -1
```

Here, chrony will force a time update if the drift is greater than 1 second. To apply the changes restart the chronyd service:

Bash

```
$ sudo systemctl restart chronyd && sudo systemctl restart chrony
```

Time sync messages related to systemd-timesyncd

In some cases, the systemd-timesyncd service might still be enabled and trying to do a sync upon a reboot, if you are still seeing messages in syslog that look similar to:

```
systemd-timesyncd[945]: Network configuration changed, trying to establish
connection.
Aug  1 12:59:45 vm-name systemd-timesyncd[945]: Network configuration
changed, trying to establish connection.
Aug  1 12:59:45 vm-name systemd-timesyncd[945]: Network configuration
changed, trying to establish connection.
Aug  1 12:59:45 vm-name systemd-timesyncd[945]: Network configuration
changed, trying to establish connection.
Aug  1 12:59:45 vm-name systemd-timesyncd[945]: Network configuration
changed, trying to establish connection.
Aug  1 12:59:45 vm-name systemd-timesyncd[945]: Synchronized to time server
185.125.190.56:123 (ntp.ubuntu.com)
```

You can disable it by using:

Bash

```
$ sudo systemctl disable systemd-timesyncd
```

In most cases, systemd-timesyncd will try during boot but once chrony starts up it will overwrite and become the default time sync source.

For more information about Ubuntu and NTP, see Time Synchronization.

For more information about Red Hat and NTP, see Configure NTP.

For more information about chrony, see Using chrony.

systemd

On SUSE and Ubuntu releases before 19.10, time sync is configured using systemd. For more information about Ubuntu, see Time Synchronization. For more information about SUSE, see Section 4.5.8 in SUSE Linux Enterprise Server 12 SP3 Release Notes.

cloud-init

Images that use cloud-init to provision the VM can use the `ntp` section to setup a time sync service. An example of cloud-init installing chrony and configuring it to use the PTP clock source for Ubuntu VMs:

YAML

```yaml
#cloud-config
ntp:
  enabled: true
  ntp_client: chrony
  config:
    confpath: /etc/chrony/chrony.conf
    packages:
     - chrony
    service_name: chrony
    template: |
       ## template:jinja
       driftfile /var/lib/chrony/chrony.drift
       logdir /var/log/chrony
       maxupdateskew 100.0
       refclock PHC /dev/ptp_hyperv poll 3 dpoll -2 offset 0 stratum 2
       makestep 1.0 -1
```

You can then base64 the above cloud-config for use in the `osProfile` section in an ARM template:

PowerShell

```powershell
[Convert]::ToBase64String((Get-Content -Path ./cloud-config.txt -Encoding
```

```
Byte))
```

JSON

```
"osProfile": {
  "customData":
"I2Nsb3VkLWNvbmZpZwpudHA6CiAgZW5hYmxlZDogdHJ1ZQogIG50cF9jbGllbnQ6IGNocm9ueQo
gIGNvbmZpZzoKICAgIGNvbmZwYXRoOiAvZXRjL2Nocm9ueS9jaHJvbnkuY29uZgogICAgcGFja2F
nZXM6CiAgICAgLSBjaHJvbnkKICAgIHNlcnZpY2VfbmFtZTogY2hyb255CiAgICB0ZW1wbGF0ZTo
gfAogICAgICAgIyMgdGVtcGxhdGU6amluamEKICAgICAgIGRyaWZ0ZmlsZSAvdmFyL2xpYi9jaHJ
vbnkvY2hyb255LmRyaWZ0CiAgICAgICBsb2dkaXIgL3Zhci9sb2cvY2hyb255CiAgICAgICBtYXh
1cGRhdGVza2V5IDEwMC4wCiAgICAgICByZWZjbG9jayBQSEMgL2Rldi9wdHBfaHlwZXJ2IHBvbGw
gMyBkcG9sbCAtMgogICAgICAgbWFrZXN0ZXAgMS4wIC0x"
}
```

For more information about cloud-init on Azure, see Overview of cloud-init support for Linux VMs in Azure.

Next steps

For more information, see Accurate time for Windows Server 2016.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Time sync for Windows VMs in Azure

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Time sync is important for security and event correlation. Sometimes it is used for distributed transactions implementation. Time accuracy between multiple computer systems is achieved through synchronization. Synchronization can be affected by multiple things, including reboots and network traffic between the time source and the computer fetching the time.

Azure is now backed by infrastructure running Windows Server 2016. Windows Server 2016 has improved algorithms used to correct time and condition the local clock to synchronize with UTC. Windows Server 2016 also improved the VMICTimeSync service that governs how VMs sync with the host for accurate time. Improvements include more accurate initial time on VM start or VM restore and interrupt latency correction for samples provided to Windows Time (W32time).

> ⓘ **Note**
>
> For a quick overview of Windows Time service, take a look at this high-level overview video.
>
> For more information, see Accurate time for Windows Server 2016.

Overview

Accuracy for a computer clock is gauged on how close the computer clock is to the Coordinated Universal Time (UTC) time standard. UTC is defined by a multinational sample of precise atomic clocks that can only be off by one second in 300 years. But, reading UTC directly requires specialized hardware. Instead, time servers are synced to UTC and are accessed from other computers to provide scalability and robustness. Every computer has time synchronization service running that knows what time servers to use and periodically checks if computer clock needs to be corrected and adjusts time if needed.

Azure hosts are synchronized to internal Microsoft time servers that take their time from Microsoft-owned Stratum 1 devices, with GPS antennas. Virtual machines in Azure can either depend on their host to pass the accurate time (*host time*) on to the VM or the VM can directly get time from a time server, or a combination of both.

Virtual machine interactions with the host can also affect the clock. During memory preserving maintenance, VMs are paused for up to 30 seconds. For example, before maintenance begins the VM clock shows 10:00:00 AM and lasts 28 seconds. After the VM resumes, the clock on the VM would still show 10:00:00 AM, which would be 28 seconds off. To correct for this, the VMICTimeSync service monitors what is happening on the host and prompts for changes to happen on the VMs to compensate.

The VMICTimeSync service operates in either sample or sync mode and will only influence the clock forward. In sample mode, which requires W32time to be running, the VMICTimeSync service polls the host every 5 seconds and provides time samples to W32time. Approximately every 30 seconds, the W32time service takes the latest time sample and uses it to influence the guest's clock. Sync mode activates if a guest has been resumed or if a guest's clock drifts more than 5 seconds behind the host's clock. In cases where the W32time service is properly running, the latter case should never happen.

Without time synchronization working, the clock on the VM would accumulate errors. When there is only one VM, the effect might not be significant unless the workload requires highly accurate timekeeping. But in most cases, we have multiple, interconnected VMs that use time to track transactions and the time needs to be consistent throughout the entire deployment. When time between VMs is different, you could see the following effects:

- Authentication will fail. Security protocols like Kerberos or certificate-dependent technology rely on time being consistent across the systems.
- It's very hard to figure out what have happened in a system if logs (or other data) don't agree on time. The same event would look like it occurred at different times, making correlation difficult.
- If clock is off, the billing could be calculated incorrectly.

The best results for Windows deployments are achieved by using Windows Server 2016 as the guest operating system, which ensures you can use the latest improvements in time synchronization.

Configuration options

There are three options for configuring time sync for your Windows VMs hosted in Azure:

- Host time and time.windows.com. This is the default configuration used in Azure Marketplace images.
- Host-only.

- Use another, external time server with or without using host time. For this option follow the Time mechanism for Active Directory Windows Virtual Machines in Azure guide.

Use the default

By default Windows OS VM images are configured for w32time to sync from two sources:

- The NtpClient provider, which gets information from time.windows.com.
- The VMICTimeSync service, used to communicate the host time to the VMs and make corrections after the VM is paused for maintenance. Azure hosts use Microsoft-owned Stratum 1 devices to keep accurate time.

w32time would prefer the time provider in the following order of priority: stratum level, root delay, root dispersion, time offset. In most cases, w32time on an Azure VM would prefer host time due to evaluation it would do to compare both time sources.

For domain joined machines the domain itself establishes time sync hierarchy, but the forest root still needs to take time from somewhere and the following considerations would still hold true.

Host-only

Because time.windows.com is a public NTP server, syncing time with it requires sending traffic over the internet, varying packet delays can negatively affect quality of the time sync. Removing time.windows.com by switching to host-only sync can sometimes improve your time sync results.

Switching to host-only time sync makes sense if you experience time sync issues using the default configuration. Try out the host-only sync to see if that would improve the time sync on VM.

External time server

If you have specific time sync requirements, there is also an option of using external time servers. External time servers can provide specific time, which can be useful for test scenarios, ensuring time uniformity with machines hosted in non-Microsoft datacenters, or handling leap seconds in a special way.

You can combine external servers with the VMICTimeSync service and VMICTimeProvider to provide results similar to the default configuration.

Check your configuration

Check if the NtpClient time provider is configured to use explicit NTP servers (NTP) or domain time sync (NT5DS).

```
w32tm /dumpreg /subkey:Parameters | findstr /i "type"
```

If the VM is using NTP, you will see the following output:

```
Value Name                Value Type          Value Data
Type                      REG_SZ              NTP
```

To see what time server the NtpClient time provider is using, at an elevated command prompt type:

```
w32tm /dumpreg /subkey:Parameters | findstr /i "ntpserver"
```

If the VM is using the default, the output will look like this:

```
NtpServer                 REG_SZ              time.windows.com,0x8
```

To see what time provider is being used currently.

```
w32tm /query /source
```

Here is the output you could see and what it would mean:

- **time.windows.com** - in the default configuration, w32time would get time from time.windows.com. The time sync quality depends on internet connectivity to it and is affected by packet delays. This is the usual output you would get on a physical machine.
- **VM IC Time Synchronization Provider** - the VM is syncing time from the host. This is the usual output you would get on a virtual machine running on Azure.

- *Your domain server* - the current machine is in a domain and the domain defines the time sync hierarchy.
- *Some other server* - w32time was explicitly configured to get the time from that another server. Time sync quality depends on this time server quality.
- **Local CMOS Clock** - clock is unsynchronized. You can get this output if w32time hasn't had enough time to start after a reboot or when all the configured time sources are not available.

Opt in for host-only time sync

Azure is constantly working on improving time sync on hosts and can guarantee that all the time sync infrastructure is collocated in Microsoft-owned datacenters. If you have time sync issues with the default setup that prefers to use time.windows.com as the primary time source, you can use the following commands to opt in to host-only time sync.

Mark the VMIC provider as enabled.

```
reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\V
MICTimeProvider /v Enabled /t REG_DWORD /d 1 /f
```

Mark the NTPClient provider as disabled.

```
reg add
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\N
tpClient /v Enabled /t REG_DWORD /d 0 /f
```

Restart the w32time Service.

```
net stop w32time && net start w32time
```

Windows Server 2012 and R2 VMs

Windows Server 2012 and Windows Server 2012 R2 have different default settings for time sync. The w32time by default is configured in a way that prefers low overhead of

the service over precise time.

If you want to move your Windows Server 2012 and 2012 R2 deployments to use the newer defaults that prefer precise time, you can apply the following settings.

Update the w32time poll and update intervals to match Windows Server 2016 settings.

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config
/v MinPollInterval /t REG_DWORD /d 6 /f
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config
/v MaxPollInterval /t REG_DWORD /d 10 /f
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w32time\Config
/v UpdateInterval /t REG_DWORD /d 100 /f
w32tm /config /update
```

For `w32time` to be able to use the new poll intervals the NtpServers need to be marked as using them. If servers are annotated with the `0x1` bitflag mask, that would override this mechanism and `w32time` would use `SpecialPollInterval` instead. Make sure that specified NTP servers are either using the `0x8` flag or no flag at all:

Check what flags are being used for the NTP servers.

```
w32tm /dumpreg /subkey:Parameters | findstr /i "ntpserver"
```

Next steps

Below are links to more details about the time sync:

- Windows Time Service Tools and Settings
- Windows Server 2016 Improvements
- Accurate Time for Windows Server 2016
- Support boundary to configure the Windows Time service for high-accuracy environments

---

Feedback

Was this page helpful? Yes No

Configure the time mechanism for Active Directory Windows Virtual Machines in Azure

Article • 04/29/2025

Applies to: ✔ Windows Virtual Machines

Use this guide to learn how to set up time synchronization for your Azure Windows Virtual Machines that belong to an Active Directory Domain.

Time sync hierarchy in Active Directory Domain Services

Time synchronization in Active Directory should be managed by only allowing the **PDC** to access an external time source or NTP Server.

All other Domain Controllers would then sync time against the PDC, and all other members will get their time from the Domain Controller that satisfied that member's authentication request.

If you have an Active Directory domain running on virtual machines hosted in Azure, follow these steps to properly set up Time Sync.

> **Note**
>
> This guide focuses on using the **Group Policy Management** console to perform the configuration. You can achieve the same results by using the Command Prompt, PowerShell, or by manually modifying the Registry; however, those methods are not in scope in this article.

GPO to allow the PDC to synchronize with an External NTP Source

To check current time source in your **PDC**, from an elevated command prompt run *w32tm /query /source* and note the output for later comparison.

1. From *Start* run *gpmc.msc*.
2. Browse to the Forest and Domain where you want to create the GPO.
3. Create a new GPO, for example *PDC Time Sync*, in the container *Group Policy Objects*.

4. Right-click on the newly created GPO and Edit.
5. Navigate to the *Global Configuration Settings* policy under *Computer Configuration -> Administrative Templates -> System -> Windows Time Service*.
6. Set it to *Enabled* and configure the *AnnounceFlags* parameter to **5**.
7. Navigate to *Computer Configuration -> Administrative Templates -> System -> Windows Time Service -> Time Providers*.
8. Double click the *Configure Windows NTP Client* policy and set it to *Enabled*, configure the parameter *NTPServer* to point to an IP address or FQDN of a time server followed by `,0x9` for example: `131.107.13.100,0x9` and configure *Type* to **NTP**. For all the other parameters you can use the default values, or use custom ones according to your corporate needs.
9. Click the *Next Setting* button, set the *Enable Windows NTP Client* policy to *Enabled* and click *OK*
10. In the *Scope* tab of the newly created GPO navigate to **Security Filtering** and highlight the *Authenticated Users* group -> Click the *Remove* button -> *OK* -> *OK*
11. Create a WMI Filter to dynamically get the Domain Controller that holds the PDC role:

    - In the *Group Policy Management* console, navigate to *WMI Filters*, right-click on it and select *New*.
    - In the *New WMI Filter* window, give a name to the new filter, for example, *Get PDC Emulator* -> Fill out the *Description* field (optional) -> Click the *Add* button.
    - In the *WMI Query* window leave the *Namespace* as is, in the *Query* text box paste the following string `Select * from Win32_ComputerSystem where DomainRole = 5`, then click the *OK* button.
    - Back in the *New WMI Filter* window click the *Save* button.

12. In the *Scope* tab of the newly created GPO navigate to the **WMI Filtering** drop-down menu and select the previously created WMI filter then click *OK*.
13. In the *Scope* tab of the newly created GPO navigate to the **Security Filtering** click the *Add* button and browse for the *Domain Controllers* group, then click the *OK* button.
14. Link the GPO to the **Domain Controllers** Organizational Unit.

> ⓘ **Note**
>
> It can take up to 15 minutes for these changes to be reflected by the system.

From an elevated command prompt rerun *w32tm /query /source* and compare the output to the one you noted at the beginning of the configuration. Now it will be set to the NTP Server you chose.

> 💡 **Tip**

If you want to speed-up the process of changing the NTP source on your **PDC**, from an elevated command prompt run *gpupdate /force*, followed by *w32tm /resync /nowait*, then rerun *w32tm /query /source*; the output should be the NTP Server you used in the above GPO.

> ⓘ **Note**
>
> If after the changes are applied, you still don't get the output to the NTP Server you chose, you may need to set the value to **0** in the key **Enabled** under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMICTimeProvider** on your PDC and DCs too.

GPO for Members

Usually, NTP in Active Directory Domain Services will follow the AD DS Time Hierarchy mentioned at the beginning of this article and no further configuration is required.

Nevertheless, virtual machines hosted in Azure have specific security settings applied to them directly by the Cloud platform.

For all other domain members that are not Domain Controllers, you will need to modify the registry and set the value to **0** in the key **Enabled** under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMICTimeProvider**

> ⓘ **Important**
>
> Remember that serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully and test them on a couple of test virtual machines to make sure you will get the expected outcome. For added protection, back up the registry before you modify it. Then, you can restore the registry if a problem occurs. For how to back up and restore the Windows Registry follow the steps below.

Back up the Registry

1. From *Start* type *regedit.exe*, and then press `Enter`. If you are prompted for an administrator password or for confirmation, type the password or provide confirmation.

2. In the *Registry Editor* window, locate and click the registry key or subkey that you want to back up.
3. From the *File* menu select *Export*.
4. In the *Export Registry File* dialog box, select the location to which you want to save the backup copy, type a name for the backup file in the *File name* field, and then click *Save*.

Restore a Registry backup

1. From *Start* type *regedit.exe*, and then press `Enter`. If you are prompted for an administrator password or for confirmation, type the password or provide confirmation.
2. In the *Registry Editor* window, from the *File* menu select *Import*.
3. In the *Import Registry File* dialog box, select the location to which you saved the backup copy, select the backup file, and then click *Open*.

GPO to disable the VMICTimeProvider

Configure the following Group Policy Object to enable domain members to synchronize time with Domain Controllers in their corresponding Active Directory Site:

To check current time source, login to any domain member and from an elevated command prompt run *w32tm /query /source* and note the output for later comparison.

1. From a Domain Controller go to *Start* run *gpmc.msc*.
2. Browse to the Forest and Domain where you want to create the GPO.
3. Create a new GPO, for example *Clients Time Sync*, in the container *Group Policy Objects*.
4. Right-click on the newly created GPO and Edit.
5. Navigate to *Computer Configuration* -> *Preferences* -> *Windows Settings* -> Right-click on *Registry* -> *New* -> *Registry Item*
6. In the *New Registry Properties* window set the following values:

    - On **Action:** *Update*
    - On **Hive:** *HKEY_LOCAL_MACHINE*
    - On **Key Path**: browse to *SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMICTimeProvider*
    - On **Value Name** type *Enabled*
    - On **Value Type**: *REG_DWORD*
    - On **Value Data**: type *0*

7. For all the other parameters use the default values and click *OK*
8. Link the GPO to the Organizational Unit where your members are located.
9. Wait or manually force a *Group Policy Update* on the domain member.

Go back to the domain member and from an elevated command prompt rerun *w32tm /query /source* and compare the output to the one you noted at the beginning of the configuration. Now it will be set to the Domain Controller that satisfied the member's authentication request.

Next steps

Below are links to more details about the time sync:

- Windows Time Service Tools and Settings
- Windows Server 2016 Improvements
- Accurate Time for Windows Server 2016
- Support boundary to configure the Windows Time service for high-accuracy environments

Run scripts in your VM by using Run Command

Summarize this article for me

Run Command uses the virtual machine (VM) agent to run scripts within an Azure Windows or Linux VM. You can use these scripts for general machine or application management. They can help you to quickly diagnose and remediate VM access and network issues and get the VM back to a good state. Scripts can be embedded in the properties or referenced to a pre published gallery script.

The original set of commands are action orientated. The updated set of commands are management orientated and enable you to run multiple scripts and has less restrictions. This article will explain the difference between the two sets of run commands and help you decide which set is the right one to use in your scenario.

> ⓘ **Important**
>
> **Managed Run Command** is currently available in Azure CLI, PowerShell, and API at this time. Portal functionality isn't currently available.

When to use action or managed commands

The original set of commands are action orientated. You should consider using this set of commands for situations where you need to run:

- A small script to get a content from a VM
- A script to configure a VM (set registry keys, change configuration)
- A one time script for diagnostics

See Action Run Commands for Linux and Action Run Commands for Windows for available action commands and instructions on how to apply them.

The updated set of commands are management orientated. Consider using managed run commands if your needs align to the following examples:

- Script needs to run as part of VM deployment
- Recurrent script execution is needed
- Multiple scripts needs to execute sequentially
- Bootstrap a VM by running installation scripts
- Publish custom script to be shared and reused

See Managed Run Command for Linux and Managed Run Command for Windows to learn how to use them.

> **Note**
>
> The maximum number of allowed Managed Run Commands is currently limited to 25.

Compare feature support

Expand table

Feature support	Action RunCommand	Managed RunCommand
ARM template	No, it's a POST action	Yes, it's a resource type
Long running	90 min limit	Customer specified timeout
Execution account	System account / root	Customer specified user
Multiple run commands	Only one active	Multiple in parallel or sequenced
Large output	Limited to 4k (in status blob)	Uploaded to customer append blob
Progress tracking	Reports only final status	Reports progress and last 4k output during execution
Async execution	Goal state/provisioning waits for script to complete	Customer specified async flag if provisioning waits for the script
Virtual machine scale set support	Only on VM instance	Support virtual machine scale set model and scale out
SAS generation	No blob support	Automated, CRP generates SAS for customer blobs and manages them
Gallery (custom commands)	Only built-in commandIds	Customer can publish scripts and share them

Next steps

Get started with Managed Run Command for Linux or Managed Run Command for Windows.

Last updated on 02/23/2026

Run scripts in your Linux VM by using action Run Commands

08/27/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The Run Command feature uses the virtual machine (VM) agent to run shell scripts within an Azure Linux VM. You can use these scripts for general machine or application management. They can help you to quickly diagnose and remediate VM access and network issues and get the VM back to a good state.

Benefits

You can access your virtual machines in multiple ways. Run Command can run scripts on your virtual machines remotely by using the VM agent. You use Run Command through the Azure portal, REST API, or Azure CLI for Linux VMs.

This capability is useful in all scenarios where you want to run a script within a virtual machine. It's one of the only ways to troubleshoot and remediate a virtual machine that doesn't have the RDP or SSH port open because of network or administrative user configuration.

Prerequisites

Linux Distro's Supported

Expand table

Publisher	Distribution	x64	ARM64
Alma Linux Community	Alma Linux	8.x+, 9.x+	8.x+, 9.x+
Credativ	Debian	10+	11.x+
Kinvolk	Flatcar Linux	3374.2.x+	3374.2.x+
Microsoft	Azure Linux	2.x	2.x
openSUSE Project	openSUSE	12.3+	*Not supported*
Oracle	Oracle Linux	6.4+, 7.x+, 8.x+	*Not supported*

Publisher	Distribution	x64	ARM64
Red Hat	Red Hat Enterprise Linux	6.7+, 7.x+, 8.x+, 9.x+, 10.x+	8.6+, 9.0+, 10.x+
CIQ	Rocky Linux	9.x+	9.x+
SUSE	SLES	12.x+, 15.x+	15.x SP4+
Canonical	Ubuntu (LTS releases)	18.04+, 20.04+, 22.04+, 24.04+	20.04+, 22.04+, 24.04+

Restrictions

The following restrictions apply when you're using Run Command:

- Output is limited to the last 4,096 bytes.
- The minimum time to run a script is about 20 seconds.
- Scripts run by default as an elevated user on Linux.
- You can run one script at a time.
- Scripts that prompt for information (interactive mode) aren't supported.
- You can't cancel a running script.
- The maximum time a script can run is 90 minutes. After that, the script will time out.
- Outbound connectivity from the VM is required to return the results of the script.

> **Note**
>
> To function correctly, Run Command requires connectivity (port 443) to Azure public IP addresses. If the extension doesn't have access to these endpoints, the scripts might run successfully but not return the results. If you're blocking traffic on the virtual machine, you can use service tags to allow traffic to Azure public IP addresses by using the `AzureCloud` tag.

Available commands

This table shows the list of commands available for Linux VMs. You can use the **RunShellScript** command to run any custom script that you want. When you're using the Azure CLI or PowerShell to run a command, the value that you provide for the `--command-id` or `-CommandId` parameter must be one of the following listed values. When you specify a value that isn't an available command, you receive this error:

```
error
```

```
The entity was not found in this Azure location
```

Expand table

Name	Description	More Info
RunShellScript	Runs a Linux shell script.	
ifconfig	Gets the configuration of all network interfaces.	readme

Azure CLI

The following example uses the az vm run-command command to run a shell script on an Azure Linux VM.

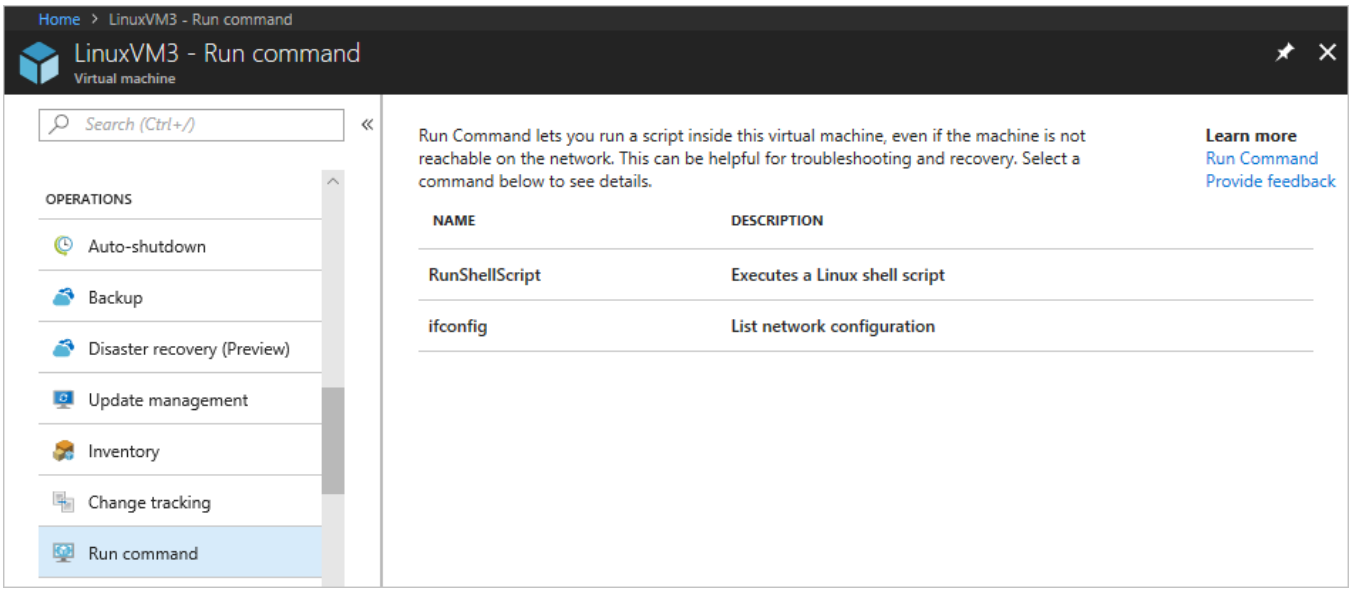
Azure CLI

```
az vm run-command invoke -g myResourceGroup -n myVm --command-id RunShellScript --scripts "apt-get update && apt-get install -y nginx"
```

> **Note**
>
> To run commands as a different user, enter `sudo -u` to specify a user account.

Azure portal

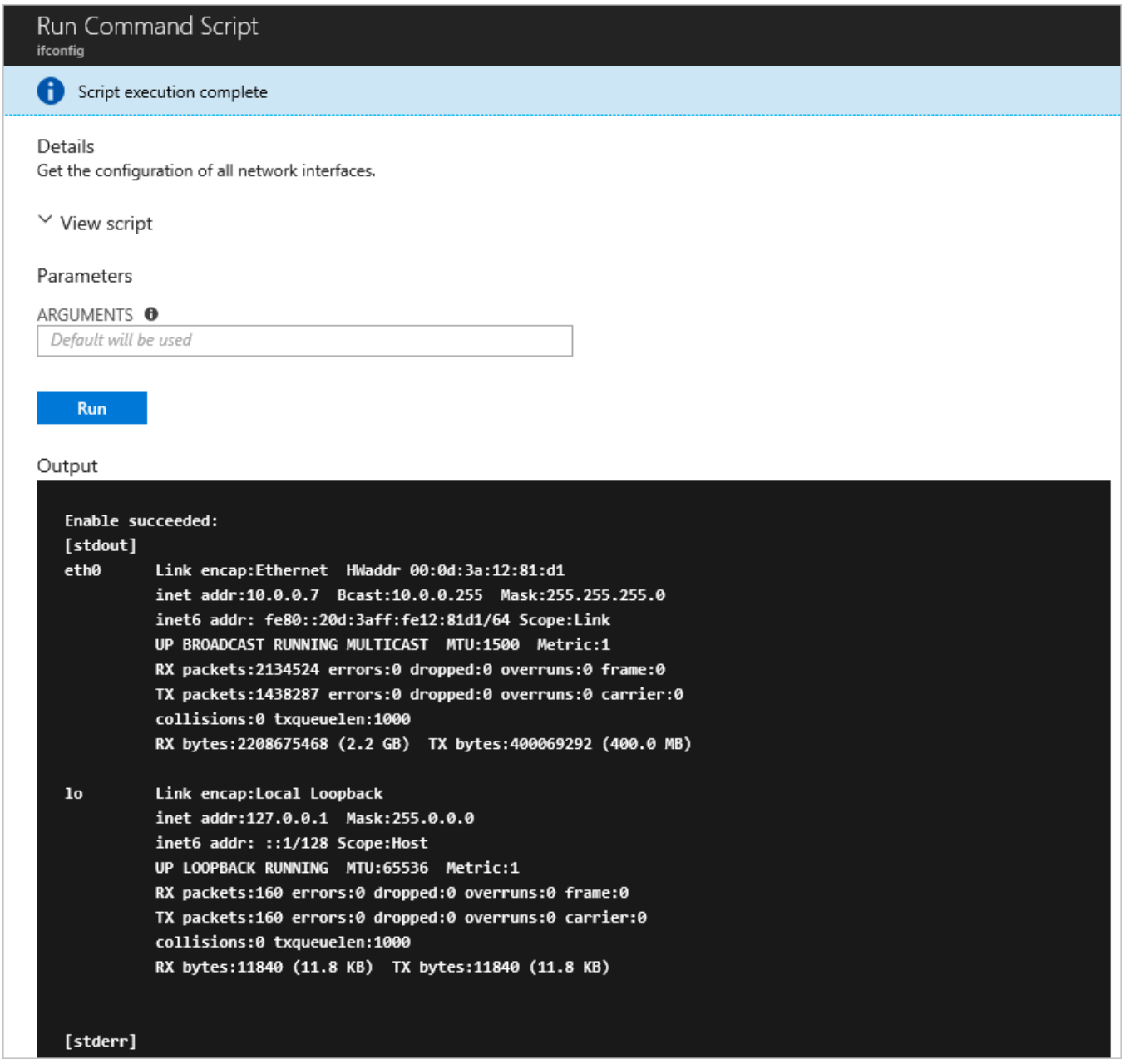
Go to a VM in the Azure portal and select **Run command** in the left menu, under **Operations**. You see a list of the available commands to run on the VM.



Choose a command to run. Some of the commands might have optional or required input parameters. For those commands, the parameters are presented as text fields for you to provide the input values. For each command, you can view the script that's being run by expanding **View script**. **RunShellScript** is different from the other commands, because it allows you to provide your own custom script.

> ⓘ **Note**
>
> The built-in commands are not editable.

After you choose the command, select **Run** to run the script. After the script finishes, it returns the output and any errors in the output window. The following screenshot shows an example output from running the **ifconfig** command.



PowerShell

The following example uses the Invoke-AzVMRunCommand cmdlet to run a PowerShell script on an Azure VM. The cmdlet expects the script referenced in the `-ScriptPath` parameter to be local to where the cmdlet is being run.

PowerShell

```
Invoke-AzVMRunCommand -ResourceGroupName '<myResourceGroup>' -Name '<myVMName>' -
CommandId 'RunShellScript' -ScriptPath '<pathToScript>' -Parameter @{"arg1" =
"var1";"arg2" = "var2"}
```

Limiting access to Run Command

Listing the run commands or showing the details of a command requires the `Microsoft.Compute/locations/runCommands/read` permission on Subscription level. The built-in Reader role and higher levels have this permission.

Running a command requires the `Microsoft.Compute/virtualMachines/runCommands/write` permission. The Virtual Machine Contributor role and higher levels have this permission.

You can use one of the built-in roles or create a custom role to use Run Command.

Action Run Command Linux troubleshooting

When troubleshooting action run command for Linux environments, refer to the *handler* log file typically located in the following directory: `/var/log/azure/run-command-handler/handler.log` for further details.

Known issues

The Linux action run command logs have a few notable differences compared to the action run command Windows logs:

- The sequence number is reported with each line of the log as 'seq=#'
- There won't be a line that contains `Awaiting completion...` as this will be in action run command Windows only.
- The line `Command existed with code: #` is also only present in action run command Windows logging.

Action Run Command Removal

If needing to remove your action run command Linux extension, refer to the below steps for Azure PowerShell and CLI:

Replace *rgname* and *vmname* with your relevant resource group name and virtual machine name in the following removal examples.

PowerShell

```
 Invoke-AzVMRunCommand -ResourceGroupName 'rgname' -VMName 'vmname' -CommandId
'RemoveRunCommandLinuxExtension'
```

Azure CLI

```
az vm run-command invoke  --command-id RemoveRunCommandLinuxExtension --name
vmname -g rgname
```

> **Note**
>
> When you apply a Run Command again, the extension will get installed automatically. You can use the extension removal command to troubleshoot any issues related to the extension.

Next steps

To learn about other ways to run scripts and commands remotely in your VM, see Run scripts in your Linux VM.

Run scripts in your Windows VM by using action Run Commands

The Run Command feature uses the virtual machine (VM) agent to run PowerShell scripts within an Azure Windows VM. You can use these scripts for general machine or application management. They can help you to quickly diagnose and remediate VM access and network issues and get the VM back to a good state.

Benefits

You can access your virtual machines in multiple ways. Run Command can run scripts on your virtual machines remotely by using the VM agent. You use Run Command through the Azure portal, REST API, or PowerShell for Windows VMs.

This capability is useful in all scenarios where you want to run a script within a virtual machine. It's one of the only ways to troubleshoot and remediate a virtual machine that doesn't have the RDP or SSH port open because of improper network or administrative user configuration.

Prerequisites

Windows Operating Systems Supported

Expand table

OS Version	x64	ARM64
Windows 10	Supported	Supported
Windows 11	Supported	Supported
Windows Server 2016	Supported	Supported
Windows Server 2016 Core	Supported	Supported
Windows Server 2019	Supported	Supported
Windows Server 2019 Core	Supported	Supported
Windows Server 2022	Supported	Supported
Windows Server 2022 Core	Supported	Supported
Windows Server 2022 Azure Edition	Supported	Supported

OS Version	x64	ARM64
Windows Server 2025	Supported	Supported
Windows Server 2025 Core	Supported	Supported
Windows Server 2025 Azure Edition	Supported	Supported

Restrictions

The following restrictions apply when you're using Run Command:

- Output is limited to the last 4,096 bytes.
- The minimum time to run a script is about 20 seconds.
- Scripts run as System on Windows.
- One script at a time can run.
- Scripts that prompt for information (interactive mode) aren't supported.
- You can't cancel a running script.
- The maximum time a script can run is 90 minutes. After that, it will time out.
- Outbound connectivity from the VM is required to return the results of the script.
- It isn't recommended to run a script that will cause a stop or update of the VM Agent. This can let the extension in a Transitioning state, leading to a time-out.

> **Note**
>
> To function correctly, Run Command requires connectivity (port 443) to Azure public IP addresses. If the extension doesn't have access to these endpoints, the scripts might run successfully but not return the results. If you're blocking traffic on the virtual machine, you can use service tags to allow traffic to Azure public IP addresses by using the `AzureCloud` tag.
>
> The Run Command feature doesn't work if the VM agent status is NOT READY. Check the agent status in the VM's properties in the Azure portal.

Available commands

This table shows the list of commands available for Windows VMs. You can use the **RunPowerShellScript** command to run any custom script that you want. When you're using the Azure CLI or PowerShell to run a command, the value that you provide for the `--command-id` or `-CommandId` parameter must be one of the following listed values. When you specify a value that isn't an available command, you receive this error:

error

```
The entity was not found in this Azure location
```

Available Commands

Expand table

CommandID	Description	More Info
RunPowerShellScript	Runs a PowerShell script	
DisableNLA	Disables Network Level Authentication (NLA). You must restart the VM after the script completes for the change to take effect. The script itself does not restart the VM.	readme
DisableWindowsUpdate	Disable Automatic Updates through Windows Update.	readme
EnableAdminAccount	Checks if the local Administrator account is disabled, and if so enables it.	readme
EnableEMS	Enable Emergency Management Services (EMS) to allow for serial console connection in troubleshooting scenarios.	readme
EnableRemotePS	Enable remote PowerShell.	readme
EnableWindowsUpdate	Enable Automatic Updates through Windows Update.	readme
IMDSCertCheck	Checks IMDS Health and then analyzes currently installed certificates that IMDS depends upon. If missing, shows additional details and mitigation steps.	readme
IPConfig	Shows detailed information for the IP address, subnet mask and default gateway for each adapter bound to TCP/IP. For usage, refer to Run scripts	readme
RDPSettings	Checks registry settings and domain policy settings. Suggests policy actions if machine is part of a domain or modifies the settings to default values.	readme

CommandID	Description	More Info
ResetRDPCert	Removes the SSL certificate tied to the RDP listener and restores the RDP listener security to default. Use this script if you see any issues with the certificate.	readme
SetRDPPort	Sets the default or user specified port number for Remote Desktop connections. Enables firewall rule for inbound access to the port.	readme
WindowsActivationValidation	Checks the current Windows license status (activated or not) and alert is generated if the machine is not property activated.	readme
WindowsGhostedNicValidationScript	This script scans the Windows registry for network adapters on PCI and VMBUS buses, compares them with currently active network adapters, and identifies ghosted NICs. Useful for troubleshooting network issues or cleaning up old NICs.	readme
WindowsUpgradeAssessmentValidation	This script is designed to assess the readiness of a Windows machine (client or server) for an in-place OS upgrade, with special considerations for Azure Virtual Machines. It evaluates OS version, supported upgrade paths, system disk space, and Azure security features like Trusted Launch, Secure Boot, and vTPM.	readme

Limiting access to Run Command

Listing the run commands or showing the details of a command requires the `Microsoft.Compute/locations/runCommands/read` permission on Subscription Level. The built-in Reader role and higher levels have this permission.

Running a command requires the `Microsoft.Compute/virtualMachines/runCommand/action` permission. The Virtual Machine Contributor role and higher levels have this permission.

You can use one of the built-in roles or create a custom role to use Run Command.

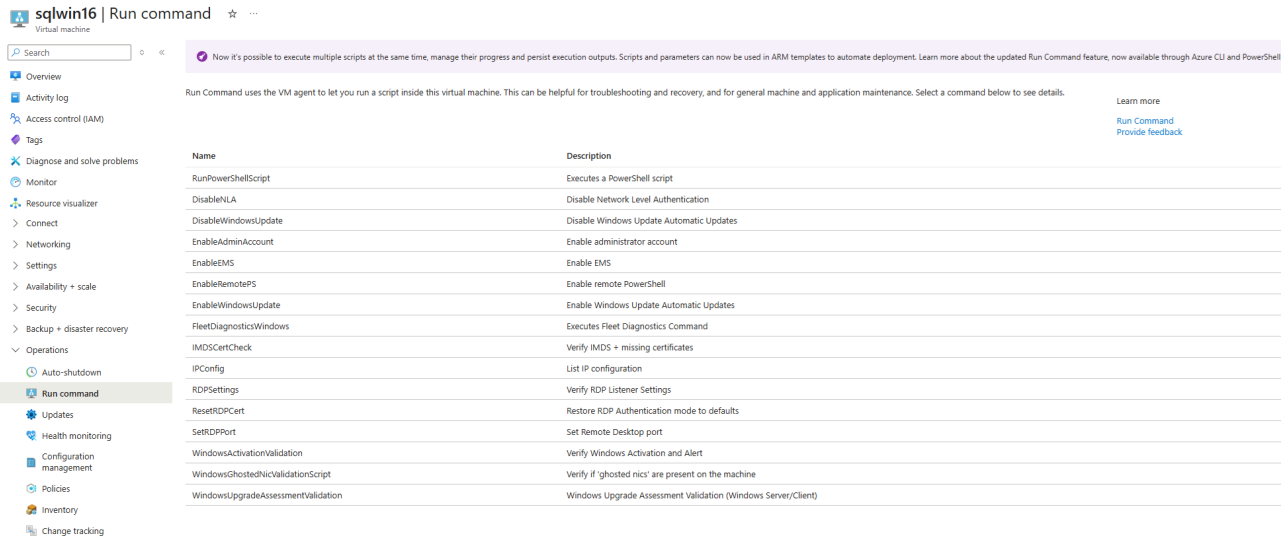
Using Run Commands

> ⓘ Note

- Parameter values can be string type only and the script is responsible for converting them to other types if needed.
- Depending on which modality is used to execute, some escaping may be needed. For example, if you're executing the command in a PowerShell session, the path to the script file will need to have quotes.
- The built-in commands are not editable.

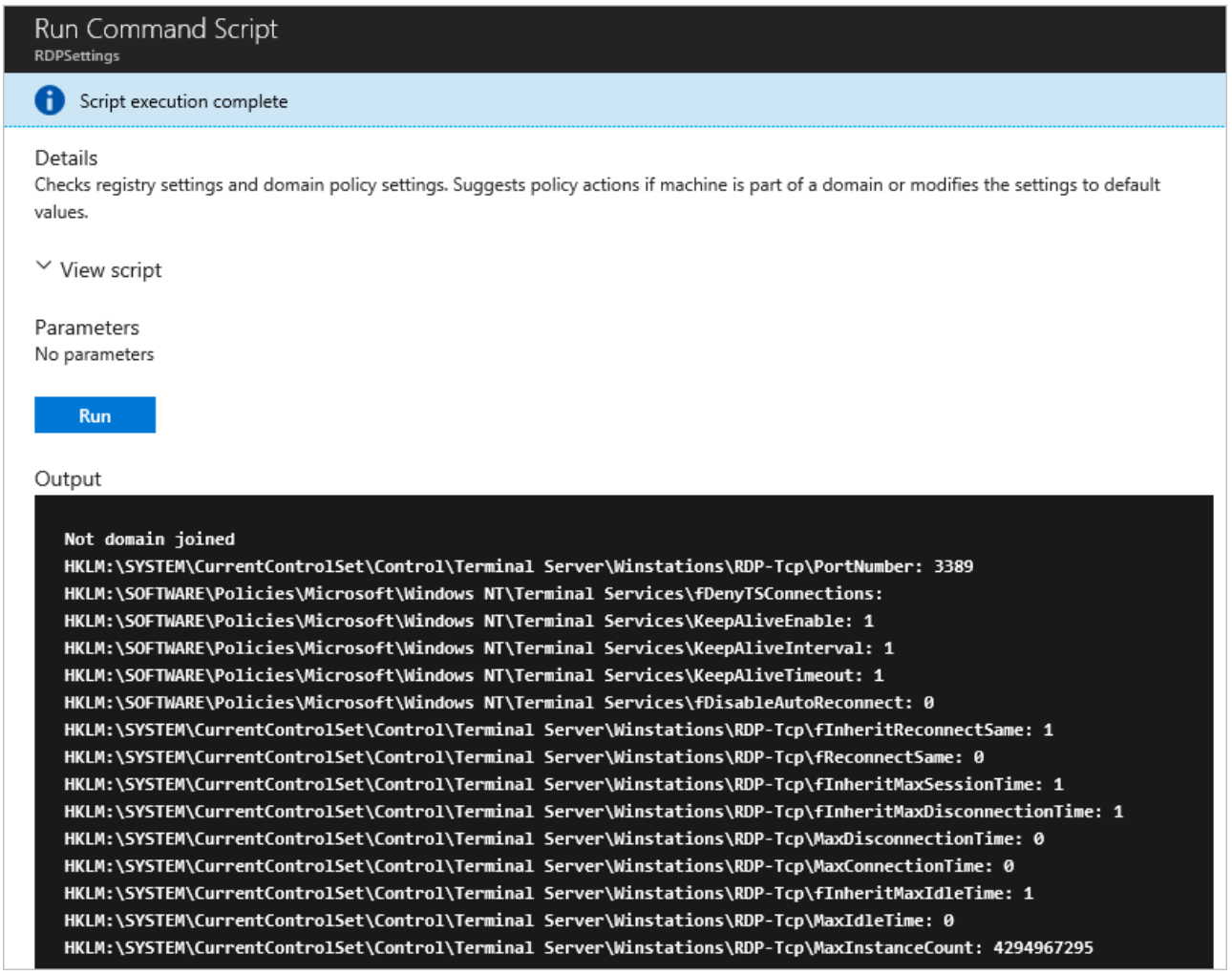
Portal

Go to a VM in the Azure portal and select **Run command** from the left menu, under **Operations**. You see a list of the available commands to run on the VM.



Choose a command to run. Some of the commands might have optional or required input parameters. For those commands, the parameters are presented as text fields for you to provide the input values. For each command, you can view the script that's being run by expanding **View script**. **RunPowerShellScript** is different from the other commands, because it allows you to provide your own custom script.

After you choose the command, select **Run** to run the script. After the script finishes, it returns the output and any errors in the output window. The following screenshot shows an example output from running the **RDPSettings** command.



Action Run Command Windows troubleshooting

When troubleshooting action run command for Windows environments, refer to the *RunCommandExtension* log file typically located in the following directory: `C:\WindowsAzure\Logs\Plugins\Microsoft.CPlat.Core.RunCommandWindows\<version>\RunCommandExtension.log` for further details.

Known issues

- Your Action Run Command Extension might fail to execute in your Windows environment if the command contains reserved characters. For example:

  If the `&` symbol is passed in the parameter of your command such as the below PowerShell script, it might fail.

  PowerShell

  ```
  $paramm='abc&jj'
  Invoke-AzVMRunCommand -ResourceGroupName AzureCloudService1 -Name test -
  ```

```
CommandId 'RunPowerShellScript' -ScriptPath
C:\data\228332902\PostAppConfig.ps1 -Parameter @{"Prefix" = $paramm}
```

Use the `^` character to escape the `&` in the argument, such as `$paramm='abc^&jj'`

- The Run Command extension might also fail to execute if command to be executed contains "\n" in the path, as it will be treated as a new line. For example, `C:\Windows\notepad.exe` contains the `\n` in the file path. Consider replacing `\n` with `\N` in your path.

- Ensure you don't have any custom setting in the registry key `HKLM\SOFTWARE\Microsoft\Command Processor\AutoRun` (detailed here). This could trigger during the RunCommand Extension install or enable phases and cause an error like *'XYZ is not recognized as an internal or external command, operable program or batch file*'.

Action Run Command Removal

If needing to remove your action run command Windows extension, refer to the below steps for Azure PowerShell and CLI:

Replace *rgname* and *vmname* with your relevant resource group name and virtual machine name in the following removal examples.

PowerShell

PowerShell

```
Invoke-AzVMRunCommand -ResourceGroupName 'rgname' -VMName 'vmname' -CommandId
'RemoveRunCommandWindowsExtension'
```

> **Note**
>
> When you apply a Run Command again, the extension will get installed automatically. You can use the extension removal command to troubleshoot any issues related to the extension.

Next steps

To learn about other ways to run scripts and commands remotely in your VM, see Run scripts in your Windows VM.

Last updated on 12/13/2025

Run scripts in your Linux VM by using managed Run Commands

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Flexible scale sets

> **Important**
>
> **Managed Run Command** is currently available in Azure CLI, PowerShell, and API at this time. Portal functionality is on the roadmap.

The Run Command feature uses the virtual machine (VM) agent to run scripts within an Azure Linux VM. You can use these scripts for general machine or application management. They can help you quickly diagnose and remediate VM access and network issues and get the VM back to a good state.

The *updated* managed Run Command uses the same VM agent channel to execute scripts and provides the following enhancements over the original action oriented Run Command:

- Support for updated Run Command through ARM deployment template
- Parallel execution of multiple scripts
- Sequential execution of scripts
- User specified script timeout
- Support for long running (hours/days) scripts
- Passing secrets (parameters, passwords) in a secure manner

Prerequisites

> **Important**
>
> The minimum supported Linux Guest Agent is version 2.4.0.2 Older versions do not support managed Run Command. For more information, see Troubleshooting known issues.

Linux Distro's Supported

Expand table

Publisher	Distribution	x64	ARM64
Alma Linux Community	Alma Linux	8.x+, 9.x+	8.x+, 9.x+
Credativ	Debian	10+	11.x+
Kinvolk	Flatcar Linux	3374.2.x+	3374.2.x+
Microsoft	Azure Linux	2.x	2.x
openSUSE Project	openSUSE	12.3+	*Not supported*
Oracle	Oracle Linux	6.4+, 7.x+, 8.x+	*Not supported*
Red Hat	Red Hat Enterprise Linux	6.7+, 7.x+, 8.x+, 9.x+, 10.x+	8.6+, 9.0+, 10.x+
CIQ	Rocky Linux	9.x+	9.x+
SUSE	SLES	12.x+, 15.x+	15.x SP4+
Canonical	Ubuntu (LTS releases)	18.04+, 20.04+, 22.04+, 24.04+	20.04+, 22.04+, 24.04+

Available Commands

Expand table

CommandID	Description
RunShellScript	Runs a Linux shell script.
ifconfig	Gets the configuration of all network interfaces. For usage, refer to Run scripts

> **Note**
>
> The above command IDs are applicable to Managed Run Commands for Linux VMs. You can list all commandIds using command "`Get-AzVMRunCommandDocument -Location {RegionName}`".

Limiting access to Run Command

Listing the run commands or showing the details of a command requires the `Microsoft.Compute/locations/runCommands/read` permission on Subscription level. The built-in

Reader role and higher levels have this permission.

Running a command requires the `Microsoft.Compute/virtualMachines/runCommand/write` permission. The Virtual Machine Contributor role and higher levels have this permission.

You can use one of the built-in roles or create a custom role to use Run Command.

Azure CLI

The following examples use az vm run-command to run shell script on an Azure Linux VM.

Execute a script with the VM

This command delivers the script to the VM, execute it, and return the captured output.

Azure CLI

```
az vm run-command create --name "myRunCommand" --vm-name "myVM" --resource-group
"myRG" --script "echo Hello World!"
```

List all deployed RunCommand resources on a VM

This command returns a full list of previously deployed Run Commands along with their properties.

Azure CLI

```
az vm run-command list --vm-name "myVM" --resource-group "myRG"
```

Get execution status and results

This command retrieves current execution progress, including latest output, start/end time, exit code, and terminal state of the execution.

Azure CLI

```
az vm run-command show --name "myRunCommand" --vm-name "myVM" --resource-group
"myRG" --expand instanceView
```

> ⓘ Note

Output and error fields in `instanceView` is limited to last 4KB. If you'd like to access the full output and error, you have the option of forwarding the output and error data to storage append blobs using `-outputBlobUri` and `-errorBlobUri` parameters while executing Run Command using `Set-AzVMRunCommand` or `Set-AzVMssRunCommand`.

Delete RunCommand resource from the VM

Remove the RunCommand resource previously deployed on the VM. If the script execution is still in progress, execution will be terminated.

Azure CLI

```
az vm run-command delete --name "myRunCommand" --vm-name "myVM" --resource-group "myRG"
```

PowerShell

Execute a script with the VM

This command delivers the script to the VM, execute it, and return the captured output.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM" -Location "EastUS" -RunCommandName "RunCommandName" –SourceScript "echo Hello World!"
```

Execute a script on the VM using SourceScriptUri parameter

`OutputBlobUri` and `ErrorBlobUri` are optional parameters.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName -VMName -RunCommandName -SourceScriptUri “< SAS URI of a storage blob with read access or public URI>" -OutputBlobUri “< SAS URI of a storage append blob with read, add, create, write access>” -ErrorBlobUri “< SAS URI of a storage append blob with read, add, create, write access>”
```

Execute a long running Run Command on a VM

The execution time should run longer than default timeout of 90 minutes. It should run for 100 minutes (refer 6,000-seconds -TimeoutInSecond parameter) provided script is expected to run that long. After 100 minutes, the script execution should stop.

PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG -VMName MyVM -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceScriptUri <SourceScriptUri> -
AsyncExecution -TimeoutInSecond 6000
```

List all deployed RunCommand resources on a VM

This command returns a full list of previously deployed Run Commands along with their properties.

Azure PowerShell

```
Get-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM"
```

Get execution status and results

This command retrieves current execution progress, including latest output, start/end time, exit code, and terminal state of the execution.

Azure PowerShell

```
Get-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM" -RunCommandName
"RunCommandName" -Expand instanceView
```

Create or update Run Command on a VM using SourceScriptURI (storage blob SAS URL)

Create or update Run Command on a Windows VM using a SAS URL of a storage blob that contains a PowerShell script. `SourceScriptUri` can be a storage blob's full SAS URL or public URL.

> **Note**
>
> The SAS URL must provide read access to the blob. An expiration time of 24 hours is suggested for the SAS URL. SAS URLs can be generated on Azure portal using blob's options, or SAS token using `New-AzStorageBlobSASToken`. If generating SAS token using

`New-AzStorageBlobSASToken`, your SAS URL = "base blob URL" + "?" + "SAS token from New-AzStorageBlobSASToken"

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceScriptUri <SourceScriptURI>
```

Get a Run Command Instance View for a VM after Creating or Updating Run Command

Get a Run Command for VM with Instance View. Instance View contains the execution state of Run Command (Succeeded, Failed, etc.), exit code, standard output, and standard error generated by executing the script using Run Command. A nonzero ExitCode indicates an unsuccessful execution.

Azure PowerShell

```
$x = Get-AzVMRunCommand -ResourceGroupName MyRG -VMName MyVM -RunCommandName
MyRunCommand -Expand InstanceView
$x.InstanceView
```

Expected output:

Azure PowerShell

```
ExecutionState   : Succeeded
ExecutionMessage :
ExitCode         : 0
Output           :
output       : uid=0(root) gid=0(root) groups=0(root)
                 HelloWorld

Error            :
StartTime        : 10/27/2022 9:10:52 PM
EndTime          : 10/27/2022 9:10:55 PM
Statuses         :
```

`InstanceView.ExecutionState` -Status of user's Run Command script. Refer this state to know whether your script was successful or not.

`ProvisioningState` - Status of general extension provisioning end to end (whether extension platform was able to trigger Run Command script or not).

Create or update Run Command on a VM using SourceScript (script text)

Create or update Run Command on a VM passing the script content directly to -SourceScript parameter. Use `;` to separate multiple commands.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVML -RunCommandName
MyRunCommand2 -Location EastUS2EUAP -SourceScript "id; echo HelloWorld"
```

Create or update Run Command on a VM using SourceCommandId

Create or update Run Command on a VM using preexisting `commandId`. Available commandIds can be retrieved using Get-AzVMRunCommandDocument.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceCommandId ipconfig
```

Create, or update Run Command on a VM using OutputBlobUri, ErrorBlobUri to stream standard output and standard error messages to output, and error Append blobs

Create or update Run Command on a VM and stream standard output and standard error messages to output and error Append blobs.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVML -RunCommandName
MyRunCommand3 -Location EastUS2EUAP EastUS2EUAP -SourceScriptUri <SourceScriptUri>
-OutputBlobUri <OutputBlobUri> -ErrorBlobUri <errorbloburi>
```

> **Note**
>
> Output and error blobs must be of type `AppendBlob` and their SAS URLs must provide read, append, create, write access to the blob. An expiration time of 24 hours is suggested for the SAS URL. If output or error blob does not exist, a blob of type AppendBlob will be created. SAS URLs can be generated on Azure portal using blob's options, or SAS token

using New-AzStorageBlobSASToken. If generating SAS token using `New-AzStorageBlobSASToken`, SAS URL = base blob URL + "?" + SAS token from `New-AzStorageBlobSASToken`.

Create or update Run Command on a VM, run the Run Command as a different user using RunAsUser and RunAsPassword parameters

Create or update Run Command on a VM, run the Run Command as a different user using `RunAsUser` and `RunAsPassword` parameters. For RunAs to work properly, contact admin of VM, and make sure user is added on the VM, user has access to resources accessed by the Run Command (Directories, Files, Network etc.), and in case of Windows VM, 'Secondary Logon' service is running on the VM.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP EastUS2EUAP -SourceScriptUri <SourceScriptUri> -
RunAsUser myusername -RunAsPassword mypassword
```

Create or update Run Command on a Virtual Machine Scale Sets resource using SourceScriptUri (storage blob SAS URL).

Create or update Run Command on a Virtual Machine Scale Sets resource using a SAS URL of a storage blob that contains a bash script.

Azure PowerShell

```
Set-AzVmssVMRunCommand -ResourceGroupName MyRG0 -VMScaleSetName MyVMSS -InstanceId
0 -RunCommandName MyRunCommand -Location EastUS2EUAP -SourceScriptUri
<SourceScriptUri>
```

> **Note**
>
> Note SAS URL must provide read access to the blob. An expiration time of 24 hours is suggested for SAS URL. SAS URLs can be generated on Azure portal using blob's options, or SAS token using New-AzStorageBlobSASToken. If generating SAS token using New-AzStorageBlobSASToken, SAS URL = base blob URL + "?" + SAS token from `New-AzStorageBlobSASToken`.

Create or update Run Command on a VM instance using Parameter and ProtectedParameter parameters (Public and Protected Parameters to script)

Azure PowerShell

```
$PublicParametersArray =
@([Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInputPa
rameter]@{name='publicParam1';value='publicParam1value'},
>>
[Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInputPara
meter]@{name='publicParam2';value='publicParam2value'})

$ProtectedParametersArray =
@([Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInputPa
rameter]@{name='secret1';value='secret1value'},
>>
[Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInputPara
meter]@{name='secret2';value='secret2value'})

Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceScriptUri <SourceScriptUri> -Parameter
$PublicParametersArray -ProtectedParameter $ProtectedParametersArray
```

Use `ProtectedParameter` to pass any sensitive inputs to script such as passwords, keys etc.

- Windows: Parameters and ProtectedParameters are passed to script as arguments are passed to script and run like this `myscript.ps1 -publicParam1 publicParam1value -publicParam2 publicParam2value -secret1 secret1value -secret2 secret2value`
- Linux: Named Parameters and its values are set to environment config, which should be accessible within the `.sh` script. For Nameless arguments, pass an empty string to name input. Nameless arguments are passed to script and run like this - `myscript.sh publicParam1value publicParam2value secret1value secret2value`

Delete RunCommand resource from the VM

Remove the RunCommand resource previously deployed on the VM. If the script execution is still in progress, execution is terminated.

Azure PowerShell

```
Remove-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM" -RunCommandName
"RunCommandName"
```

REST API

To deploy a new Run Command, execute a PUT on the VM directly and specify a unique name for the Run Command instance.

rest

```
PUT
/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Micros
oft.Compute/virtualMachines/<vmName>/runcommands?api-version=2019-12-01
```

JSON

```
{
"location": "<location>",
"properties": {
    "source": {
        "script": "Write-Host Hello World!",
        "scriptUri": "<SAS URI of a storage blob with read access or public URI>",
        "commandId": "<Id>"
        },
    "parameters": [
        {
            "name": "param1",
            "value": "value1"
            },
        {
            "name": "param2",
            "value": "value2"
            }
        ],
    "protectedParameters": [
        {
            "name": "secret1",
            "value": "value1"
            },
        {
            "name": "secret2",
            "value": "value2"
            }
        ],
    "runAsUser": "userName",
    "runAsPassword": "userPassword",
    "timeoutInSeconds": 3600,
    "outputBlobUri": "< SAS URI of a storage append blob with read, add, create,
write access>",
    "errorBlobUri": "< SAS URI of a storage append blob with read, add, create,
write access >"
    }
}
```

Notes

- You can provide an inline script, a script URI, or a built-in script command ID as the input source. Script URI is either storage blob SAS URI with read access or public URI.
- Only one type of source input is supported for one command execution.
- Run Command supports writing output and error to Storage blobs using outputBlobUri and errorBlobUri parameters, which can be used to store large script outputs. Use SAS URI of a storage append blob with read, add, create, write access. The blob should be of type AppendBlob. Writing the script output or error blob would fail otherwise. The blob will be overwritten if it already exists. It will be created if it does not exist.

List running instances of Run Command on a VM

rest

```
GET
/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Micros
oft.Compute/virtualMachines/<vmName>/runcommands?api-version=2019-12-01
```

Get output details for a specific Run Command deployment

rest

```
GET
/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Micros
oft.Compute/virtualMachines/<vmName>/runcommands/<runCommandName>?
$expand=instanceView&api-version=2019-12-01
```

Delete a specific Run Command deployment

Delete the instance of Run Command

rest

```
DELETE
/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Micros
oft.Compute/virtualMachines/<vmName>/runcommands/<runCommandName>?api-version=2019-
12-01
```

Deploy scripts in an ordered sequence

To deploy scripts sequentially, use a deployment template, specifying a `dependsOn` relationship between sequential scripts.

JSON

```
{
    "type":"Microsoft.Compute/virtualMachines/runCommands",
    "name":"secondRunCommand",
    "apiVersion":"2019-12-01",
    "location":"[parameters('location')]",
    "dependsOn":<full resourceID of the previous other Run Command>,
    "properties":{
        "source":{
            "script":"echo Hello World!"
        },
        "timeoutInSeconds":60
    }
}
```

Execute multiple Run Commands sequentially

By default, if you deploy multiple RunCommand resources using deployment template, they will be executed simultaneously on the VM. If you have a dependency on the scripts and a preferred order of execution, you can use the `dependsOn` property to make them run sequentially.

In this example, **secondRunCommand** will execute after **firstRunCommand**.

JSON

```
{
   "$schema":"https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
   "contentVersion":"1.0.0.0",
   "resources":[
      {
         "type":"Microsoft.Compute/virtualMachines/runCommands",
         "name":"[concat(parameters('vmName'),'/firstRunCommand')]",
         "apiVersion":"2019-12-01",
         "location":"[parameters('location')]",
         "dependsOn":[
            "[concat('Microsoft.Compute/virtualMachines/', parameters('vmName'))]"
         ],
         "properties":{
            "source":{
               "script":"echo First: Hello World!"
            },
            "parameters":[
               {
```

```
                "name":"param1",
                "value":"value1"
             },
             {
                "name":"param2",
                "value":"value2"
             }
          ],
          "timeoutInSeconds":20
       }
    },
    {
       "type":"Microsoft.Compute/virtualMachines/runCommands",
       "name":"[concat(parameters('vmName'),'/secondRunCommand')]",
       "apiVersion":"2019-12-01",
       "location":"[parameters('location')]",
       "dependsOn":[
          "[concat('Microsoft.Compute/virtualMachines/',
parameters('vmName'),'runcommands/firstRunCommand')]"
       ],
       "properties":{
          "source":{
             "scriptUri":"http://github.com/myscript.ps1"
          },
          "timeoutInSeconds":60
       }
    }
 ]
}
```

Troubleshooting known issues

Unsupported Linux guest agent version

Issue: If you deploy a VM with managed Run Command when the Linux Guest Agent version is 2.4.0.1 or earlier, managed Run Command doesn't execute.

Cause: Managed Run Command requires Linux Guest Agent version 2.4.0.2 or later to install and execute.

Mitigation: Upgrade Linux Guest Agent to version 2.4.0.2 or later before you use managed Run Command. Alternatively, use an image that already has a baked-in Linux Guest Agent version 2.4.0.2 or later.

To check the Linux Guest Agent version, run:

Azure CLI

```
az vm get-instance-view --resource-group $RGName$ --name $VMName$ --subscription
$SUBSCRIPTION --query 'instanceView.vmAgent.vmAgentVersion'
```

Next steps

To learn about other ways to run scripts and commands remotely in your VM, see Run scripts in your Linux VM.

Last updated on 02/10/2026

Run scripts in your Windows VM by using managed Run Commands

Applies to: ✔ Windows VMs ✔ Flexible scale sets

The Run Command feature uses the virtual machine (VM) agent to execute scripts within an Azure Windows VM. You can use these scripts for general machine or application management. They can help you quickly diagnose and remediate VM access and network issues and get the VM back to a good state.

The *updated* managed Run Command uses the same VM agent channel to execute scripts and provides the following enhancements over the original action orientated Run Command:

- Support for updated Run Command through ARM deployment template
- Parallel execution of multiple scripts
- Sequential execution of scripts
- User specified script time-out
- Support for long running (hours/days) scripts
- Passing secrets (parameters, passwords) in a secure manner

> **Important**
>
> **Managed Run Command** is currently available in Azure CLI, PowerShell, and API at this time.

Prerequisites

Windows Operating Systems Supported

Expand table

OS Version	x64	ARM64
Windows 10	Supported	Supported
Windows 11	Supported	Supported
Windows Server 2016	Supported	Supported
Windows Server 2016 Core	Supported	Supported
Windows Server 2019	Supported	Supported

OS Version	x64	ARM64
Windows Server 2019 Core	Supported	Supported
Windows Server 2022	Supported	Supported
Windows Server 2022 Core	Supported	Supported
Windows Server 2022 Azure Edition	Supported	Supported
Windows Server 2025	Supported	Supported
Windows Server 2025 Core	Supported	Supported
Windows Server 2025 Azure Edition	Supported	Supported

Available Commands

Expand table

CommandID	Description	More Info
RunPowerShellScript	Runs a PowerShell script	
DisableNLA	Disables Network Level Authentication (NLA). You must restart the VM after the script completes for the change to take effect. The script itself does not restart the VM.	readme
DisableWindowsUpdate	Disable Automatic Updates through Windows Update.	readme
EnableAdminAccount	Checks if the local Administrator account is disabled, and if so enables it.	readme
EnableEMS	Enable Emergency Management Services (EMS) to allow for serial console connection in troubleshooting scenarios.	readme
EnableRemotePS	Enable remote PowerShell.	readme
EnableWindowsUpdate	Enable Automatic Updates through Windows Update.	readme
IMDSCertCheck	Checks IMDS Health and then analyzes currently installed certificates that IMDS depends upon. If missing, shows additional details and mitigation steps.	readme

CommandID	Description	More Info
IPConfig	Shows detailed information for the IP address, subnet mask and default gateway for each adapter bound to TCP/IP. For usage, refer to Run scripts	readme
RDPSettings	Checks registry settings and domain policy settings. Suggests policy actions if machine is part of a domain or modifies the settings to default values.	readme
ResetRDPCert	Removes the SSL certificate tied to the RDP listener and restores the RDP listener security to default. Use this script if you see any issues with the certificate.	readme
SetRDPPort	Sets the default or user specified port number for Remote Desktop connections. Enables firewall rule for inbound access to the port.	readme
WindowsActivationValidation	Checks the current Windows license status (activated or not) and alert is generated if the machine is not property activated.	readme
WindowsGhostedNicValidationScript	This script scans the Windows registry for network adapters on PCI and VMBUS buses, compares them with currently active network adapters, and identifies ghosted NICs. Useful for troubleshooting network issues or cleaning up old NICs.	readme
WindowsUpgradeAssessmentValidation	This script is designed to assess the readiness of a Windows machine (client or server) for an in-place OS upgrade, with special considerations for Azure Virtual Machines. It evaluates OS version, supported upgrade paths, system disk space, and Azure security features like Trusted Launch, Secure Boot, and vTPM.	readme

> **Note**
>
> The above command IDs are applicable to Managed Run Commands for Windows VMs. You can list all commandIds using command "`Get-AzVMRunCommandDocument -Location {RegionName}`".

Limiting access to Run Command

Listing the run commands or showing the details of a command requires the `Microsoft.Compute/locations/runCommands/read` permission on Subscription Level. The built-in Reader role and higher levels have this permission.

Running a command requires the `Microsoft.Compute/virtualMachines/runCommand/write` permission. The Virtual Machine Contributor role and higher levels have this permission.

You can use one of the built-in roles or create a custom role to use Run Command.

Using Run Commands

PowerShell

Execute a script with the VM

This command delivers the script to the VM, execute it, and return the captured output.

PowerShell

```
Set-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM" -Location "EastUS" -
RunCommandName "RunCommandName" –SourceScript "echo Hello World!"
```

Execute a script on the VM using SourceScriptUri parameter

`OutputBlobUri` and `ErrorBlobUri` are optional parameters.

PowerShell

```
Set-AzVMRunCommand -ResourceGroupName "myRg" `
-VMName "myVM" `
-RunCommandName "RunCommandName" `
-SourceScriptUri “<SAS_URI_of_a_storage_blob_with_read_access_or_public_URI>" `
-OutputBlobUri
“<SAS_URI_of_a_storage_append_blob_with_read_add_create_write_access>" `
-ErrorBlobUri
“<SAS_URI_of_a_storage_append_blob_with_read_add_create_write_access>”
```

Execute a long running Run Command on a VM

The execution time should run longer than default timeout of 90 minutes. It should run for 100 minutes (refer 6,0000-seconds -TimeoutInSecond parameter) provided script is expected to run that long. After 100 minutes, the script execution should stop.

PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG -VMName MyVM -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceScriptUri <SourceScriptUri> -
AsyncExecution -TimeoutInSecond 6000
```

List all deployed RunCommand resources on a VM

This command returns a full list of previously deployed Run Commands along with their properties.

PowerShell

```
Get-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM"
```

Get execution status and results

This command retrieves current execution progress, including latest output, start/end time, exit code, and terminal state of the execution.

PowerShell

```
Get-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM" -RunCommandName
"RunCommandName" -Expand InstanceView
```

Create or update Run Command on a VM using SourceScriptUri (storage blob SAS URL)

Create or update Run Command on a Windows VM using a SAS URL of a storage blob that contains a PowerShell script. `SourceScriptUri` can be a storage blob's full SAS URL or public URL.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceScriptUri <SourceScriptUri>
```

ⓘ Note

SAS URL must provide read access to the blob. An expiration time of 24 hours is suggested for SAS URL. SAS URLs can be generated on Azure portal using blob's options , or SAS token using `New-AzStorageBlobSASToken`. If generating SAS token using `New-AzStorageBlobSASToken`, your SAS URL = "base blob URL" + "?" + "SAS token from `New-AzStorageBlobSASToken`"

Get a Run Command Instance View for a VM after Creating or Updating Run Command

Get a Run Command for VM with Instance View. Instance View contains execution state of run command (Succeeded, Failed, etc.), exit code, standard output, and standard error generated by executing the script using Run Command. A nonzero ExitCode indicates an unsuccessful execution.

Azure PowerShell

```
$x = Get-AzVMRunCommand -ResourceGroupName MyRG -VMName MyVM -RunCommandName
MyRunCommand -Expand InstanceView
$x.InstanceView
```

Example output

Azure PowerShell

```
ExecutionState   : Succeeded
ExecutionMessage :
ExitCode         : 0
Output           :
output       : uid=0(root) gid=0(root) groups=0(root)
                 HelloWorld

Error            :
StartTime        : 10/27/2022 9:10:52 PM
EndTime          : 10/27/2022 9:10:55 PM
Statuses         :
```

`InstanceView.ExecutionState`: Status of user's Run Command script. Refer this state to know whether your script was successful or not. `ProvisioningState`: Status of general extension provisioning end to end (whether extension platform was able to trigger Run Command script or not).

Create or update Run Command on a VM using ScriptLocalPath (local script file)

Create or update Run Command on a VM using a local script file that is on the client machine where cmdlet is executed.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -ScriptLocalPath
"C:\MyScriptsDir\MyScript.ps1"
```

Create or update Run Command on a VM using SourceScript (script text)

Create or update Run Command on a VM passing the script content directly to -SourceScript parameter. Use `;` to separate multiple commands

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVML -RunCommandName
MyRunCommand2 -Location EastUS2EUAP -SourceScript "id; echo HelloWorld"
```

Create or update Run Command on a VM using SourceCommandId

Create or update Run Command on a VM using preexisting commandId. Available commandIds can be retrieved using Get-AzVMRunCommandDocument.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceCommandId DisableWindowsUpdate
```

Create, or update Run Command on a VM using OutputBlobUri, ErrorBlobUri to stream standard output and standard error messages to output, and error Append blobs

Create or update Run Command on a VM and stream standard output and standard error messages to output and error Append blobs.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVML -RunCommandName
MyRunCommand3 -Location EastUS2EUAP -ScriptLocalPath
"C:\MyScriptsDir\MyScript.ps1" -OutputBlobUri <OutPutBlobUrI> -ErrorBlobUri
"ErrorBlobUri
```

> ⓘ Note
>
> Output and error blobs must be the AppendBlob type and their SAS URLs must provide read, append, create, write access to the blob. An expiration time of 24 hours is suggested for SAS URL. If output or error blob does not exist, a blob of type AppendBlob is created. SAS URLs can be generated on Azure portal using blob's options , or SAS token from using `New-AzStorageBlobSASToken`.

Create or update Run Command on a VM as a different user using RunAsUser and RunAsPassword parameters

Create or update Run Command on a VM as a different user using `RunAsUser` and `RunAsPassword` parameters. For RunAs to work properly, contact admin of VM, and make sure user is added on the VM, user has access to resources accessed by the Run Command (Directories, Files, Network etc.), and in case of Windows VM, 'Secondary Logon' service is running on the VM.

Azure PowerShell

```
Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -ScriptLocalPath
"C:\MyScriptsDir\MyScript.ps1" -RunAsUser myusername -RunAsPassword mypassword
```

Create or update Run Command on a Virtual Machine Scale Sets resource using SourceScriptUri (storage blob SAS URL)

Create or update Run Command on a Windows Virtual Machine Scale Sets resource using a SAS URL of a storage blob that contains a PowerShell script.

Azure PowerShell

```
Set-AzVmssVMRunCommand -ResourceGroupName MyRG0 -VMScaleSetName MyVMSS -
InstanceId 0 -RunCommandName MyRunCommand -Location EastUS2EUAP -SourceScriptUri
<SourceScriptUri>
```

> ⓘ Note
>
> SAS URL must provide read access to the blob. An expiry time of 24 hours is suggested for SAS URL. SAS URLs can be generated on Azure portal using blob's options, or SAS token using `New-AzStorageBlobSASToken`. If generating SAS token

using `New-AzStorageBlobSASToken`, the SAS URL format is: base blob URL + "?" + the SAS token from `New-AzStorageBlobSASToken`.

Create or update Run Command on a VM instance using Parameter and ProtectedParameter parameters (Public and Protected Parameters to script)

Use ProtectedParameter to pass any sensitive inputs to script such as passwords, keys etc.

Azure PowerShell

```
$PublicParametersArray =
@([Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInpu
tParameter]@{name='publicParam1';value='publicParam1value'},
>>
[Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInputP
arameter]@{name='publicParam2';value='publicParam2value'})

$ProtectedParametersArray =
@([Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInpu
tParameter]@{name='secret1';value='secret1value'},
>>
[Microsoft.Azure.PowerShell.Cmdlets.Compute.Models.Api20210701.IRunCommandInputP
arameter]@{name='secret2';value='secret2value'})

Set-AzVMRunCommand -ResourceGroupName MyRG0 -VMName MyVMEE -RunCommandName
MyRunCommand -Location EastUS2EUAP -SourceScriptUri <SourceScriptUri> -Parameter
$PublicParametersArray -ProtectedParameter $ProtectedParametersArray
```

- Windows: Parameters and ProtectedParameters are passed to script as arguments are passed to script and run like this - `myscript.ps1 -publicParam1 publicParam1value -publicParam2 publicParam2value -secret1 secret1value -secret2 secret2value`
- Linux: Named Parameters and its values are set to environment config, which should be accessible within the .sh script. For Nameless arguments, pass an empty string to name input. Nameless arguments are passed to script and run like this - `myscript.sh publicParam1value publicParam2value secret1value secret2value`

Delete RunCommand resource from the VM

Remove the RunCommand resource previously deployed on the VM. If the script execution is still in progress, execution is terminated.

PowerShell

```
Remove-AzVMRunCommand -ResourceGroupName "myRG" -VMName "myVM" -RunCommandName
"RunCommandName"
```

Execute multiple Run Commands sequentially

By default, if you deploy multiple RunCommand resources using deployment template, they will be executed simultaneously on the VM. If you have a dependency on the scripts and a preferred order of execution, you can use the `dependsOn` property to make them run sequentially.

In this example, **secondRunCommand** will execute after **firstRunCommand**.

JSON

```
{
   "$schema":"https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
   "contentVersion":"1.0.0.0",
   "resources":[
      {
         "type":"Microsoft.Compute/virtualMachines/runCommands",
         "name":"[concat(parameters('vmName'),'/firstRunCommand')]",
         "apiVersion":"2023-03-01",
         "location":"[parameters('location')]",
         "dependsOn":[
            "[concat('Microsoft.Compute/virtualMachines/', parameters('vmName'))]"
         ],
         "properties":{
            "source":{
               "script":"Write-Host First: Hello World!"
            },
            "parameters":[
               {
                  "name":"param1",
                  "value":"value1"
               },
               {
                  "name":"param2",
                  "value":"value2"
               }
            ],
            "timeoutInSeconds":20
         }
      },
      {
         "type":"Microsoft.Compute/virtualMachines/runCommands",
         "name":"[concat(parameters('vmName'),'/secondRunCommand')]",
         "apiVersion":"2019-12-01",
         "location":"[parameters('location')]",
         "dependsOn":[
```

```
            "[concat('Microsoft.Compute/virtualMachines/',
parameters('vmName'),'runcommands/firstRunCommand')]"
         ],
         "properties":{
            "source":{
               "scriptUri":"http://github.com/myscript.ps1"
            },
            "timeoutInSeconds":60
         }
      }
   ]
}
```

Next steps

To learn about other ways to run scripts and commands remotely in your VM, see Run scripts in your Windows VM.

Last updated on 12/13/2025

Azure virtual machine extensions and features

Extensions are small applications that provide post-deployment configuration and automation on Azure virtual machines (VMs). The Azure platform hosts many extensions covering VM configuration, monitoring, security, and utility applications. Publishers take an application, wrap it into an extension, and simplify the installation. All you need to do is provide mandatory parameters.

> **Note**
>
> Extensions are only supported on Endorsed Linux distributions on Azure.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

View available extensions

You can view available extensions for a VM in the Azure portal.

1. In the portal, go to the **Overview** page for a VM.
2. Under **Settings**, select **Extensions + Applications**.

The list of available extensions are displayed. To see the complete list of extensions, see Discovering VM Extensions for Linux and Discovering VM Extensions for Windows.

Install and use extensions

Azure VM extensions can be managed by using the Azure CLI, PowerShell, Azure Resource Manager (ARM) templates, and the Azure portal.

1. From the **Extensions + Applications** for the VM, on the **Extensions** tab, select **+ Add**.
2. Locate the **Custom Script Extension** option. Select the extension option, then select **Next**.

You can then pass in a command or script to run the extension.

For more information, see Linux Custom Script Extension and Windows Custom Script Extension.

Check for prerequisites

Some individual VM extension applications might have their own environmental prerequisites, such as access to an endpoint. Each extension has an article that explains any prerequisites, including which operating systems are supported.

Manage extension application lifecycle

You don't need to connect to a VM directly to install or delete an extension. The Azure extension lifecycle is managed outside of the VM and integrated into the Azure platform.

Automatic extension upgrade can be used to monitor for new minor versions and automatically upgrade extensions in a safe rolling manner.

Troubleshoot extensions

If you're looking for general troubleshooting steps for Windows VM extensions, refer to Troubleshooting Azure Windows VM extension failures .

Otherwise, specific troubleshooting information for each extension can be found in the **Troubleshoot and support** section in the overview for the extension. Here's a list of the troubleshooting information available:

Expand table

Namespace	Troubleshooting
microsoft.azure.monitoring.dependencyagent.dependencyagentlinux	Azure Monitor Dependency for Linux
microsoft.azure.monitoring.dependencyagent.dependencyagentwindows	Azure Monitor Dependency for Windows
microsoft.azure.security.azurediskencryptionforlinux	Azure Disk Encryption for Linux
microsoft.azure.security.azurediskencryption	Azure Disk Encryption for Windows
microsoft.compute.customscriptextension	Custom Script for Windows
microsoft.ostcextensions.customscriptforlinux	

Namespace	Troubleshooting
microsoft.powershell.dsc	Desired State Configuration for Windows
microsoft.hpccompute.nvidiagpudriverlinux	NVIDIA GPU Driver Extension for Linux
microsoft.hpccompute.nvidiagpudriverwindows	NVIDIA GPU Driver Extension for Windows
microsoft.azure.security.iaasantimalware	Antimalware Extension for Windows
microsoft.enterprisecloud.monitoring.omsagentforlinux	Azure Monitor for Linux
microsoft.enterprisecloud.monitoring.microsoftmonitoringagent	Azure Monitor for Windows
stackify.linuxagent.extension.stackifylinuxagentextension	Stackify Retrace for Linux
vmaccessforlinux.microsoft.ostcextensions	VMAccess for Linux
microsoft.recoveryservices.vmsnapshot	Snapshot for Linux
microsoft.recoveryservices.vmsnapshot	Snapshot for Windows

Next steps

- For more information about how the Linux Agent and extensions work, see Azure VM extensions and features for Linux.
- For more information about how the Windows Guest Agent and extensions work, see Azure VM extensions and features for Windows.
- To install the Linux Agent, see Azure Linux Virtual Machine Agent overview.
- To install the Windows Guest Agent, see Azure Windows Virtual Machine Agent overview.
- To automatically update the extensions to latest version, see Automatic Extension Upgrade for VMs and Scale Sets in Azure

Last updated on 11/10/2025

Virtual machine extensions and features for Linux

Azure virtual machine (VM) extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs. For example, if a VM requires software installation, antivirus protection, or the ability to run a script inside it, you can use a VM extension.

You can run Azure VM extensions by using the Azure CLI, Azure PowerShell, Azure Resource Manager templates (ARM templates), and the Azure portal. You can bundle extensions with a new VM deployment or run them against any existing system.

This article provides an overview of Azure VM extensions, prerequisites for using them, and guidance on how to detect, manage, and remove them. This article provides general information because many VM extensions are available. Each one has a potentially unique configuration and its own documentation.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Use cases and samples

Each Azure VM extension has a specific use case. Examples include:

- Configure monitoring of a VM by using the Microsoft Monitoring Agent VM extension.
- Configure monitoring of your Azure infrastructure by using the Chef or Datadog extension.

In addition to process-specific extensions, a Custom Script extension is available for both Windows and Linux VMs. The Custom Script extension for Linux allows any Bash script to be run on a VM. Custom scripts are useful for designing Azure deployments that require configuration beyond what native Azure tooling can provide.

Prerequisites

> **Note**

Extensions are supported only on Endorsed Linux distributions on Azure.

Azure Linux Agent

To handle the extension on the VM, you must have the Azure Linux Agent installed. Some individual extensions have prerequisites, such as access to resources or dependencies.

The Azure Linux Agent manages interactions between an Azure VM and the Azure fabric controller. The agent is responsible for many functional aspects of deploying and managing Azure VMs, including running VM extensions.

The Azure Linux Agent is preinstalled on Azure Marketplace images. You can also install it manually on supported operating systems.

The agent runs on multiple operating systems. However, the extensions framework has a limit for the operating systems that extensions use. Some extensions aren't supported across all operating systems and might emit error code 51 ("Unsupported OS"). Check the individual extension documentation for supportability.

Network access

Extension packages are downloaded from the Azure Storage extension repository. Extension status uploads are posted to Azure Storage.

If you use a supported version of the Azure Linux Agent, you don't need to allow access to Azure Storage in the VM region. You can use the agent to redirect the communication to the Azure fabric controller for agent communications. If you're on an unsupported version of the agent, you need to allow outbound access to Azure Storage in that region from the VM.

> **Important**
>
> If you block access to the private IP address 168.63.129.16 by using the guest firewall, extensions fail even if you use a supported version of the agent or you configure outbound access. Also, if the VM has no outbound access to `*.blob.windows.net` and `*.blob.storage.azure.net`, initialization of the Azure Linux Agent and installation of extensions incurs more delays. To avoid those delays, ensure that access to these endpoints is allowed.

You can use agents only to download extension packages and reporting status. For example, if an extension installation needs to download a script from GitHub (Custom Script extension) or needs access to Azure Storage (Azure Backup), you need to open more firewall or network

security group (NSG) ports. Different extensions have different requirements because they're applications. For extensions that require access to Azure Storage, you can allow access by using Azure NSG service tags.

To redirect agent traffic requests, the Azure Linux Agent has proxy server support. This proxy server support doesn't apply extensions. You must configure each individual extension to work with a proxy.

Discover VM extensions

Azure CLI

Many VM extensions are available for use with Azure VMs. To see a complete list, use az vm extension image list. The following example lists all available extensions in the *westus* location:

Azure CLI

```
az vm extension image list --location westus --output table
```

Run VM extensions

Azure VM extensions run on existing VMs. That capability is useful when you need to make configuration changes or recover connectivity on an already deployed VM. VM extensions can also be bundled with ARM template deployments. By using extensions with ARM templates, you can deploy and configure Azure VMs without post-deployment intervention.

You can use the following methods to run an extension against an existing VM.

Azure CLI

You can run Azure VM extensions against an existing VM by using the az vm extension set command. The following example runs the Custom Script extension against a VM named *myVM* in a resource group named *myResourceGroup*. Replace the example resource group name, VM name, and script to run `https:\//raw.githubusercontent.com/me/project/hello.sh` with your own information.

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
```

```
  --vm-name myVM \
  --name customScript \
  --publisher Microsoft.Azure.Extensions \
  --settings '{"fileUris":
["https://raw.githubusercontent.com/me/project/hello.sh"],"commandToExecute":
"./hello.sh"}'
```

When the extension runs correctly, the output is similar to the following example:

Output

```
info:    Executing command vm extension set
+ Looking up the VM "myVM"
+ Installing extension "CustomScript", VM: "mvVM"
info:    vm extension set command OK
```

Azure PowerShell

You can run Azure VM extensions against an existing VM by using the Set-AzVMExtension command. The following example runs the Custom Script extension against a VM named *myVM* in a resource group named *myResourceGroup*. Replace the example resource group name, VM name, and script to run `https:\//raw.githubusercontent.com/me/project/hello.sh` with your own information.

Azure PowerShell

```
$Params = @{
    ResourceGroupName  = 'myResourceGroup'
    VMName             = 'myVM'
    Name               = 'CustomScript'
    Publisher          = 'Microsoft.Azure.Extensions'
    ExtensionType      = 'CustomScript'
    TypeHandlerVersion = '2.1'
    Settings          = @{fileUris =
@('https://raw.githubusercontent.com/me/project/hello.sh'); commandToExecute =
'./hello.sh'}
}
Set-AzVMExtension @Params
```

When the extension runs correctly, the output is similar to the following example:

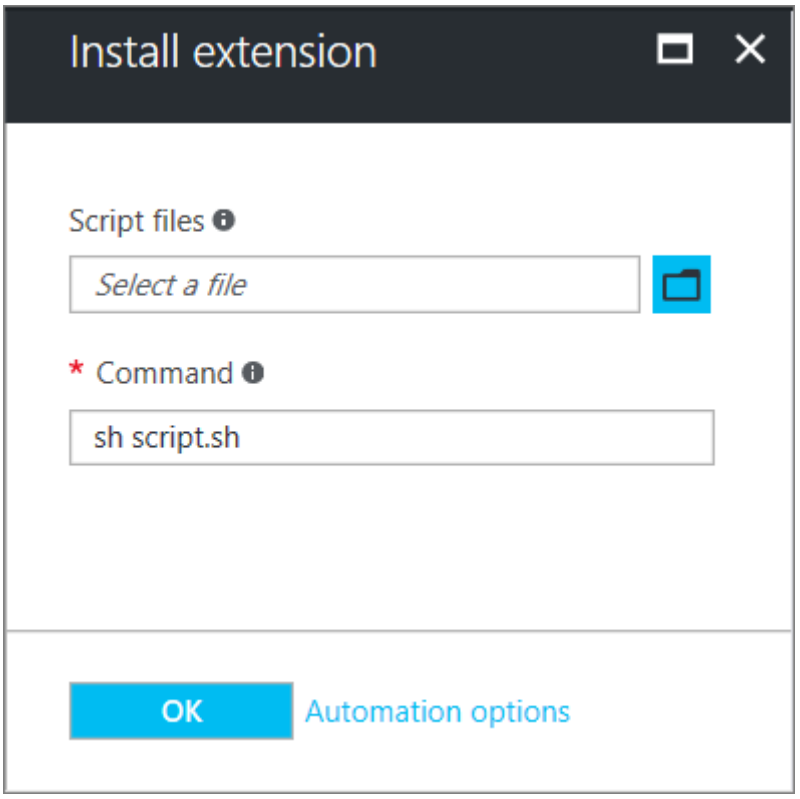
Output

```
RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

Azure portal

You can apply VM extensions to an existing VM through the Azure portal. Select the VM in the portal, select **Extensions**, and then select **Add**. Choose the extension that you want from the list of available extensions, and follow the instructions in the wizard.

The following image shows the installation of the Custom Script extension for Linux from the Azure portal.



Azure Resource Manager templates

You can add VM extensions to an ARM template and run them with the deployment of the template. When you deploy an extension with a template, you can create fully configured Azure deployments.

For example, the following JSON is taken from a full ARM template that deploys a set of load-balanced VMs and an Azure SQL database. Then it installs a .NET Core application on each VM. The VM extension takes care of the software installation.

JSON

```
{
    "apiVersion": "2015-06-15",
    "type": "extensions",
    "name": "config-app",
    "location": "[resourceGroup().location]",
    "dependsOn": [
```

```
    "[concat('Microsoft.Compute/virtualMachines/',
concat(variables('vmName'),copyindex()))]"
    ],
    "tags": {
    "displayName": "config-app"
    },
    "properties": {
    "publisher": "Microsoft.Azure.Extensions",
    "type": "CustomScript",
    "typeHandlerVersion": "2.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-linux/scripts/config-music.sh"
        ]
    },
    "protectedSettings": {
        "commandToExecute": "[concat('sudo sh config-music.sh
',variables('musicStoreSqlName'), ' ', parameters('adminUsername'), ' ',
parameters('sqlAdminPassword'))]"
    }
    }
}
```

For more information on how to create ARM templates, see Virtual machines in an Azure Resource Manager template.

Help secure VM extension data

When you run a VM extension, it might be necessary to include sensitive information such as credentials, storage account names, and access keys. Many VM extensions include a protected configuration that encrypts data and only decrypts it inside the target VM. Each extension has a specific protected configuration schema. Each extension is detailed in extension-specific documentation.

The following example shows an instance of the Custom Script extension for Linux. The command to run includes a set of credentials. In this example, the command to run isn't encrypted.

JSON

```
{
  "apiVersion": "2015-06-15",
  "type": "extensions",
  "name": "config-app",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/',
```

```
concat(variables('vmName'),copyindex()))]"
  ],
  "tags": {
    "displayName": "config-app"
  },
  "properties": {
    "publisher": "Microsoft.Azure.Extensions",
    "type": "CustomScript",
    "typeHandlerVersion": "2.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-linux/scripts/config-music.sh"
      ],
      "commandToExecute": "[concat('sudo sh config-music.sh
',variables('musicStoreSqlName'), ' ', parameters('adminUsername'), ' ',
parameters('sqlAdminPassword'))]"
    }
  }
}
```

Moving the `commandToExecute` property to the `protected` configuration helps secure the execution string, as shown in the following example:

JSON

```
{
  "apiVersion": "2015-06-15",
  "type": "extensions",
  "name": "config-app",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/',
concat(variables('vmName'),copyindex()))]"
  ],
  "tags": {
    "displayName": "config-app"
  },
  "properties": {
    "publisher": "Microsoft.Azure.Extensions",
    "type": "CustomScript",
    "typeHandlerVersion": "2.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-linux/scripts/config-music.sh"
      ]
    },
    "protectedSettings": {
      "commandToExecute": "[concat('sudo sh config-music.sh
',variables('musicStoreSqlName'), ' ', parameters('adminUsername'), ' ',
```

```
parameters('sqlAdminPassword'))]"
    }
  }
}
```

How agents and extensions are updated

Agents and extensions share the same automatic update mechanism.

When an update is available and automatic updates are enabled, the update is installed on the VM only after there's a change to an extension or after other VM model changes, such as:

- Data disks
- Extensions
- Extension tags
- Boot diagnostics container
- Guest operating system secrets
- VM size
- Network profile

Publishers make updates available to regions at various times, so you could have VMs in different regions on different versions.

> **Note**
>
> Some updates might require extra firewall rules. For more information, see **Network access**.

Agent updates

The Linux VM Agent contains *Provisioning Agent code* and *extension-handling code* in one package. They can't be separated.

You can disable the Provisioning Agent when you want to provision on Azure by using cloud-init.

Supported versions of the agents can use automatic updates. The only code that can be updated is the extension-handling code, not the Provisioning Agent code. The Provisioning Agent code is run-once code.

The extension-handling code is responsible for:

- Communicating with the Azure fabric.

- Handling the VM extension operations, such as installations, reporting status, updating the individual extensions, and removing extensions. Updates contain security fixes, bug fixes, and enhancements to the extension-handling code.

When the agent is installed, a parent daemon is created. This parent then spawns a child process that handles extensions. If an update is available for the agent, it's downloaded. The parent stops the child process, upgrades it, and then restarts it. If a problem occurs with the update, the parent process rolls back to the previous child version.

The parent process can't be automatically updated. Only a distribution package can update the parent.

To check what version you're running, check `waagent`:

Bash

```
waagent --version
```

The output is similar to the following example:

Output

```
WALinuxAgent-2.2.45 running on <Linux Distro>
Python: 3.6.9
Goal state agent: 2.7.1.0
```

In the preceding example output, the parent (or package deployed version) is `WALinuxAgent-2.2.45`. The `Goal state agent` value is the auto-update version.

We highly recommend that you always enable automatic update for the agent: AutoUpdate.Enabled=y. If you don't enable automatic update, you need to keep manually updating the agent. You also don't get bug and security fixes.

Extension updates

When an extension update is available and automatic updates are enabled, after a change to the VM model occurs, the Azure Linux Agent downloads and upgrades the extension.

Automatic extension updates are either *minor* or *hotfix*. You can opt in or opt out of minor updates when you provision the extension. The following example shows how to automatically upgrade minor versions in an ARM template by using `"autoUpgradeMinorVersion": true,`:

JSON

```
    "publisher": "Microsoft.Azure.Extensions",
    "type": "CustomScript",
    "typeHandlerVersion": "2.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-linux/scripts/config-music.sh"
        ]
    },
```

To get the latest minor-release bug fixes, we highly recommend that you always select automatic update in your extension deployments. You can't opt out of hotfix updates that carry security or key bug fixes.

If you disable automatic updates or you need to upgrade a major version, use az vm extension set or Set-AzVMExtension and specify the target version.

How to identify extension updates

Identify if the extension is set with autoUpgradeMinorVersion on a VM

Azure CLI

You can see from the VM model if the extension was provisioned with `autoUpgradeMinorVersion`. To check, use az vm show and provide the resource group and VM name:

Azure CLI

```
az vm show --resource-group myResourceGroup --name myVM
```

The following example output shows that `autoUpgradeMinorVersion` is set to `true`:

JSON

```
  "resources": [
    {
      "autoUpgradeMinorVersion": true,
      "forceUpdateTag": null,
      "id":
"/subscriptions/guid/resourceGroups/myResourceGroup/providers/Microsoft.Compute/
virtualMachines/myVM/extensions/customScript",
```

Identify when an autoUpgradeMinorVersion event occurred

To see when an update to the extension occurred, review the agent logs on the VM at */var/log/waagent.log*.

In the following example, the VM had `Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9025` installed. A hotfix was available to `Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027`.

Bash

```
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Expected handler state:
enabled
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Decide which version to use
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Use version: 2.3.9027
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Current handler state is:
NotInstalled
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Download extension package
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Unpack extension package
INFO Event: name=Microsoft.OSTCExtensions.LinuxDiagnostic, op=Download,
message=Download succeeded
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Initialize extension
directory
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Update settings file:
0.settings
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9025] Disable extension.
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9025] Launch
command:diagnostic.py -disable
...
INFO Event: name=Microsoft.OSTCExtensions.LinuxDiagnostic, op=Disable,
message=Launch command succeeded: diagnostic.py -disable
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Update extension.
INFO [Microsoft.OSTCExtensions.LinuxDiagnostic-2.3.9027] Launch
command:diagnostic.py -update
2017/08/14 20:21:57 LinuxAzureDiagnostic started to handle.
```

Agent permissions

To perform its tasks, the agent needs to run as *root*.

Troubleshoot VM extensions

Each VM extension might have specific troubleshooting steps. For example, when you use the Custom Script extension, you can find script execution details locally on the VM where the extension was run.

The following troubleshooting actions apply to all VM extensions:

- Check the Azure Linux Agent log. Look at the activity when your extension was being provisioned in */var/log/waagent.log*.
- Check the extension logs for more details in */var/log/azure/<extensionName>*.
- Check troubleshooting sections in extension-specific documentation for error codes, known issues, and other extension-specific information.
- Look at the system logs. Check to see if other operations interfered with the extension, such as a long-running installation of another application that required exclusive access to the package manager.

Common reasons for extension failures

- Extensions have 20 minutes to run. (Exceptions are Custom Script and Chef, which have 90 minutes.) If your deployment exceeds this time, it's marked as a timeout. The cause of a timeout can be:
  - Low-resource VMs.
  - Other VM configurations or startup tasks that consume large amounts of resources while the extension is trying to provision.

- Minimum prerequisites aren't met. Some extensions have dependencies on VM versions, such as high-performance computing images. Extensions might have certain networking access requirements, such as communicating with Azure Storage or public services. Other examples might be access to package repositories, running out of disk space, or security restrictions.

- Package manager access is exclusive. In some cases, a long-running VM configuration and extension installation might conflict because they both need exclusive access to the package manager.

View extension status

Azure CLI

After a VM extension runs against a VM, use az vm get-instance-view to return extension status:

Azure CLI

```
az vm get-instance-view \
    --resource-group myResourceGroup \
    --name myVM \
    --query "instanceView.extensions"
```

The output is similar to the following example:

Bash

```
  {
    "name": "customScript",
    "statuses": [
      {
        "code": "ProvisioningState/failed/0",
        "displayStatus": "Provisioning failed",
        "level": "Error",
        "message": "Enable failed: failed to execute command: command terminated
with exit status=127\n[stdout]\n\n[stderr]\n/bin/sh: 1: ech: not found\n",
        "time": null
      }
    ],
    "substatuses": null,
    "type": "Microsoft.Azure.Extensions.CustomScript",
    "typeHandlerVersion": "2.1.6"
  }
```

You can also find extension execution status in the Azure portal. Select the VM, select **Extensions**, and then select the extension that you want.

Rerun a VM extension

There might be cases in which a VM extension must be rerun. To rerun an extension, remove it, and then rerun the extension with an execution method of your choice.

Azure CLI

To remove an extension, use az vm extension delete:

Azure CLI

```
az vm extension delete \
    --resource-group myResourceGroup \
    --vm-name myVM \
    --name customScript
```

You can also remove an extension in the Azure portal:

1. Select a VM.
2. Select **Extensions**.
3. Select the extension.
4. Select **Uninstall**.

Common VM extension reference

Expand table

Extension name	Description
Custom Script extension for Linux	Run scripts against an Azure VM.
VMAccess extension	Regain access to an Azure VM. You can also use it to manage users and credentials.

Related content

- For more information about VM extensions, see Azure virtual machine extensions and features.

Last updated on 11/20/2025

Azure Linux VM Agent overview

The Azure Linux VM Agent (waagent) manages Linux and FreeBSD provisioning, along with virtual machine (VM) interaction with the Azure fabric controller. In addition to the Linux agent providing provisioning functionality, Azure provides the option of using cloud-init for some Linux operating systems.

> **Important**
>
> FIPS 140-3 Enforced isn't supported on RHEL/Ubuntu with extensions by using `protectedSettings`.
>
> Support for FIPS 140-3 will be added September 2025.

The Linux agent provides the following functionality for Linux and FreeBSD Azure Virtual Machines deployments. For more information, see the Azure Linux VM Agent readme on GitHub.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Image provisioning

- Creates a user account.
- Configures SSH authentication types.
- Deploys Secure Shell (SSH) public keys and key pairs.
- Sets the host name.
- Publishes the host name to the platform Domain Name System (DNS).
- Reports the SSH host key fingerprint to the platform.
- Manages the resource disk.
- Formats and mounts the resource disk.
- Configures swap space.

Networking

- Manages routes to improve compatibility with platform DHCP servers.
- Ensures the stability of the network interface name.

Kernel

- Configures virtual NUMA (disabled for kernel 2.6.37).
- Consumes Hyper-V entropy for */dev/random*.
- Configures SCSI timeouts for the root device, which can be remote.

Diagnostics

- Provides console redirection to the serial port.

System Center Virtual Machine Manager deployments

- Detects and bootstraps the Virtual Machine Manager agent for Linux when it's running in a System Center Virtual Machine Manager 2012 R2 environment.

VM extension

- Injects components authored by Microsoft and partners into Linux VMs to enable software and configuration automation.

You can find a VM extension reference implementation on GitHub.

Communication

Information flow from the platform to the agent occurs through two channels:

- A boot-time attached DVD for VM deployments. This DVD includes an Open Virtualization Format (OVF)-compliant configuration file that contains all provisioning information other than the SSH key pairs.
- A TCP endpoint that exposes a REST API that's used to get deployment and topology configuration.

Requirements

Testing confirms that the following systems work with the Azure Linux VM Agent.

> **Note**
>
> This list might differ from the endorsed Linux distributions on Azure.

Expand table

Publisher	Distribution	x64	ARM64
Alma Linux Community	Alma Linux	8.x+, 9.x+	8.x+, 9.x+
Credativ	Debian	10+	11.x+
Kinvolk	Flatcar Linux	3374.2.x+	3374.2.x+
Microsoft	Azure Linux	2.x	2.x
openSUSE Project	openSUSE	12.3+	*Not supported*
Oracle	Oracle Linux	6.4+, 7.x+, 8.x+	*Not supported*
Red Hat	Red Hat Enterprise Linux	6.7+, 7.x+, 8.x+, 9.x+, 10.x+	8.6+, 9.0+, 10.x+
CIQ	Rocky Linux	9.x+	9.x+
SUSE	SLES	12.x+, 15.x+	15.x SP4+
Canonical	Ubuntu (LTS releases)	18.04+, 20.04+, 22.04+, 24.04+	20.04+, 22.04+, 24.04+

Other supported systems:

- The agent works on more systems than the ones listed in the documentation. However, we don't test or provide support for distros that aren't on the endorsed list. In particular, FreeBSD isn't endorsed. You can try FreeBSD 8. If you run into problems, you can open an issue in our GitHub repository and we might be able to help.

The Linux agent depends on these system packages to function properly:

- Python 2.6+
- OpenSSL 1.0+
- OpenSSH 5.3+
- File system utilities: sfdisk, fdisk, mkfs, parted
- Password tools: chpasswd, sudo
- Text processing tools: sed, grep
- Network tools: ip-route
- Kernel support for mounting UDF file systems

Ensure that your VM has access to IP address 168.63.129.16. For more information, see What is IP address 168.63.129.16?.

Installation

The supported method of installing and upgrading the Azure Linux VM Agent uses an RPM or a DEB package from your distribution's package repository. All the endorsed distribution providers integrate the Azure Linux VM Agent package into their images and repositories. Some Linux distributions might disable the Azure Linux VM Agent **Auto Update** feature. Some of the repositories might also contain older versions. Older versions might have issues with modern extensions, so we recommend that you install the latest stable version.

You need to make sure that the Azure Linux VM Agent is updating properly. We recommend that you have the option `AutoUpdate.Enabled=Y` in the `/etc/waagent.conf` file because commenting out that option results in defaults too. Having `AutoUpdate.Enabled=N` doesn't allow the Azure Linux VM Agent to update properly.

For advanced installation options, such as installing from a source or to custom locations or prefixes, see Microsoft Azure Linux VM Agent. Other than these scenarios, we don't support or recommend upgrading or reinstalling the Azure Linux VM Agent from source.

Command-line options

Flags

- `verbose`: Increases verbosity of the specified command.
- `force`: Skips interactive confirmation for some commands.

Commands

- `help`: Lists the supported commands and flags.

- `deprovision`: Attempts to clean the system and make it suitable for reprovisioning. The operation deletes:
  - All SSH host keys, if `Provisioning.RegenerateSshHostKeyPair` is `y` in the configuration file.
  - `Nameserver` configuration in */etc/resolv.conf*.
  - The root password from */etc/shadow*, if `Provisioning.DeleteRootPassword` is `y` in the configuration file.
  - Cached DHCP client leases.

  The client resets the host name to `localhost.localdomain`.

> ⚠ **Warning**
>
> Deprovisioning doesn't guarantee that the image is cleared of all sensitive information and suitable for redistribution.

- `deprovision+user`: Performs everything in `deprovision` and deletes the last provisioned user account (obtained from */var/lib/waagent*) and associated data. Use this parameter when you deprovision an image that was previously provisioned on Azure so that it can be captured and reused.
- `version`: Displays the version of waagent.
- `serialconsole`: Configures GRUB to mark ttyS0, the first serial port, as the boot console. This option ensures that kernel boot logs are sent to the serial port and made available for debugging.
- `daemon`: Runs waagent as a daemon to manage interaction with the platform. This argument is specified to waagent in the waagent *init* script.
- `start`: Runs waagent as a background process.

Configuration

The */etc/waagent.conf* configuration file controls the actions of waagent. Here's an example of a configuration file:

config

```
Provisioning.Enabled=y
Provisioning.DeleteRootPassword=n
Provisioning.RegenerateSshHostKeyPair=y
Provisioning.SshHostKeyPairType=rsa
Provisioning.MonitorHostName=y
Provisioning.DecodeCustomData=n
Provisioning.ExecuteCustomData=n
Provisioning.AllowResetSysUser=n
Provisioning.PasswordCryptId=6
Provisioning.PasswordCryptSaltLength=10
ResourceDisk.Format=y
ResourceDisk.Filesystem=ext4
ResourceDisk.MountPoint=/mnt/resource
ResourceDisk.MountOptions=None
ResourceDisk.EnableSwap=n
ResourceDisk.SwapSizeMB=0
LBProbeResponder=y
Logs.Verbose=n
```

```
OS.RootDeviceScsiTimeout=300
OS.OpensslPath=None
HttpProxy.Host=None
HttpProxy.Port=None
AutoUpdate.Enabled=y
```

Configuration options are of three types: `Boolean`, `String`, or `Integer`. You can specify the `Boolean` configuration options as `y` or `n`. The special keyword `None` might be used for some string type configuration entries.

Provisioning.Enabled

txt

```
Type: Boolean
Default: y
```

This option allows the user to enable or disable the provisioning functionality in the agent. Valid values are `y` and `n`. If provisioning is disabled, SSH host and user keys in the image are preserved and configuration in the Azure provisioning API is ignored.

> ⓘ **Note**
>
> The `Provisioning.Enabled` parameter defaults to `n` on Ubuntu Cloud Images that use cloud-init for provisioning.

Provisioning.DeleteRootPassword

txt

```
Type: Boolean
Default: n
```

If the value is `y`, the agent erases the root password in the */etc/shadow* file during the provisioning process.

Provisioning.RegenerateSshHostKeyPair

txt

```
Type: Boolean
Default: y
```

If the value is `y`, the agent deletes all SSH host key pairs from */etc/ssh/* during the provisioning process, including ECDSA, DSA, and RSA. The agent generates a single fresh key pair.

Configure the encryption type for the fresh key pair by using the `Provisioning.SshHostKeyPairType` entry. Some distributions re-create SSH key pairs for any missing encryption types when the SSH daemon is restarted. An example is after a reboot.

Provisioning.SshHostKeyPairType

```txt
Type: String
Default: rsa
```

You can set this option to an encryption algorithm type that the SSH daemon supports on the VM. The typically supported values are `rsa`, `dsa`, and `ecdsa`. The *putty.exe* file on Windows doesn't support `ecdsa`. If you intend to use *putty.exe* on Windows to connect to a Linux deployment, use `rsa` or `dsa`.

Provisioning.MonitorHostName

```txt
Type: Boolean
Default: y
```

If the value is `y`, waagent monitors the Linux VM for a host name change, as returned by the `hostname` command. Waagent then automatically updates the networking configuration in the image to reflect the change. To push the name change to the DNS servers, networking restarts on the VM. This restart results in brief loss of internet connectivity.

Provisioning.DecodeCustomData

```txt
Type: Boolean
Default: n
```

If the value is `y`, waagent decodes `CustomData` from Base64.

Provisioning.ExecuteCustomData

txt

```
Type: Boolean
Default: n
```

If the value is `y`, waagent runs `CustomData` after provisioning.

Provisioning.AllowResetSysUser

txt

```
Type: Boolean
Default: n
```

This option allows the password for the system user to be reset. It's disabled by default.

Provisioning.PasswordCryptId

txt

```
Type: String
Default: 6
```

This option specifies the algorithm that `crypt` uses when it's generating a password hash. Valid values are:

- `1`: MD5
- `2a`: Blowfish
- `5`: SHA-256
- `6`: SHA-512

Provisioning.PasswordCryptSaltLength

txt

```
Type: String
Default: 10
```

This option specifies the length of random salt used in generating a password hash.

ResourceDisk.Format

```txt
Type: Boolean
Default: y
```

If the value is `y`, waagent formats and mounts the resource disk that the platform provides, unless the file system type that the user requested in `ResourceDisk.Filesystem` is `ntfs`. The agent makes a single Linux partition (ID 83) available on the disk. This partition isn't formatted if it can be successfully mounted.

ResourceDisk.Filesystem

```txt
Type: String
Default: ext4
```

This option specifies the file system type for the resource disk. Supported values vary by Linux distribution. If the string is `X`, then `mkfs.X` should be present on the Linux image.

ResourceDisk.MountPoint

```txt
Type: String
Default: /mnt/resource
```

This option specifies the path at which the resource disk is mounted. The resource disk is a *temporary* disk and might be emptied when the VM is deprovisioned.

ResourceDisk.MountOptions

```txt
Type: String
Default: None
```

This option specifies disk mount options to be passed to the `mount -o` command. The value is a comma-separated list of values, for example, `nodev,nosuid`. For more information, see the `mount(8)` manual page.

ResourceDisk.EnableSwap

txt

```
Type: Boolean
Default: n
```

If you set this option, the agent creates a swap file (*/swapfile*) on the resource disk and adds it to the system swap space.

ResourceDisk.SwapSizeMB

txt

```
Type: Integer
Default: 0
```

This option specifies the size of the swap file in megabytes.

Logs.Verbose

txt

```
Type: Boolean
Default: n
```

If you set this option, log verbosity is boosted. Waagent logs to */var/log/waagent.log* and uses the system `logrotate` functionality to rotate logs.

OS.EnableRDMA

txt

```
Type: Boolean
Default: n
```

If you set this option, the agent attempts to install and then load an RDMA kernel driver that matches the version of the firmware on the underlying hardware.

OS.RootDeviceScsiTimeout

txt

```
Type: Integer
Default: 300
```

This option configures the SCSI timeout in seconds on the operating system disk and data drives. If it isn't set, the system defaults are used.

OS.OpensslPath

```txt
Type: String
Default: None
```

You can use this option to specify an alternate path for the *openssl* binary to use for cryptographic operations.

HttpProxy.Host, HttpProxy.Port

```txt
Type: String
Default: None
```

If you set this option, the agent uses this proxy server to access the internet.

AutoUpdate.Enabled

```txt
Type: Boolean
Default: y
```

Enable or disable autoupdate for goal state processing. The default value is `y`.

Automatic log collection in the Azure Linux Guest Agent

As of version 2.7+, the Azure Linux Guest Agent has a feature to automatically collect some logs and upload them. This feature currently requires `systemd`. It uses a new `systemd` slice called `azure-walinuxagent-logcollector.slice` to manage resources while it performs the collection.

The purpose is to facilitate offline analysis. The agent produces a *.zip* file of some diagnostics logs before uploading them to the VM's host. Engineering teams and support professionals

can retrieve the file to investigate issues for the VM owner. For technical information on the files that the Azure Linux Guest Agent collects, see the *azurelinuxagent/common/logcollector_manifests.py* file in the agent's GitHub repository.

You can disable this option by editing */etc/waagent.conf*. Update `Logs.Collect` to `n`.

Ubuntu Cloud Images

Ubuntu Cloud Images use cloud-init to do many configuration tasks that the Azure Linux VM Agent would otherwise manage. The following differences apply:

- `Provisioning.Enabled` defaults to `n` on Ubuntu Cloud Images that use cloud-init to perform provisioning tasks.
- The following configuration parameters have no effect on Ubuntu Cloud Images that use cloud-init to manage the resource disk and swap space:
  - `ResourceDisk.Format`
  - `ResourceDisk.Filesystem`
  - `ResourceDisk.MountPoint`
  - `ResourceDisk.EnableSwap`
  - `ResourceDisk.SwapSizeMB`

To configure the resource disk mount point and swap space on Ubuntu Cloud Images during provisioning, see the following resources:

- Ubuntu wiki: AzureSwapPartitions
- Deploy applications to a Windows virtual machine in Azure with the Custom Script extension

---

Last updated on 11/20/2025

FIPS 140-3 support for Azure Linux VM extensions and guest agent

> **Note**
>
> This feature is currently in public preview. Production workloads are supported.

Linux virtual machine (VM) extensions currently comply with FIPS 140-2, but updates to the platform were required to add support for FIPS 140-3. These changes are currently being enabled across the commercial cloud and Azure Government clouds. Linux VM extensions that use protected settings are also being updated to be able to use a FIPS 140-3-compliant encryption algorithm. This article helps enable support for FIPS 140-3 on Linux VMs where compliance with FIPS 140-3 is enforced. This change isn't needed on Windows images because of the way that FIPS compliance is implemented.

For more information, see What are the Federal Information Processing Standards (FIPS)?.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Confirmed supported extensions

Expand table

Extension	Supported clouds Commercial, Government, Air-Gap
MICROSOFT.AKS.COMPUTE.AKS.LINUX.AKSNODE	Commercial, Government
MICROSOFT.AKS.COMPUTE.AKS.LINUX.BILLING	Commercial, Government

Extension	Supported clouds Commercial, Government, Air-Gap
MICROSOFT.AZURE.AZUREDEFENDERFORSERVERS.MDE.LINUX	Commercial, Government
MICROSOFT.AZURE.EXTENSIONS.CUSTOMSCRIPT	Commercial, Government
MICROSOFT.AZURE.KEYVAULT.KEYVAULTFORLINUX	Commercial, Government
MICROSOFT.AZURE.MONITOR.AZUREMONITORLINUXAGENT	Commercial, Government
MICROSOFT.AZURE.MONITORING.DEPENDENCYAGENT.DEPENDENCYAGENTLINUX	Commercial
MICROSOFT.AZURE.NETWORKWATCHER.NETWORKWATCHERAGENTLINUX	Commercial, Government
MICROSOFT.AZURE.RECOVERYSERVICES.VMSNAPSHOT	Commercial
MICROSOFT.AZURE.SECURITY.MONITORING.AZURESECURITYLINUXAGENT	Commercial
MICROSOFT.CPLAT.CORE.LINUXPATCHEXTENSION	Commercial, Government
MICROSOFT.CPLAT.CORE.RUNCOMMANDLINUX	Commercial, Government
MICROSOFT.CPLAT.CORE.VMAPPLICATIONMANAGERLINUX	Commercial, Government
MICROSOFT.CPLAT.PROXYAGENT.PROXYAGENTLINUX	Commercial, Government
MICROSOFT.CPLAT.PROXYAGENT.PROXYAGENTLINUXARM64	Commercial, Government
MICROSOFT.GUESTCONFIGURATION.CONFIGURATIONFORLINUX	Commercial
MICROSOFT.MANAGEDSERVICES.APPLICATIONHEALTHLINUX	Commercial, Government
MICROSOFT.OSTCEXTENSIONS.VMACCESSFORLINUX	Commercial, Government

Prerequisites

You must meet the following four requirements to use a FIPS 140-3-compliant VM in Azure:

- The VM must be in a region where FIPS 140-3 platform changes are rolled out.
- Your Azure subscription must be opted in to FIPS 140-3 enablement.
- Each VM must be enrolled in FIPS 140-3 enablement in Azure Resource Manager.
- Inside the guest operating system (OS), the OS must be configured for FIPS 140 mode. The OS must run a version of the Azure guest agent (waagent), which is also compliant with FIPS 140-3.

Afterward, validate to ensure the functionality of the VM extensions.

Implement prerequisites

1. Subscription enablement/opt-in

Because not all extensions are onboarded by using FIPS 140-3 encryption yet, we require the subscription to opt in to the feature `_Microsoft.Compute/OptInToFips1403Compliance_`.

Azure CLI

```
az feature register --namespace Microsoft.Compute --name OptInToFips1403Compliance
```

Verify with the following command:

```
az feature list | jq '.[] |
select(.name=="Microsoft.Compute/OptInToFips1403Compliance")'
```

JSON

```
{
  "id": "/subscriptions/<SUBSCRIPTION
ID>/providers/Microsoft.Features/providers/Microsoft.Compute/features/OptInToFips14
03Compliance",
  "name": "Microsoft.Compute/OptInToFips1403Compliance",
  "properties": {
    "state": "Registered"
  },
```

```
  "type": "Microsoft.Features/providers/features"
}
```

---

2. Per-VM opt-in

There are different methods available for opting in to each VM. You can make the changes at deployment for a new VM. You can also alter an existing VM to add the FIPS 140-3 enablement on the Azure platform.

> **⚠ Warning**
>
> We don't recommend using the following opt-in methods on Red Hat Enterprise Linux (RHEL) 9.5 and 9.6 by using version 2.7.0.6 of `WALinuxAgent` on production systems. An issue can surface after rebooting, after the FIPS enablement and subsequent reboot. On these VMs, the `waagent.service` enters an internal loop and never comes to a `Ready` state. Because of this error, no extensions can function. For testing, you can try the RHEL 9 workaround.

Deploy a new VM

To deploy a new VM with FIPS 140-3 enablement turned on immediately, use an Azure Resource Manager template (ARM template) or the Azure CLI. Add the `enableFips1403Encryption` property to the `additionalCapabilities` section of the `virtualMachines` object definition.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachines",
  "apiVersion": "2024-11-01",
  "name": "[parameters('vmName')]",
  "location": "[parameters('location')]",
  "properties": {
      "additionalCapabilities": {
      "enableFips1403Encryption": true
    }
  }
}
```

Modify an existing VM

az cli commands

> **Note**
>
> For the Government cloud, use `https://management.usgovcloudapi.net` instead of `https://management.azure.com`.

While updates to the SDK/CLI are still in progress, you can continue to use `az cli` to add the property.

```
az rest \
--method put \
--url 'https://management.azure.com/subscriptions/<SUBSCRIPTION
ID>/resourceGroups/<RESOURCE GROUP>/providers/Microsoft.Compute/virtualMachines/<VM
NAME>/?api-version=2024-11-01' \
--body '{"location": "<LOCATION>", "properties": {"additionalCapabilities":
{"enableFips1403Encryption": true}}}'
```

Running the `put` command outputs the resulting JSON for the modified VM. For later verification, you can run this `get` command against the VM object, which outputs the full JSON again.

```
az rest \
--method get \
--url 'https://management.azure.com/subscriptions/<SUBSCRIPTION
ID>/resourceGroups/<RESOURCE GROUP>/providers/Microsoft.Compute/virtualMachines/<VM
NAME>/?api-version=2024-11-01'
```

The command output should include:

JSON

```
{
 "enableFips1403Encryption": true
}
```

To more easily find the property in the output, you can add `jq` to parse out the specific section needed. This block is the new command:

```
az rest \
--method get \
--url 'https://management.azure.com/subscriptions/<SUBSCRIPTION
ID>/resourceGroups/<RESOURCE GROUP>/providers/Microsoft.Compute/virtualMachines/<VM
NAME>/?api-version=2024-11-01' \
| jq .properties.additionalCapabilities
```

For comparison, one possible outcome when you try to enable FIPS 140-3 on a VM when the VM isn't in an enabled region, the `put` command can output the following code, which indicates that the action isn't possible in the region.

JSON

```
({
  "error": {
    "code": "BadRequest",
    "message": "Creation of VMs using a Fips 140-3 compliant encryption for
extension settings isn't supported in this region."
  }
})
```

---

3. In-guest considerations

Important changes must be made to the Linux OS environment to enable and support FIPS 140-3 compliance.

Configure the OS for FIPS enablement

The following distributions support FIPS 140-3 and provide instructions for enabling:

- Ubuntu 22.04 LTS and newer: Use an Ubuntu pro client or pro image.
- RHEL 9: Use the steps to enable FIPS on RHEL.

Older versions of these operating systems operate at the FIPS 140-2 level and don't require any of these special considerations.

Linux guest agent

Minimum Goal State Agent version: v2.14.0.1. To be sure that the goal state is updating, the `AutoUpdate.Enabled` flag should be `y` or commented out entirely so that the default behavior is used.

`/etc/waagent.conf`:

```
AutoUpdate.Enabled=y
```

RHEL 9 workaround

This workaround is intended for testing purposes only and doesn't support all VM deployment scenarios. After you enable FIPS on a running VM, run the following commands to proceed:

```
systemctl stop waagent

# apply the patch
sed -i -E '/(.+)(self._initialize_telemetry\(\))/s//\1# \2/'
/usr/lib/python3.9/site-packages/azurelinuxagent/daemon/main.py
```

To verify that the previous change was applied successfully, use the following command:

```
grep self\._initialize_telemetry /usr/lib/python3.9/site-
packages/azurelinuxagent/daemon/main.py
```

The output should be exactly this text:

```
        # self._initialize_telemetry()
```

After verification, restart the agent:

```
systemctl start waagent
```

Validation

To validate proper functionality of the VM extensions:

- Check that the agent status is `Ready`.

- Test an extension by using the protected settings of the VM extensions.
  - By using the `Reset Password` function of the Azure portal or `az cli`, reset a password or create a new temporary user.
  - Run a custom script.

If these tests fail, force the Azure platform to generate a new personal information exchange (PFX) package.

Reset password

Use either the Azure portal or an `az cli` command, such as this example, to set a user's password or create a temporary user. Check the execution state for success or failure.

Bash

```
az vm user update \
  --resource-group <YourResourceGroup> \
  --name <YourVMName> \
  --username <NewUsername> \
  --password <NewPassword>
```

Run a custom script

Use the Custom Script extension documentation to send a basic script, such as `cat /etc/os-release`, to test extension functionality.

Fix a validation failure

If the validations fail to execute, force the Azure platform to generate a new PFX package. There are two methods to force this regeneration:

- Reallocate the VM.
- Apply an Azure Key Vault certificate.

Deallocate/Reallocate the VM

You can use the Azure CLI, the Azure portal, or any other method to deallocate the VM. Wait for the deallocation to occur, and then start the VM.

Add a Key Vault certificate

Create the Key Vault certificate, add it to the modified ARM template, and deploy.

For more information, see Get started with Key Vault certificates.

The following example shows the `properties` section of the VM model:

JSON

```
      "secrets": [
        {
          "sourceVault": {
            "id": "/subscriptions/<subId>/resourceGroups/<resource
group>/providers/Microsoft.KeyVault/vaults/<keyvault name>"
          },
          "vaultCertificates": [
            {
              "certificateUrl": "https://<keyvault
name>.vault.azure.net/secrets/rsa/5cc588f8f3404268b2ca4d05e544e7fb"
            }
          ]
        }
      ],
```

Last updated on 03/25/2026

Virtual machine extensions and features for Windows

Summarize this article for me

Azure virtual machine (VM) extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs. For example, if a virtual machine requires software installation, antivirus protection, or the ability to run a script inside of the VM, you can use a VM extension.

You can run Azure VM extensions by using the Azure CLI, PowerShell, Azure Resource Manager (ARM) templates, and the Azure portal. You can bundle extensions with a new VM deployment or run them against any existing system.

This article provides an overview of Azure VM extensions, including prerequisites and guidance on how to detect, manage, and remove extensions. This article provides generalized information because many VM extensions are available. Each extension has a potentially unique configuration and its own documentation.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Use cases and samples

Each Azure VM extension has a specific use case. Here are some examples:

- Configure monitoring of a VM by using the Azure Monitor agent and VM insights.
- Configure an Azure VM by using Chef.
- Configure monitoring of your Azure infrastructure by using the Datadog extension.

In addition to process-specific extensions, a Custom Script Extension is available for both Windows and Linux virtual machines. The Custom Script Extension for Windows allows any PowerShell script to run on a VM. Custom scripts are useful for designing Azure deployments that require configuration beyond what native Azure tooling can provide.

Prerequisites

Review the following prerequisites for working with Azure VM extensions.

Azure VM Agent

To handle extensions on the VM, you need the Azure Virtual Machine Agent for Windows installed. This agent is also referred to as the Azure VM Agent or the Windows Guest Agent. As you prepare to install extensions, keep in mind that some extensions have individual prerequisites, such as access to resources or dependencies.

The Azure VM Agent manages interactions between an Azure VM and the Azure fabric controller. The agent is responsible for many functional aspects of deploying and managing Azure VMs, including running VM extensions.

The Azure VM Agent is preinstalled on Azure Marketplace images. The agent can also be installed manually on supported operating systems.

The agent runs on multiple operating systems. However, the extensions framework has a limit for the operating systems that extensions use. Some extensions aren't supported across all operating systems and might emit error code 51 ("Unsupported OS"). Check the individual extension documentation for supportability.

Network access

Extension packages are downloaded from the Azure Storage extension repository. Extension status uploads are posted to Azure Storage.

If you use a supported version of the Azure VM Agent, you don't need to allow access to Azure Storage in the VM region. You can use the VM Agent to redirect the communication to the Azure fabric controller for agent communications (via the `HostGAPlugin` feature through the privileged channel on private IP address 168.63.129.16). If you're on an unsupported version of the VM Agent, you need to allow outbound access to Azure Storage in that region from the VM.

> **ⓘ Important**
>
> If you block access to IP address 168.63.129.16 by using the guest firewall or via a proxy, extensions fail. Failure occurs even if you use a supported version of the VM Agent or you configure outbound access. Ports 80 and 32526 are required.

Agents can only be used to download extension packages and report status. For example, if an extension installation needs to download a script from GitHub (Custom Script Extension) or requires access to Azure Storage (Azure Backup), then you need to open other firewall or network security group (NSG) ports. Different extensions have different requirements because they're applications in their own right. For extensions that require access to Azure Storage or Microsoft Entra ID, you can allow access by using Azure NSG service tags.

The Azure VM Agent doesn't provide proxy server support to enable redirection of agent traffic requests. The VM Agent relies on your custom proxy (if you have one) to access resources on the internet or on the host through IP address 168.63.129.16.

Discover VM extensions

Many VM extensions are available for use with Azure VMs. To see a complete list, use the Get-AzVMExtensionImage PowerShell cmdlet.

The following command lists all available VM extensions in the West US region location:

PowerShell

```
Get-AzVmImagePublisher -Location "West US" |
Get-AzVMExtensionImageType |
Get-AzVMExtensionImage | Select Type, Version
```

This command provides output similar to the following example:

PowerShell

```
Type                Version
----                -------
AcronisBackup       1.0.33
AcronisBackup       1.0.51
AcronisBackupLinux  1.0.33
AlertLogicLM        1.3.0.1
AlertLogicLM        1.3.0.0
AlertLogicLM        1.4.0.1
```

Run VM extensions

Azure VM extensions run on existing VMs, which is useful when you need to make configuration changes or recover connectivity on an already deployed VM. VM extensions can also be bundled with ARM template deployments. By using extensions with ARM templates, you can deploy and configure Azure VMs without post-deployment intervention.

You can use the following methods to run an extension against an existing VM.

> **Note**
>
> Some of the following examples use `"<placeholder>"` parameter values in the commands. Before you run each command, make sure to replace any `"<placeholder>"` values with specific values for your configuration.

PowerShell

Several PowerShell commands exist for running individual extensions. To see a list, use the Get-Command command and filter on *Extension*:

PowerShell

```powershell
Get-Command Set-Az*Extension* -Module Az.Compute
```

This command provides output similar to the following example:

PowerShell

```
CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Set-AzVMAccessExtension                            4.5.0      Az.Compute
Cmdlet          Set-AzVMADDomainExtension                          4.5.0      Az.Compute
Cmdlet          Set-AzVMAEMExtension                               4.5.0      Az.Compute
Cmdlet          Set-AzVMBackupExtension                            4.5.0      Az.Compute
Cmdlet          Set-AzVMBginfoExtension                            4.5.0      Az.Compute
Cmdlet          Set-AzVMChefExtension                              4.5.0      Az.Compute
Cmdlet          Set-AzVMCustomScriptExtension                      4.5.0      Az.Compute
Cmdlet          Set-AzVMDiagnosticsExtension                       4.5.0      Az.Compute
Cmdlet          Set-AzVMDiskEncryptionExtension                    4.5.0      Az.Compute
Cmdlet          Set-AzVMDscExtension                               4.5.0      Az.Compute
Cmdlet          Set-AzVMExtension                                  4.5.0      Az.Compute
Cmdlet          Set-AzVMSqlServerExtension                         4.5.0      Az.Compute
Cmdlet          Set-AzVmssDiskEncryptionExtension                  4.5.0      Az.Compute
```

The following example uses the Custom Script Extension to download a script from a GitHub repository onto the target virtual machine and then run the script.

PowerShell

```powershell
Set-AzVMCustomScriptExtension -ResourceGroupName "<myResourceGroup>" `
    -VMName "<myVM>" -Name "<myCustomScript>" `
    -FileUri "https://raw.githubusercontent.com/neilpeterson/nepeters-azure-
```

```
templates/master/windows-custom-script-simple/support-scripts/Create-File.ps1" `
    -Run "Create-File.ps1" -Location "<myVMregion>"
```

The following example uses the VMAccess extension to reset the administrative password of a Windows VM to a temporary password. After you run this code, you should reset the password at first sign-in.

PowerShell

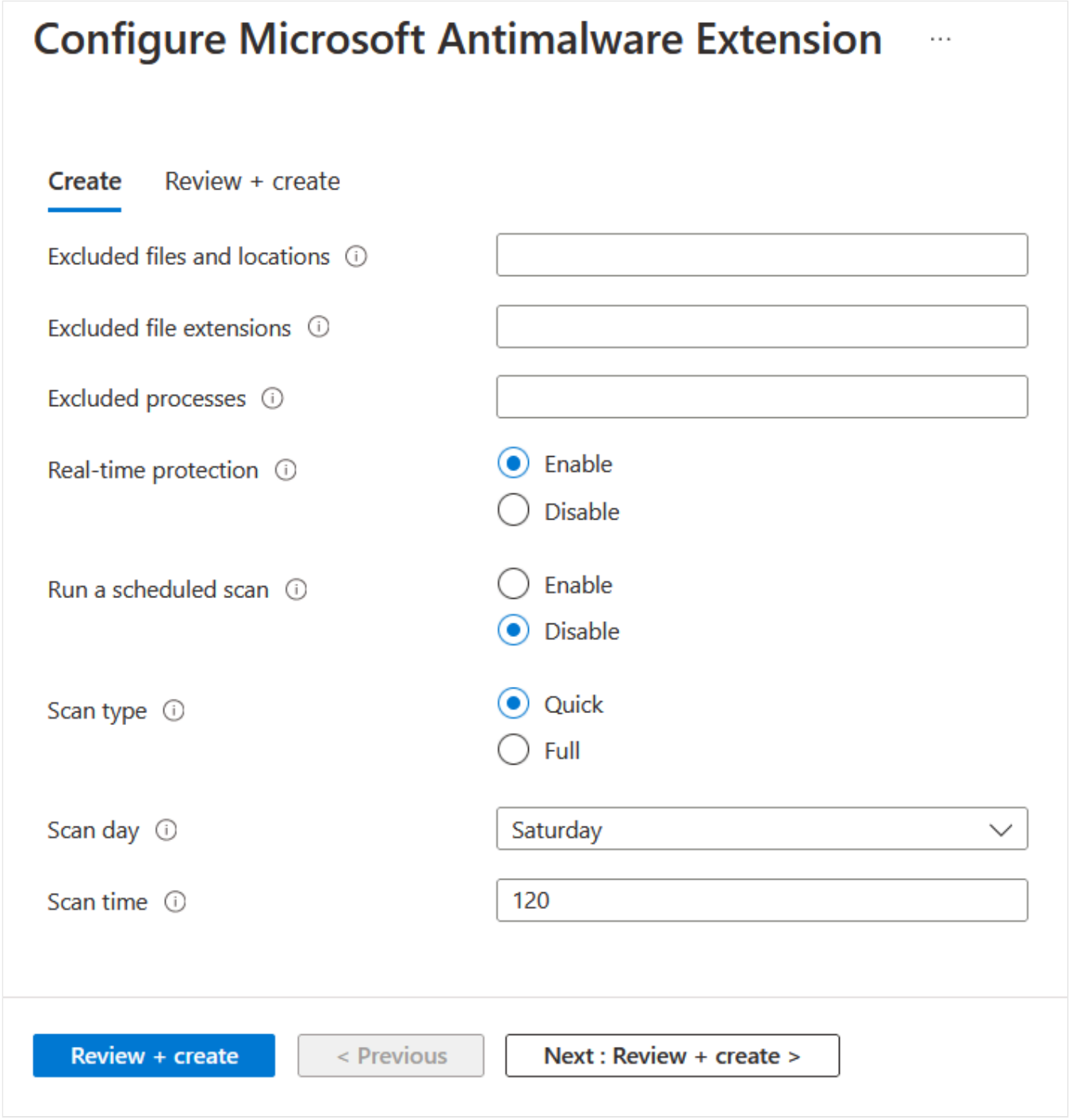
```
$cred=Get-Credential

Set-AzVMAccessExtension -ResourceGroupName "myResourceGroup" -VMName "myVM" -Name
"myVMAccess" `
    -Location "myVMregion" -UserName $cred.GetNetworkCredential().Username `
    -Password $cred.GetNetworkCredential().Password -typeHandlerVersion "2.0"
```

You can use the Set-AzVMExtension command to start any VM extension.

Azure portal

You can apply VM extensions to an existing VM through the Azure portal. Select the VM in the portal, select **Extensions + Applications**, and then select **+ Add**. Choose the extension that you want from the list of available extensions, and follow the instructions in the wizard.

The following example shows the installation of the Microsoft Antimalware extension from the Azure portal:



Azure Resource Manager templates

You can add VM extensions to an ARM template and run them with the deployment of the template. When you deploy an extension with a template, you can create fully configured Azure deployments.

The following JSON example is from an ARM template that deploys a set of load-balanced VMs and an Azure SQL database, and then installs a .NET Core application on each VM. The VM extension takes care of the software installation.

JSON

```
{
    "apiVersion": "2015-06-15",
    "type": "extensions",
    "name": "config-app",
    "location": "[resourceGroup().location]",
    "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/',
variables('vmName'),copyindex())]",
    "[variables('musicstoresqlName')]"
    ],
    "tags": {
    "displayName": "config-app"
    },
    "properties": {
    "publisher": "Microsoft.Compute",
    "type": "CustomScriptExtension",
    "typeHandlerVersion": "1.9",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1"
        ]
    },
    "protectedSettings": {
        "commandToExecute": "[concat('powershell -ExecutionPolicy Unrestricted -
File configure-music-app.ps1 -user ',parameters('adminUsername'),' -password
',parameters('adminPassword'),' -sqlserver
',variables('musicstoresqlName'),'.database.windows.net')]"
    }
    }
}
```

For more information on creating ARM templates, see Virtual machines in an ARM template.

Help secure VM extension data

When you run a VM extension, it might be necessary to include sensitive information such as credentials, storage account names, and access keys. Many VM extensions include a protected configuration that encrypts data and only decrypts it inside the target VM. Each extension has a specific protected configuration schema, and each schema is detailed in extension-specific documentation.

The following JSON example shows an instance of the Custom Script Extension for Windows. The command to run includes a set of credentials. In this example, the command to run isn't encrypted.

JSON

```
{
    "apiVersion": "2015-06-15",
    "type": "extensions",
    "name": "config-app",
    "location": "[resourceGroup().location]",
    "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/',
variables('vmName'),copyindex())]",
    "[variables('musicstoresqlName')]"
    ],
    "tags": {
    "displayName": "config-app"
    },
    "properties": {
    "publisher": "Microsoft.Compute",
    "type": "CustomScriptExtension",
    "typeHandlerVersion": "1.9",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1"
        ],
        "commandToExecute": "[concat('powershell -ExecutionPolicy Unrestricted -
File configure-music-app.ps1 -user ',parameters('adminUsername'),' -password
',parameters('adminPassword'),' -sqlserver
',variables('musicstoresqlName'),'.database.windows.net')]"
    }
    }
}
```

Moving the `commandToExecute` property to the `protected` configuration helps secure the execution string, as shown in the following example:

JSON

```
{
    "apiVersion": "2015-06-15",
    "type": "extensions",
    "name": "config-app",
    "location": "[resourceGroup().location]",
    "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/',
variables('vmName'),copyindex())]",
    "[variables('musicstoresqlName')]"
    ],
    "tags": {
    "displayName": "config-app"
    },
    "properties": {
    "publisher": "Microsoft.Compute",
    "type": "CustomScriptExtension",
```

```
    "typeHandlerVersion": "1.9",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1"
        ]
    },
    "protectedSettings": {
        "commandToExecute": "[concat('powershell -ExecutionPolicy Unrestricted -
File configure-music-app.ps1 -user ',parameters('adminUsername'),' -password
',parameters('adminPassword'),' -sqlserver
',variables('musicstoresqlName'),'.database.windows.net')]"
    }
    }
}
```

On an Azure infrastructure as a service (IaaS) VM that uses extensions, in the certificates console, you might see certificates that have the subject **Windows Azure CRP Certificate Generator**. On a classic RedDog Front End (RDFE) VM, these certificates have the subject name **Windows Azure Service Management for Extensions**.

These certificates secure the communication between the VM and its host during the transfer of protected settings (password and other credentials) that extensions use. The Azure fabric controller builds the certificates and passes them to the Azure VM Agent. If you stop and start the VM every day, the fabric controller might create a new certificate. The certificate is stored in the computer's personal certificate store. These certificates can be deleted. The Azure VM Agent re-creates certificates if needed.

How agents and extensions are updated

Agents and extensions share the same automatic update mechanism.

When an update is available and automatic updates are enabled, the update is installed on the VM only after an extension or other VM model changes. Changes can include:

- Data disks
- Extensions
- Extension Tags
- Boot diagnostics container
- Guest OS secrets
- VM size
- Network profile

Publishers make updates available to regions at various times. It's possible you can have VMs in different regions on different versions.

> ⓘ Note
>
> Some updates might require additional firewall rules. For more information, see Network access.

List extensions deployed to a VM

You can use the following command to list the extensions deployed to a VM:

PowerShell

```
$vm = Get-AzVM -ResourceGroupName "<myResourceGroup>" -VMName "<myVM>"
$vm.Extensions | select Publisher, VirtualMachineExtensionType, TypeHandlerVersion
```

This command produces output similar to the following example:

PowerShell

```
Publisher             VirtualMachineExtensionType          TypeHandlerVersion
---------             ---------------------------          ------------------
Microsoft.Compute     CustomScriptExtension                1.9
```

Agent updates

The Azure VM Agent contains only *extension-handling code*. The *Windows provisioning code* is separate. You can uninstall the Azure VM Agent. You can't disable the automatic update of the Azure VM Agent.

The extension-handling code is responsible for the following tasks:

- Communicate with the Azure fabric.
- Handle the VM extension operations, such as installations, reporting status, updating the individual extensions, and removing extensions. Updates contain security fixes, bug fixes, and enhancements to the extension-handling code.

To check what version you're running, see Detect the Azure VM Agent.

Extension updates

When an extension update is available and automatic updates are enabled, if a VM model changes, the Azure VM Agent downloads and upgrades the extension.

Automatic extension updates are either *minor* or *hotfix*. You can opt in or opt out of minor updates when you provision the extension. The following example shows how to automatically upgrade minor versions in an ARM template by using the `"autoUpgradeMinorVersion": true,` parameter:

JSON

```
    "properties": {
    "publisher": "Microsoft.Compute",
    "type": "CustomScriptExtension",
    "typeHandlerVersion": "1.9",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "fileUris": [
        "https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-
templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1"
        ]
    },
```

To get the latest minor-release bug fixes, we highly recommend that you always select automatic update in your extension deployments. You can't opt out of hotfix updates that carry security or key bug fixes.

If you disable automatic updates or you need to upgrade a major version, use the Set-AzVMExtension command and specify the target version.

Check Extension Version

> ⓘ Note
>
> **Model vs. Instance View**
>
> In Azure, the **model view** captures the configuration you defined for an extension (publisher, type, and requested version), while the **instance view** shows the live state and the actual handler version running on each VM or scale-set instance—use instance view as the source of truth when verifying what's really installed.

Check a single VM

Azure CLI

Bash

```
az vm get-instance-view -g <rg> -n <vm> \
  --query "extensions[].
```

```
{name:name,type:type,version:typeHandlerVersion,status:statuses[0].displayStatus}"
\
  -o table
```

Reads the VM's **instance view** to show the actually installed handler version and status.

PowerShell

PowerShell

```
Get-AzVM -ResourceGroupName <rg> -Name <vmName> -Status |
  Select-Object -ExpandProperty Extensions
```

Inspect `TypeHandlerVersion` and `Statuses` for each extension to confirm the running version and health.

Check scale sets (Uniform or Flexible)

Azure CLI

Bash

```
# Scale set summary (health)
az vmss get-instance-view -g <rg> -n <vmss>

# Per-instance extension versions
az vmss list-instances -g <rg> -n <vmss> --expand instanceView \
  --query "[].{instanceId:instanceId,
extVers:instanceView.extensions[].typeHandlerVersion}"

# Specific instance
az vmss vm get-instance-view -g <rg> -n <vmss> --instance-id <id>
```

Use instance view at the scale-set or per-instance level to validate the actual handler versions.

PowerShell

PowerShell

```
Get-AzVmssVM -ResourceGroupName <rg> -VMScaleSetName <vmss> -InstanceView
```

Returns per-instance runtime state, including extension versions and statuses.

How to identify extension updates

There are a few ways you can identify updates for an extension.

Identify if the extension is set with autoUpgradeMinorVersion on a VM

You can view the VM model to determine if the extension is provisioned with the `autoUpgradeMinorVersion` parameter. To check the VM model, use the Get-AzVm command and provide the resource group and VM name as follows:

powerShell

```
$vm = Get-AzVm -ResourceGroupName "myResourceGroup" -VMName "myVM"
$vm.Extensions
```

The following example output shows the `autoUpgradeMinorVersion` parameter is set to `true`:

PowerShell

```
ForceUpdateTag             :
Publisher                  : Microsoft.Compute
VirtualMachineExtensionType : CustomScriptExtension
TypeHandlerVersion         : 1.9
AutoUpgradeMinorVersion    : True
```

Identify when an autoUpgradeMinorVersion event occurs

To see when an update to the extension occurred, you can review the agent logs on the VM at *C:\WindowsAzure\Logs\WaAppAgent.log*.

The following example shows the VM with `Microsoft.Compute.CustomScriptExtension` version `1.8` installed, and a hotfix available for version `1.9`.

PowerShell

```
[INFO]  Getting plugin locations for plugin
'Microsoft.Compute.CustomScriptExtension'. Current Version: '1.8', Requested
Version: '1.9'
[INFO]  Auto-Upgrade mode. Highest public version for plugin
'Microsoft.Compute.CustomScriptExtension' with requested version: '1.9', is: '1.9'
```

Agent permissions

To perform its tasks, the Azure VM Agent needs to run as *Local System*.

Troubleshoot VM extensions

Each VM extension might have specific troubleshooting steps. For example, when you use the Custom Script Extension, you can find script execution details locally on the VM where the extension is run.

The following troubleshooting actions apply to all VM extensions:

- To check the Azure VM Agent Log, look at the activity when your extension was being provisioned in *C:\WindowsAzure\Logs\WaAppAgent.log*.
- Check the extension logs for more details in *C:\WindowsAzure\Logs\Plugins<extensionName>*.
- Check troubleshooting sections in extension-specific documentation for error codes, known issues, and other extension-specific information.
- Look at the system logs. Check for other operations that might have interfered with the extension, such as a long-running installation of another application that required exclusive access to the package manager.
- In a VM, if there's an existing extension with a failed provisioning state, any other new extension fails to install.

Common reasons for extension failures

Here are some common reasons an extension can fail:

- Extensions have 20 minutes to run. (Exceptions are Custom Script, Chef, and DSC, which have 90 minutes.) If your deployment exceeds this time, it's marked as a timeout. The cause of this issue can be low-resource VMs, or other VM configurations or startup tasks are consuming large amounts of resources while the extension is trying to provision.
- Minimum prerequisites aren't met. Some extensions have dependencies on VM SKUs, such as HPC images. Extensions might have certain networking access requirements, such as communicating with Azure Storage or public services. Other examples might be access to package repositories, running out of disk space, or security restrictions.
- Package manager access is exclusive. In some cases, a long-running VM configuration and extension installation might conflict because they both need exclusive access to the package manager.

View extension status

After a VM extension is run against a VM, use the Get-AzVM command to return extension status. The `Substatuses[0]` result shows that the extension provisioning succeeded, which

means it successfully deployed to the VM. If you see the `Substatuses[1]` result, then the execution of the extension inside the VM failed.

PowerShell

```powershell
Get-AzVM -ResourceGroupName "myResourceGroup" -VMName "myVM" -Status
```

The output is similar to the following example:

PowerShell

```
Extensions[0]           :
  Name                  : CustomScriptExtension
  Type                  : Microsoft.Compute.CustomScriptExtension
  TypeHandlerVersion    : 1.9
  Substatuses[0]        :
    Code                : ComponentStatus/StdOut/succeeded
    Level               : Info
    DisplayStatus       : Provisioning succeeded
    Message             : Windows PowerShell \nCopyright (C) Microsoft Corporation.
All rights reserved.\n
  Substatuses[1]        :
    Code                : ComponentStatus/StdErr/succeeded
    Level               : Info
    DisplayStatus       : Provisioning succeeded
    Message             : The argument 'cseTest%20Scriptparam1.ps1' to the -File
parameter does not exist. Provide the path to an existing '.ps1' file as an
argument to the

-File parameter.
  Statuses[0]           :
    Code                : ProvisioningState/failed/-196608
    Level               : Error
    DisplayStatus       : Provisioning failed
    Message             : Finished executing command
```

You can also find extension execution status in the Azure portal. Select the VM, select **Extensions**, and then select the desired extension.

Rerun a VM extension

In certain cases, you might need to rerun a VM extension. You can rerun an extension by removing the extension, and then rerunning the extension with an execution method of your choice. To remove an extension, use the Remove-AzVMExtension command as follows:

PowerShell

```
Remove-AzVMExtension -ResourceGroupName "myResourceGroup" -VMName "myVM" -Name
"myExtensionName"
```

You can also remove an extension in the Azure portal. Select a VM, select **Extensions**, and then select the desired extension. Select **Uninstall**.

Common VM extension reference

The following table provides some common references for VM extensions.

Expand table

Extension name	Description
Custom Script Extension for Windows	Run scripts against an Azure virtual machine.
Azure Diagnostics extension	Manage Azure Diagnostics.
VMAccess extension	Manage users and credentials.

Next steps

For more information about VM extensions, see Azure virtual machine extensions and features.

Last updated on 02/11/2026

Azure Windows VM Agent overview

The Microsoft Azure Windows VM Agent is a secure, lightweight process that manages virtual machine (VM) interaction with the Azure fabric controller. The Azure Windows VM Agent has a primary role in enabling and executing Azure virtual machine extensions. VM extensions enable post-deployment configuration of VMs, such as installing and configuring software. VM extensions also enable recovery features such as resetting the administrative password of a VM. Without the Azure Windows VM Agent, you can't run VM extensions.

This article describes how to install and detect the Azure Windows VM Agent.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

The Azure Windows VM Agent supports the x64 and ARM64 architecture for these Windows operating systems:

Expand table

OS Version	x64	ARM64
Windows 10	Supported	Supported
Windows 11	Supported	Supported
Windows Server 2016	Supported	Supported
Windows Server 2016 Core	Supported	Supported
Windows Server 2019	Supported	Supported
Windows Server 2019 Core	Supported	Supported
Windows Server 2022	Supported	Supported
Windows Server 2022 Core	Supported	Supported

OS Version	x64	ARM64
Windows Server 2025	Supported	Supported
Windows Server 2025 Core	Supported	Supported

Important

- The Azure Windows VM Agent needs at least Windows Server 2016 (64-bit) to run, with the .NET Framework 4.0. See Minimum version support for virtual machine agents in Azure.
- Ensure that your VM has access to IP address 168.63.129.16. For more information, see What is IP address 168.63.129.16?.
- If you only have one private IP on your VM's NIC, then we highly recommend having DHCP enabled in the guest VM. If you need a static private IP address, you should configure it through the Azure portal or PowerShell, and make sure the DHCP option inside the VM is enabled. Learn more about setting up a static IP address by using PowerShell. This will ensure that the IP configuration will always match what is configured on the VM in Azure.
- If you have multiple private IPs assigned to your VM's NIC, then ensure that you carefully follow the steps to assign the IP configurations correctly. After this, if the Guest Agent isn't able to communicate with 168.63.129.16, then please check that the primary IP in Windows matches the primary IP in your VM's NIC in Azure.
- Running the Azure Windows VM Agent in a nested virtualization VM might lead to unpredictable behavior, so it's not supported in that dev/test scenario.

Install the Azure Windows VM Agent

Azure Marketplace image

The Azure Windows VM Agent is installed by default on any Windows VM deployed from an Azure Marketplace image. When you deploy an Azure Marketplace image from the Azure portal, PowerShell, the Azure CLI, or an Azure Resource Manager template, the Azure Windows VM Agent is also installed.

The Azure Windows VM Agent package has two parts:

- Azure Windows Provisioning Agent (PA)
- Azure Windows Guest Agent (WinGA)

To boot a VM, you must have the PA installed on the VM. However, the WinGA does not need to be installed. At VM deploy time, you can select not to install the WinGA. The following example shows how to select the `provisionVmAgent` option with an Azure Resource Manager template:

JSON

```
{
    "resources": [{
        "name": ["parameters('virtualMachineName')"],
        "type": "Microsoft.Compute/virtualMachines",
        "apiVersion": "2016-04-30-preview",
        "location": ["parameters('location')"],
        "dependsOn": ["[concat('Microsoft.Network/networkInterfaces/',
parameters('networkInterfaceName'))]"],
        "properties": {
            "osProfile": {
                "computerName": ["parameters('virtualMachineName')"],
                "adminUsername": ["parameters('adminUsername')"],
                "adminPassword": ["parameters('adminPassword')"],
                "windowsConfiguration": {
                    "provisionVmAgent": "false"
                }
            }
        }
    }]
}
```

If you don't have the agents installed, you can't use some Azure services, such as Azure Backup or Azure Security. These services require an extension to be installed. If you deploy a VM without the WinGA, you can install the latest version of the agent later.

Manual installation

To manually install the Azure Windows VM Agent, download the installer and select the latest release. You can also search for a specific version in the GitHub page for Azure Windows VM Agent releases. The Azure Windows VM Agent is supported on Windows Server 2008 (64 bit) and later.

> **Note**

- If a VM was created from a custom or Marketplace image without `ProvisionVMAgent` enabled, then it is important to update the `AllowExtensionOperations` to `true` after you manually install the Azure Windows VM Agent.
- This is NOT required if the VM was created from a specialized disk. For example, if the VM was created directly from an existing disk.

PowerShell

```
$vm = Get-AzVM -Name <VM Name> -ResourceGroupName <Resource Group Name>
$vm.OSProfile.AllowExtensionOperations = $true
$vm | Update-AzVM
```

Detect the Azure Windows VM Agent

PowerShell

You can use the Azure Resource Manager PowerShell module to get information about Azure VMs. To see information about a VM, such as the provisioning state for the Azure Windows VM Agent, use Get-AzVM:

PowerShell

```
Get-AzVM
```

The following condensed example output shows the `ProvisionVMAgent` property nested inside `OSProfile`. You can use this property to determine if the VM agent has been deployed to the VM.

PowerShell

```
OSProfile                  :
  ComputerName             : myVM
  AdminUsername            : myUserName
  WindowsConfiguration     :
    ProvisionVMAgent       : True
    EnableAutomaticUpdates : True
```

Use the following script to return a concise list of VM names (running Windows OS) and the state of the Azure Windows VM Agent:

PowerShell

```
$vms = Get-AzVM

foreach ($vm in $vms) {
    $agent = $vm | Select -ExpandProperty OSProfile | Select -ExpandProperty
Windowsconfiguration | Select ProvisionVMAgent
    Write-Host $vm.Name $agent.ProvisionVMAgent
}
```

Use the following script to return a concise list of VM names (running Linux OS) and the state of the Azure Windows VM Agent:

PowerShell

```
$vms = Get-AzVM

foreach ($vm in $vms) {
    $agent = $vm | Select -ExpandProperty OSProfile | Select -ExpandProperty
Linuxconfiguration | Select ProvisionVMAgent
    Write-Host $vm.Name $agent.ProvisionVMAgent
}
```

Manual detection

When you're logged in to a Windows VM, you can use Task Manager to examine running processes. To check for the Azure Windows VM Agent, open Task Manager, select the **Details** tab, and look for a process named *WindowsAzureGuestAgent.exe*. The presence of this process indicates that the VM agent is installed.

Upgrade the Azure Windows VM Agent

The Azure Windows VM Agent for Windows is automatically upgraded on images deployed from Azure Marketplace. The new versions are stored in Azure Storage, so ensure that you don't have firewalls blocking access. As new VMs are deployed to Azure, they receive the latest VM agent at VM provision time. If you installed the agent manually or are deploying custom VM images, you need to manually update to include the new VM agent at image creation time.

Azure Windows Guest Agent automatic log collection

The Azure Windows Guest Agent has a feature to automatically collect some logs. The *CollectGuestLogs.exe* process controls this feature. It exists for both platform as a service (PaaS) cloud services and infrastructure as a service (IaaS) VMs. Its goal is to quickly and automatically collect diagnostics logs from a VM, so they can be used for offline analysis.

The collected logs are event logs, OS logs, Azure logs, and some registry keys. The agent produces a ZIP file that's transferred to the VM's host. Engineering teams and support professionals can then use this ZIP file to investigate issues on the request of the customer who owns the VM.

Azure Windows Guest Agent and OSProfile certificates

The Azure Windows VM Agent installs the certificates referenced in the `OSProfile` value of a VM or a virtual machine scale set. If you manually remove these certificates from the Microsoft Management Console (MMC) Certificates snap-in inside the guest VM, the Azure Windows Guest Agent will add them back. To permanently remove a certificate, you have to remove it from `OSProfile`, and then remove it from within the guest operating system.

For a virtual machine, use Remove-AzVMSecret to remove certificates from `OSProfile`.

For more information on certificates for virtual machine scale sets, see Azure Virtual Machine Scale Sets - How do I remove deprecated certificates?.

Next steps

For more information about VM extensions, see Azure virtual machine extensions and features.

Last updated on 12/08/2025

Azure Diagnostics Extension for Windows VMs

Applies to: ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Overview

The Azure Diagnostics VM extension enables you to collect monitoring data, such as performance counters and event logs, from your Windows VM. You can granularly specify what data you want to collect and where you want the data to go, such as an Azure Storage account or an Azure Event Hub. You can also use this data to build charts in the Azure portal or create metric alerts.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Operating system

The Azure Diagnostics Extension can be run against Windows 10 Client, Windows Server 2008 R2, 2012, 2012 R2, and 2016.

Internet connectivity

The Azure Diagnostics Extension requires that the target virtual machine is connected to the internet.

Extension schema

The Azure Diagnostics Extension schema and property values are described in this document.

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. The JSON schema detailed in the previous section can be used in an Azure Resource Manager template to run the Azure Diagnostics extension during an Azure Resource Manager template deployment. See Use monitoring and diagnostics with a Windows VM and Azure Resource Manager templates.

Azure CLI deployment

The Azure CLI can be used to deploy the Azure Diagnostics extension to an existing virtual machine. Replace the protected settings and settings properties with valid JSON from the extension schema above.

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name IaaSDiagnostics \
  --publisher Microsoft.Azure.Diagnostics \
  --version 1.9.0.0 --protected-settings protected-settings.json \
  --settings public-settings.json
```

PowerShell deployment

The `Set-AzVMDiagnosticsExtension` command can be used to add the Azure Diagnostics extension to an existing virtual machine. See also Use PowerShell to enable Azure Diagnostics in a virtual machine running Windows.

PowerShell

```
$vm_resourcegroup = "myvmresourcegroup"
$vm_name = "myvm"
$diagnosticsconfig_path = "DiagnosticsPubConfig.xml"

Set-AzVMDiagnosticsExtension -ResourceGroupName $vm_resourcegroup `
  -VMName $vm_name `
  -DiagnosticsConfigurationPath $diagnosticsconfig_path
```

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure CLI. To see the deployment state of extensions for a given VM, run the following command using the Azure CLI.

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

See this article for a more comprehensive troubleshooting guide for the Azure Diagnostics extension.

Error: "Profile operation failed"

To enable profiling, please follow Enable Profiler for web apps on an Azure virtual machine.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Next Steps

- Learn more about the Azure Diagnostics Extension
- Review the extension schema and versions

Last updated on 12/02/2025

Azure Backup for SQL Server running in Azure VM

Azure Backup, amongst other offerings, provides support for backing up workloads such as SQL Server running in Azure VMs. Since the SQL application is running within an Azure VM, the backup service needs permission to access the application and fetch the necessary details. To do that, Azure Backup installs the **AzureBackupWindowsWorkload** extension on the VM, in which the SQL Server is running, during the registration process triggered by the user.

> ⓘ Note
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

For the list of supported scenarios, refer to the supportability matrix supported by Azure Backup.

Network connectivity

Azure Backup supports NSG Tags, deploying a proxy server or listed IP ranges; for details on each of the methods, refer this article.

Extension schema

The extension schema and property values are the configuration values (runtime settings) that service is passing to CRP API. These config values are used during registration and upgrade. **AzureBackupWindowsWorkload** extension also uses this schema. The schema is pre-set; a new parameter can be added in the objectStr field

JSON

```
    "runtimeSettings": [{
    "handlerSettings": {
    "protectedSettingsCertThumbprint": "",
    "protectedSettings": {
    "objectStr": "",
    "logsBlobUri": "",
    "statusBlobUri": ""
    }
    },
    "publicSettings": {
    "locale": "en-us",
    "taskId": "1c0ae461-9d3b-418c-a505-bb31dfe2095d",
    "objectStr": "",
    "commandStartTimeUTCTicks": "636295005824665976",
    "vmType": "vmType"
    }
    }]
    }
```

The following JSON shows the schema for the WorkloadBackup extension.

JSON

```
{
  "type": "extensions",
  "name": "WorkloadBackup",
  "location":"<myLocation>",
  "properties": {
    "publisher": "Microsoft.RecoveryServices",
    "type": "AzureBackupWindowsWorkload",
    "typeHandlerVersion": "1.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "locale":"<location>",
      "taskId":"<TaskId used by Azure Backup service to communicate with extension>",

      "objectStr": "<The configuration passed by Azure Backup service to extension>",

      "commandStartTimeUTCTicks": "<Scheduled start time of registration or upgrade task>",
      "vmType": "<Type of VM where registration got triggered Eg. Compute or ClassicCompute>"
```

```
    },
    "protectedSettings": {
      "objectStr": "<The sensitive configuration passed by Azure Backup service to extension>",
      "logsBlobUri": "<blob uri where logs of command execution by extension are written to>",
      "statusBlobUri": "<blob uri where status of the command executed by extension is written>"
    }
  }
}
```

Property values

Expand table

Name	Value/example
locale	en-us
taskId	"1c0ae461-9d3b-418c-a505-bb31dfe2095d"
objectStr (publicSettings)	"eyJjb250YWluZXJQcm9wZXJ0aWVzIjp7IkNvbnRhaW5lcklEIjoiMzVjMjQxYTItOGRjNy00ZGE5LWI4NTMtMjdjYTJhNDZlM2ZkIiwiSWRNZ210Q2
commandStartTimeUTCTicks	"636967192566036845"
vmType	"microsoft.compute/virtualmachines"
objectStr (protectedSettings)	"eyJjb250YWluZXJQcm9wZXJ0aWVzIjp7IkNvbnRhaW5lcklEIjoiMzVjMjQxYTItOGRjNy00ZGE5LWI4NTMtMjdjYTJhNDZlM2ZkIiwiSWRNZ210Q2
logsBlobUri	https://seapod01coord1exsapk732.blob.core.windows.net/bcdrextensionlogs-111111111-1111-1111-1111-111111111111/vmubuntu1404lt
statusBlobUri	https://seapod01coord1exsapk732.blob.core.windows.net/bcdrextensionlogs-111111111-1111-1111-1111-111111111111/vmubuntu1404lt

Template deployment

We recommended adding AzureBackupWindowsWorkload extension to a virtual machine is by enabling SQL Server backup on the virtual machine. This can be achieved through the Resource Manager template designed for automating backup on a SQL Server VM.

PowerShell deployment

You need to 'register' the Azure VM that contains the SQL application with a Recovery services vault. During registration, AzureBackupWindowsWorkload extension gets installed on the VM. Use Register-AzRecoveryServicesBackupContainerPS cmdlet to register the VM. Replace `VMRG Name` and `VMName` with the target resource group name and VM name.

PowerShell

```
$myVM = Get-AzVM -ResourceGroupName <VMRG Name> -Name <VMName>
Register-AzRecoveryServicesBackupContainer -ResourceId $myVM.ID -BackupManagementType AzureWorkload -WorkloadType MSSQL -VaultId
$targetVault.ID -Force
```

The command will return a **backup container** of this resource and the status will be **registered**.

Next steps

- Learn More about Azure SQL Server VM backup troubleshooting guidelines
- Common questions about backing up SQL Server databases that run on Azure virtual machines (VMs) and that use the Azure Backup service.

Last updated on 11/10/2025

Azure Disk Encryption for Linux (Microsoft.Azure.Security.AzureDiskEncryptionForLinux)

Overview

Azure Disk Encryption leverages the dm-crypt subsystem in Linux to provide full disk encryption on select Azure Linux distributions. This solution is integrated with Azure Key Vault to manage disk encryption keys and secrets.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

For a full list of prerequisites, see Azure Disk Encryption for Linux VMs, specifically the following sections:

- Supported VMs and operating systems
- Additional VM requirements
- Networking requirements
- Encryption key storage requirements

Extension Schema

There are two versions of extension schema for Azure Disk Encryption (ADE):

- v1.1 - A newer recommended schema that does not use Microsoft Entra properties.
- v0.1 - An older schema that requires Microsoft Entra properties.

To select a target schema, the `typeHandlerVersion` property must be set equal to version of schema you want to use.

Schema v1.1: No Microsoft Entra ID (recommended)

The v1.1 schema is recommended and does not require Microsoft Entra properties.

> **Note**
>
> The `DiskFormatQuery` parameter is deprecated. Its functionality has been replaced by the EncryptFormatAll option instead, which is the recommended way to format data disks at time of encryption.

JSON

```
{
  "type": "extensions",
  "name": "[name]",
  "apiVersion": "2019-07-01",
  "location": "[location]",
  "properties": {
        "publisher": "Microsoft.Azure.Security",
        "type": "AzureDiskEncryptionForLinux",
        "typeHandlerVersion": "1.1",
        "autoUpgradeMinorVersion": true,
        "settings": {
          "DiskFormatQuery": "[diskFormatQuery]",
          "EncryptionOperation": "[encryptionOperation]",
          "KeyEncryptionAlgorithm": "[keyEncryptionAlgorithm]",
          "KeyVaultURL": "[keyVaultURL]",
          "KeyVaultResourceId": "[KeyVaultResourceId]",
          "KeyEncryptionKeyURL": "[keyEncryptionKeyURL]",
          "KekVaultResourceId": "[KekVaultResourceId",
          "SequenceVersion": "sequenceVersion]",
          "VolumeType": "[volumeType]"
        }
  }
}
```

Schema v0.1: with Microsoft Entra ID

The 0.1 schema requires `AADClientID` and either `AADClientSecret` or `AADClientCertificate`.

Using `AADClientSecret`:

JSON

```
{
  "type": "extensions",
  "name": "[name]",
  "apiVersion": "2019-07-01",
  "location": "[location]",
  "properties": {
```

```
    "protectedSettings": {
      "AADClientSecret": "[aadClientSecret]",
      "Passphrase": "[passphrase]"
    },
    "publisher": "Microsoft.Azure.Security",
    "type": "AzureDiskEncryptionForLinux",
    "typeHandlerVersion": "0.1",
    "settings": {
      "AADClientID": "[aadClientID]",
      "DiskFormatQuery": "[diskFormatQuery]",
      "EncryptionOperation": "[encryptionOperation]",
      "KeyEncryptionAlgorithm": "[keyEncryptionAlgorithm]",
      "KeyEncryptionKeyURL": "[keyEncryptionKeyURL]",
      "KeyVaultURL": "[keyVaultURL]",
      "SequenceVersion": "sequenceVersion]",
      "VolumeType": "[volumeType]"
    }
  }
}
```

Using `AADClientCertificate`:

JSON

```
{
  "type": "extensions",
  "name": "[name]",
  "apiVersion": "2019-07-01",
  "location": "[location]",
  "properties": {
    "protectedSettings": {
      "AADClientCertificate": "[aadClientCertificate]",
      "Passphrase": "[passphrase]"
    },
    "publisher": "Microsoft.Azure.Security",
    "type": "AzureDiskEncryptionForLinux",
    "typeHandlerVersion": "0.1",
    "settings": {
      "AADClientID": "[aadClientID]",
      "DiskFormatQuery": "[diskFormatQuery]",
      "EncryptionOperation": "[encryptionOperation]",
      "KeyEncryptionAlgorithm": "[keyEncryptionAlgorithm]",
      "KeyEncryptionKeyURL": "[keyEncryptionKeyURL]",
      "KeyVaultURL": "[keyVaultURL]",
      "SequenceVersion": "sequenceVersion]",
      "VolumeType": "[volumeType]"
    }
  }
}
```

Property values

Note: All property values are case sensitive.

Expand table

Name	Value / Example	Data Type
apiVersion	2019-07-01	date
publisher	Microsoft.Azure.Security	string
type	AzureDiskEncryptionForLinux	string
typeHandlerVersion	1.1, 0.1	int
(0.1 schema) AADClientID	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	guid
(0.1 schema) AADClientSecret	password	string
(0.1 schema) AADClientCertificate	thumbprint	string
(optional) (0.1 schema) Passphrase	password	string
DiskFormatQuery	{"dev_path":"","name":"","file_system":""}	JSON dictionary
EncryptionOperation	EnableEncryption, EnableEncryptionFormatAll	string
(optional - default RSA-OAEP) KeyEncryptionAlgorithm	'RSA-OAEP', 'RSA-OAEP-256', 'RSA1_5'	string
KeyVaultURL	url	string
KeyVaultResourceId	url	string
(optional) KeyEncryptionKeyURL	url	string
(optional) KekVaultResourceId	url	string
(optional) SequenceVersion	uniqueidentifier	string
VolumeType	OS, Data, All	string

Template deployment

For an example of template deployment based on schema v1.1, see the Azure Quickstart Template encrypt-running-linux-vm-without-aad.

For an example of template deployment based on schema v0.1, see the Azure Quickstart Template encrypt-running-linux-vm.

> ⚠ **Warning**
>
> - If you have previously used Azure Disk Encryption with Microsoft Entra ID to encrypt a VM, you must continue use this option to encrypt your VM.
> - When encrypting Linux OS volumes, the VM should be considered unavailable. We strongly recommend to avoid SSH logins while the encryption is in progress to avoid issues blocking any open files that will need to be accessed during the encryption process. To check progress, use the **Get-AzVMDiskEncryptionStatus** PowerShell cmdlet or the **vm encryption show** CLI command. This process can be expected to take a few hours for a 30GB OS volume, plus additional time for encrypting data volumes. Data volume encryption time will be proportional to the size and quantity of the data volumes; the `encrypt format all` option is faster than in-place encryption, but will result in the loss of all data on the disks.
> - Disabling encryption on Linux VMs is only supported for data volumes. It is not supported on data or OS volumes if the OS volume has been encrypted.

> ⓘ **Note**
>
> Also if `VolumeType` parameter is set to All, data disks will be encrypted only if they are properly mounted.

Troubleshoot and support

Troubleshoot

For troubleshooting, refer to the Azure Disk Encryption troubleshooting guide.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums.

Alternatively, you can file an Azure support incident. Go to Azure support and select Get support. For information about using Azure Support, read the Microsoft Azure Support FAQ.

Next steps

- For more information about VM extensions, see Virtual machine extensions and features for Linux.
- For more information about Azure Disk Encryption for Linux, see Linux virtual machines.

Last updated on 11/10/2025

Azure Disk Encryption for Windows (Microsoft.Azure.Security.AzureDiskEncryption)

Overview

Azure Disk Encryption uses BitLocker to provide full disk encryption on Azure virtual machines running Windows. This solution is integrated with Azure Key Vault to manage disk encryption keys and secrets in your key vault subscription.

> ⓘ **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

For a full list of prerequisites, see Azure Disk Encryption for Windows VMs, specifically the following sections:

- Supported VMs and operating systems
- Networking requirements
- Group Policy requirements

Extension Schema

There are two versions of extension schema for Azure Disk Encryption (ADE):

- v2.2 - A newer recommended schema that does not use Microsoft Entra properties.
- v1.1 - An older schema that requires Microsoft Entra properties.

To select a target schema, the `typeHandlerVersion` property must be set equal to version of schema you want to use.

Schema v2.2: No Microsoft Entra ID (recommended)

The v2.2 schema is recommended for all new VMs and does not require Microsoft Entra properties.

JSON

```
{
  "type": "extensions",
  "name": "[name]",
  "apiVersion": "2019-07-01",
  "location": "[location]",
  "properties": {
        "publisher": "Microsoft.Azure.Security",
        "type": "AzureDiskEncryption",
        "typeHandlerVersion": "2.2",
        "autoUpgradeMinorVersion": true,
        "settings": {
          "EncryptionOperation": "[encryptionOperation]",
          "KeyEncryptionAlgorithm": "[keyEncryptionAlgorithm]",
          "KeyVaultURL": "[keyVaultURL]",
          "KeyVaultResourceId": "[keyVaultResourceID]",
          "KeyEncryptionKeyURL": "[keyEncryptionKeyURL]",
          "KekVaultResourceId": "[kekVaultResourceID]",
          "SequenceVersion": "sequenceVersion]",
          "VolumeType": "[volumeType]"
        }
  }
}
```

Schema v1.1: with Microsoft Entra ID

The 1.1 schema requires `aadClientID` and either `aadClientSecret` or `AADClientCertificate` and is not recommended for new VMs.

Using `aadClientSecret`:

JSON

```
{
  "type": "extensions",
  "name": "[name]",
  "apiVersion": "2019-07-01",
  "location": "[location]",
  "properties": {
    "protectedSettings": {
      "AADClientSecret": "[aadClientSecret]"
    },
    "publisher": "Microsoft.Azure.Security",
    "type": "AzureDiskEncryption",
    "typeHandlerVersion": "1.1",
    "settings": {
```

```
      "AADClientID": "[aadClientID]",
      "EncryptionOperation": "[encryptionOperation]",
      "KeyEncryptionAlgorithm": "[keyEncryptionAlgorithm]",
      "KeyVaultURL": "[keyVaultURL]",
      "KeyVaultResourceId": "[keyVaultResourceID]",
      "KeyEncryptionKeyURL": "[keyEncryptionKeyURL]",
      "KekVaultResourceId": "[kekVaultResourceID]",
      "SequenceVersion": "sequenceVersion]",
      "VolumeType": "[volumeType]"
    }
  }
}
```

Using `AADClientCertificate`:

JSON

```
{
  "type": "extensions",
  "name": "[name]",
  "apiVersion": "2019-07-01",
  "location": "[location]",
  "properties": {
    "protectedSettings": {
      "AADClientCertificate": "[aadClientCertificate]"
    },
    "publisher": "Microsoft.Azure.Security",
    "type": "AzureDiskEncryption",
    "typeHandlerVersion": "1.1",
    "settings": {
      "AADClientID": "[aadClientID]",
      "EncryptionOperation": "[encryptionOperation]",
      "KeyEncryptionAlgorithm": "[keyEncryptionAlgorithm]",
      "KeyVaultURL": "[keyVaultURL]",
      "KeyVaultResourceId": "[keyVaultResourceID]",
      "KeyEncryptionKeyURL": "[keyEncryptionKeyURL]",
      "KekVaultResourceId": "[kekVaultResourceID]",
      "SequenceVersion": "sequenceVersion]",
      "VolumeType": "[volumeType]"
    }
  }
}
```

Property values

Note: All values are case sensitive.

Expand table

Name	Value / Example	Data Type
apiVersion	2019-07-01	date
publisher	Microsoft.Azure.Security	string
type	AzureDiskEncryption	string
typeHandlerVersion	2.2, 1.1	string
(1.1 schema) AADClientID	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	guid
(1.1 schema) AADClientSecret	password	string
(1.1 schema) AADClientCertificate	thumbprint	string
EncryptionOperation	EnableEncryption	string
(optional - default RSA-OAEP) KeyEncryptionAlgorithm	'RSA-OAEP', 'RSA-OAEP-256', 'RSA1_5'	string
KeyVaultURL	url	string
KeyVaultResourceId	url	string
(optional) KeyEncryptionKeyURL	url	string
(optional) KekVaultResourceId	url	string
(optional) SequenceVersion	uniqueidentifier	string
VolumeType	OS, Data, All	string

Template deployment

For an example of template deployment based on schema v2.2, see Azure Quickstart Template encrypt-running-windows-vm-without-aad.

For an example of template deployment based on schema v1.1, see Azure Quickstart Template encrypt-running-windows-vm.

> **Note**
>
> Also if `VolumeType` parameter is set to All, data disks will be encrypted only if they are properly formatted.

Troubleshoot and support

Troubleshoot

For troubleshooting, refer to the Azure Disk Encryption troubleshooting guide.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums.

Alternatively, you can file an Azure support incident. Go to Azure support and select Get support. For information about using Azure Support, read the Microsoft Azure Support FAQ.

Next steps

- For more information about extensions, see Virtual machine extensions and features for Windows.
- For more information about Azure Disk Encryption for Windows, see Windows virtual machines.

Last updated on 11/10/2025

Key Vault virtual machine extension for Linux

Summarize this article for me

The Key Vault VM extension provides automatic refresh of certificates stored in an Azure key vault. Specifically, the extension monitors a list of observed certificates stored in key vaults. The extension retrieves and installs the corresponding certificates after detecting a change. This document details the supported platforms, configurations, and deployment options for the Key Vault VM extension for Linux.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Operating system

The Key Vault VM extension supports:

- Ubuntu 22.04 and later
- Azure Linux

Supported certificate content types

- PKCS #12
- PEM

Features

Key Vault VM extension for Linux version 3.0+ supports:

- Add ACL permissions for downloaded certificates to provide read access for users and groups
- Certificate installation location configuration
- Custom symbolic name support
- VM extension logging integration support through Fluentd

Prerequisites

- Key Vault instance with certificate. See Create a Key Vault

- Assigned managed identity on virtual machines/virtual machine scale sets.
- The **Key Vault Secrets User** role at the Key Vault scope level for VMs and Azure Virtual Machine Scale Sets managed identity. This role retrieves a secret's portion of a certificate. For more information, see the following articles:
  - Authentication in Azure Key Vault
  - Use Azure RBAC secret, key, and certificate permissions with Azure Key Vault
  - Key Vault scope role assignment
- Virtual machine scale sets should have the following identity setting: `"identity": { "type": "UserAssigned", "userAssignedIdentities": { "[parameters('userAssignedIdentityResourceId')]": {} } }`
- AKV extension should have this setting: `"authenticationSettings": { "msiEndpoint": "[parameters('userAssignedIdentityEndpoint')]", "msiClientId": "[reference(parameters('userAssignedIdentityResourceId'), variables('msiApiVersion')).clientId]" }`

Upgrading Key Vault VM extension

- If you need to upgrade from an older version to version 3.0+, you would need to delete previous version first, then install version 3.0.

Azure CLI

```
  az vm extension delete --name KeyVaultForLinux --resource-group ${resourceGroup} --vm-name ${vmName}
  az vm extension set -n "KeyVaultForLinux" --publisher Microsoft.Azure.KeyVault --resource-group "${resourceGroup}" --vm-name "${vmName}" -settings .\akvvm.json -version 3.0
```

- If the VM has certificates downloaded by previous version, deleting VM extension doesn't delete the downloaded certificates. After installing the newer version, the existing certificates aren't modified. You would need to delete the certificate files or roll-over the certificate to get the PEM file with full-chain on the VM.

Extension schema

The following JSON provides the schema for the Key Vault VM extension. All settings are specified as plain (unprotected) settings, as none are considered sensitive. To configure the extension, you must specify a list of certificates to monitor, how often to poll for updates, and the destination path for storing certificates. Specifically:

JSON

```
    {
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "name": "KVVMExtensionForLinux",
      "apiVersion": "2022-11-01",
      "location": "<location>",
      "dependsOn": [
          "[concat('Microsoft.Compute/virtualMachines/', <vmName>)]"
      ],
      "properties": {
      "publisher": "Microsoft.Azure.KeyVault",
      "type": "KeyVaultForLinux",
      "typeHandlerVersion": "3.0",
      "autoUpgradeMinorVersion": true,
      "enableAutomaticUpgrade": true,
      "settings": {
      "loggingSettings": <Optional logging settings, e.g.:
        {
              "logger": <Logger engine name. e.g.: "fluentd">,
              "endpoint": <Logger listening endpoint "tcp://localhost:24224">,
              "format": <Logging format. e.g.: "forward">,
              "servicename": <Service name used in logs. e.g.: "akvvm_service">
        }>,
        "secretsManagementSettings": {
          "pollingIntervalInS": <polling interval in seconds, e.g. "3600">,
          "linkOnRenewal": <Not available on Linux e.g.: false>,
          "requireInitialSync": <initial synchronization of certificates e..g: true>,
          "aclEnabled": <Enables ACLs for downloaded certificates, e.g.: true>,
          "observedCertificates": <An array of KeyVault URIs that represent monitored
certificates, including certificate store location, ACL permission to certificate private
key, and custom symbolic name. e.g.:
             [
                {
                    "url": <A Key Vault URI to the secret portion of the certificate.
e.g.: "https://myvault.vault.azure.net/secrets/mycertificate1">,
                    "certificateStoreLocation": <disk path where certificate is stored,
e.g.: "/var/lib/waagent/Microsoft.Azure.KeyVault/app1">,
                    "customSymbolicLinkName": <symbolic name for the certificate. e.g.:
"app1Cert1">,
                    "acls": [
                        {
                            "user": "app1",
                            "group": "appGroup1"
                        },
                        {
                            "user": "service1"
                        }
                    ]
                },
                {
                    "url": <Example:
"https://myvault.vault.azure.net/secrets/mycertificate2">,
                    "certificateStoreLocation": <disk path where the certificate is
stored, e.g.: "/var/lib/waagent/Microsoft.Azure.KeyVault/app2">,
                    "acls": [
                        {
                            "user": "app2",
                        }
```

```
                    ]
                }
            ]>
        },
        "authenticationSettings": <Optional msi settings, e.g.:
        {
          "msiEndpoint":  <Required when msiClientId is provided. MSI endpoint e.g. for
most Azure VMs: "http://169.254.169.254/metadata/identity">,
          "msiClientId":  <Required when VM has any user assigned identities. MSI
identity e.g.: "00001111-aaaa-2222-bbbb-3333cccc4444".>
        }>
       }
      }
    }
```

> **Note**
>
> Your observed certificates URLs should be of the form `https://myVaultName.vault.azure.net/secrets/myCertName`.
>
> This is because the `/secrets` path returns the full certificate, including the private key, while the `/certificates` path doesn't. More information about certificates can be found here: Key Vault Certificates

> **Important**
>
> The `authenticationSettings` property is required only when your VM or VM scale set uses **user-assigned managed identities** or when using **Azure Arc-enabled VMs**. For **system-assigned managed identities**, omit the `authenticationSettings` section. Including it will cause deployment to fail. Without this section, a VM with user assigned identities will result in the Key Vault extension failing and being unable to download certificates. Set `msiClientId` to the identity that will authenticate to Key Vault.
>
> Also **required** for **Azure Arc-enabled VMs**. Set msiEndpoint to `http://localhost:40342/metadata/identity`.

Property values

Expand table

Name	Value / Example	Data Type
`apiVersion`	2022-07-01	date
`publisher`	Microsoft.Azure.KeyVault	string

Name	Value / Example	Data Type
`type`	KeyVaultForLinux	string
`typeHandlerVersion`	3.0	int
`pollingIntervalInS`	3600	string
`certificateStoreName`	It's ignored on Linux	string
`linkOnRenewal`	false	boolean
`requireInitialSync`	true	boolean
`aclEnabled`	true	boolean
`certificateStoreLocation`	/var/lib/waagent/Microsoft.Azure.KeyVault.Store	string
`observedCertificates`	[{...}, {...}]	string array
`observedCertificates/url`	"https://myvault.vault.azure.net/secrets/mycertificate1"	string
`observedCertificates/certificateStoreLocation`	"/var/lib/waagent/Microsoft.Azure.KeyVault/app1"	string
`observedCertificates/customSymbolicLinkName` (optional)	"app1Cert1"	string
`observedCertificates/acls` (optional)	"{...}, {...}"	string array
`authenticationSettings` (optional)	{...}	object
`authenticationSettings/msiEndpoint`	http://169.254.169.254/metadata/identity	string
`authenticationSettings/msiClientId`	00001111-aaaa-2222-bbbb-3333cccc4444	string
`loggingSettings` (optional)	{...}	object
`loggingSettings/logger`	"fluentd"	string
`loggingSettings/endpoint`	"tcp://localhost:24224"	string
`loggingSettings/format`	"forward"	string
`loggingSettings/servicename`	"akvvm_service"	string

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. Templates are ideal when deploying one or more virtual machines that require post deployment refresh of certificates.

The extension can be deployed to individual VMs or virtual machine scale sets. The schema and configuration are common to both template types.

> **Note**
>
> The VM extension would require system or user managed identity to be assigned to authenticate to Key vault. See How to authenticate to Key Vault and assign a Key Vault access policy.

JSON

```
   {
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "name": "KeyVaultForLinux",
      "apiVersion": "2022-11-01",
      "location": "<location>",
      "dependsOn": [
          "[concat('Microsoft.Compute/virtualMachines/', <vmName>)]"
      ],
      "properties": {
      "publisher": "Microsoft.Azure.KeyVault",
      "type": "KeyVaultForLinux",
      "typeHandlerVersion": "3.0",
      "autoUpgradeMinorVersion": true,
      "enableAutomaticUpgrade": true,
      "settings": {
          "secretsManagementSettings": {
          "pollingIntervalInS": <polling interval in seconds, e.g. "3600">,
          "requireInitialSync": <initial synchronization of certificates e..g: false>,
          "aclEnabled": <enables/disables acls on defined certificates e.g.: true>,
          "observedCertificates": <An array of KeyVault URIs that represent monitored
certificates, including certificate store location and ACL permission to certificate
private key. Example:
              [
                 {
                     "url": <A Key Vault URI to the secret portion of the certificate.
Example: "https://myvault.vault.azure.net/secrets/mycertificate1">,
                     "certificateStoreLocation": <The certificate store location, which
currently works locally only. Example:
"/var/lib/waagent/Microsoft.Azure.KeyVault.Store">,
                     "acls": <Optional. An array of preferred acls with read access to
certificate private keys. Example:
                     [
                         {
                             "user": "app1",
                             "group": "appGroup1"
                         },
                         {
                             "user": "service1"
                         }
                     ]>
                 },
                 {
                     "url": <Example:
```

```
"https://myvault.vault.azure.net/secrets/mycertificate2">,
                    "certificateStoreName": <ignored on linux>,
                    "certificateStoreLocation": <The certificate store location, which
currently works locally only. Example:
"/var/lib/waagent/Microsoft.Azure.KeyVault.Store">,
                    "acls": <Optional. An array of preferred acls with read access to
certificate private keys. Example:
                    [
                        {
                            "user": "app2"
                        }
                    ]>
                }

            ]>
        },
        "authenticationSettings": {
            "msiEndpoint":  <Required when msiClientId is provided. MSI endpoint e.g.
for most Azure VMs: "http://169.254.169.254/metadata/identity">,
            "msiClientId":  <Required when VM has any user assigned identities. MSI
identity e.g.: "00001111-aaaa-2222-bbbb-3333cccc4444">
        }
      }
    }
  }
```

Extension Dependency Ordering

The Key Vault VM extension supports extension ordering if configured. By default the extension reports successful start as soon as polling starts. However, you can configure it to wait until it successfully downloads the complete list of certificates before reporting a successful start. If other extensions depend on installed certificates before they start, then enabling this setting will allow those extensions to declare a dependency on the Key Vault extension. This will prevent those extensions from starting until all certificates they depend on have been installed.

The extension will retry the initial download up to 25 times with increasing backoff periods, during which it remains in a `Transitioning` state. If the retries are exhausted, the extension will report an `Error` state.

To turn on extension dependency, set the following:

```
"secretsManagementSettings": {
    "requireInitialSync": true,
    ...
}
```

> ⓘ Note

Using this feature is not compatible with an ARM template that creates a system assigned identity and updates a Key Vault access policy with that identity. Doing so will result in a deadlock as the vault access policy cannot be updated until all extensions have started. You should instead use a *single user assigned MSI identity* and pre-ACL your vaults with that identity before deploying.

Azure PowerShell deployment

> ⚠ **Warning**
>
> PowerShell clients often add `\` to `"` in the settings.json which will cause akvvm_service fails with error: `[CertificateManagementConfiguration] Failed to parse the configuration settings with:not an object.`

The Azure PowerShell can be used to deploy the Key Vault VM extension to an existing virtual machine or virtual machine scale set.

- To deploy the extension on a VM:

The Azure Key Vault VM extension can be deployed with Azure PowerShell. Save Key Vault VM extension settings to a JSON file (settings.json).

The following JSON snippets provide example settings for deploying the Key Vault VM extension with PowerShell.

JSON

```
{
   "secretsManagementSettings": {
   "pollingIntervalInS": "3600",
   "linkOnRenewal": true,
   "aclEnabled": true,
   "observedCertificates":
   [
      {
          "url": "https://<examplekv>.vault.azure.net/secrets/mycertificate1",
          "certificateStoreLocation":  "/var/lib/waagent/Microsoft.Azure.KeyVault.Store",
          "acls":
          [
              {
                  "user": "app1",
                  "group": "appGroup1"
              },
              {
                  "user": "service1"
              }
          ]
      },
```

```
        {
            "url": "https://<examplekv>.vault.azure.net/secrets/mycertificate2",
            "certificateStoreLocation": "/var/lib/waagent/Microsoft.Azure.KeyVault.Store",
            "acls":
            [
                {
                    "user": "app2"
                }
            ]
        }
    ]},
    "authenticationSettings": {
        "msiEndpoint":  "http://169.254.169.254/metadata/identity/oauth2/token",
        "msiClientId":  "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxx"
    }
}
```

- To deploy the extension on a virtual machine:

PowerShell

```
# Build settings
$settings = (get-content -raw ".\settings.json")
$extName =  "KeyVaultForLinux"
$extPublisher = "Microsoft.Azure.KeyVault"
$extType = "KeyVaultForLinux"

# Start the deployment
Set-AzVmExtension -TypeHandlerVersion "3.0" -ResourceGroupName <ResourceGroupName> -
Location <Location> -VMName <VMName> -Name $extName -Publisher $extPublisher -Type
$extType -SettingString $settings
```

- To deploy the extension on a virtual machine scale set:

PowerShell

```
    # Build settings
    $settings = (get-content -raw ".\settings.json")
    $extName = "KeyVaultForLinux"
    $extPublisher = "Microsoft.Azure.KeyVault"
    $extType = "KeyVaultForLinux"

    # Add extension to Virtual Machine Scale Sets
    $vmss = Get-AzVmss -ResourceGroupName <ResourceGroupName> -VMScaleSetName <VmssName>
    Add-AzVmssExtension -VirtualMachineScaleSet $vmss  -Name $extName -Publisher
$extPublisher -Type $extType -TypeHandlerVersion "3.0" -Setting $settings

    # Start the deployment
    Update-AzVmss -ResourceGroupName <ResourceGroupName> -VMScaleSetName <VmssName> -
VirtualMachineScaleSet $vmss
```

Azure CLI deployment

The Azure CLI can be used to deploy the Key Vault VM extension to an existing virtual machine or virtual machine scale set.

- To deploy the extension on a VM:

The Azure Key Vault VM extension can be deployed by using the Azure CLI. Save Key Vault VM extension settings to a JSON file (settings.json).

The following JSON snippets provide example settings for deploying the Key Vault VM extension with the Azure CLI.

JSON

```
{
   "secretsManagementSettings": {
   "pollingIntervalInS": "3600",
   "linkOnRenewal": true,
   "aclEnabled": true,
   "observedCertificates":
   [
      {
          "url": "https://<examplekv>.vault.azure.net/secrets/mycertificate1",
          "certificateStoreLocation":  "/var/lib/waagent/Microsoft.Azure.KeyVault.Store",
          "acls":
          [
              {
                  "user": "app1",
                  "group": "appGroup1"
              },
              {
                  "user": "service1"
              }
          ]
      },
      {
          "url": "https://<examplekv>.vault.azure.net/secrets/mycertificate2",
          "certificateStoreLocation": "/var/lib/waagent/Microsoft.Azure.KeyVault.Store",
          "acls":
          [
              {
                  "user": "app2"
              }
          ]
      }
   ]},
   "authenticationSettings": {
      "msiEndpoint":  "http://169.254.169.254/metadata/identity/oauth2/token",
      "msiClientId":  "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxx"
   }
}
```

- To deploy the extension on a virtual machine

Azure CLI

```
# Start the deployment
  az vm extension set -n "KeyVaultForLinux" `
  --publisher Microsoft.Azure.KeyVault `
  -g "<resourcegroup>" `
  --vm-name "<vmName>" `
  --version 3.0 `
  --enable-auto-upgrade true `
  --settings "@settings.json"
```

- To deploy the extension on a virtual machine scale set:

Azure CLI

```
# Start the deployment
az vmss extension set -n "KeyVaultForLinux" `
--publisher Microsoft.Azure.KeyVault `
-g "<resourcegroup>" `
--vmss-name "<vmssName>" `
--version 3.0 `
--enable-auto-upgrade true `
--settings "@settings.json"
```

Please be aware of the following restrictions/requirements:

- Key Vault restrictions:
  - It must exist at the time of the deployment
  - The Key Vault Secrets User role must be assigned to Key Vault for VM identity

Troubleshoot and Support

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure PowerShell. To see the deployment state of extensions for a given VM, run the following command using the Azure PowerShell.

Azure PowerShell

PowerShell

```
Get-AzVMExtension -VMName <vmName> -ResourceGroupname <resource group name>
```

Azure CLI

Azure CLI

```
 az vm get-instance-view --resource-group <resource group name> --name  <vmName> --query "instanceView.extensions"
```

The Azure CLI can run in several shell environments, but with slight format variations. If you have unexpected results with Azure CLI commands, see How to use the Azure CLI successfully.

Logs and configuration

The Key Vault VM extension logs exist locally on the VM and are most informative when it comes to troubleshooting. You can use optional logging section to integrate with logging provider through `fluentd`

Expand table

Location	Description
/var/log/waagent.log	Shows when an update to the extension occurred.
/var/log/azure/Microsoft.Azure.KeyVault.KeyVaultForLinux/*	Examine the Key Vault VM Extension service logs to determine the status of the akvvm_service service and certificate download. You can find the download location of PEM files in files with an entry called certificate file name. If certificateStoreLocation is not specified, it will default to /var/lib/waagent/Microsoft.Azure.KeyVault.Store/
/var/lib/waagent/Microsoft.Azure.KeyVault.KeyVaultForLinux-<most recent version>/config/*	The configuration and binaries for Key Vault VM Extension service.

Using Symlink

Symbolic links or Symlinks are advanced shortcuts. To avoid monitoring the folder and to get the latest certificate automatically, you can use this symlink `([VaultName].[CertificateName])` to get the latest version of certificate on Linux.

Certificate Installation on Linux

The Key Vault VM extension for Linux installs certificates as PEM files. When a certificate is downloaded from Key Vault, the extension:

1. Creates a storage folder based on the `certificateStoreLocation` setting (defaults to `/var/lib/waagent/Microsoft.Azure.KeyVault.Store/` if not specified)

2. Installs the certificate chain and private key as stored in Key Vault. The PEM file follows RFC 5246 section 7.4.2 ordering:
    - Leaf certificate (end-entity certificate) comes first
    - Intermediate certificate(s) follow in order, where each certificate directly certifies the one before it (if present in Key Vault)
    - Root certificate, if present (though it's not required for validation if already trusted by the system)
    - Private key, corresponding to the leaf certificate, is placed at the end of the file
3. Automatically creates a symbolic link named `[VaultName].[CertificateName]` that points to the latest version of the certificate

This approach ensures that:

- Applications have access to the certificate chain as stored in Key Vault
- The certificate chain is properly ordered for TLS handshakes according to RFC standards
- The private key is available for use by the service
- Applications can reference a stable symbolic link path that automatically updates when certificates are renewed
- No application reconfiguration is needed when certificates are rotated or renewed

Example Certificate Path Structure

For a certificate from `exampleVault.vault.azure.net` with the name `myCertificate`, the directory structure would look like:

```
/var/lib/waagent/Microsoft.Azure.KeyVault.Store/
├── exampleVault.myCertificate -> exampleVault.myCertificate.1234567890abcdef
├── exampleVault.myCertificate.1234567890abcdef    # Full chain PEM file (current
version)
└── exampleVault.myCertificate.0987654321fedcba    # Previous version (if exists)
```

Applications should be configured to use the symbolic link path (`/var/lib/waagent/Microsoft.Azure.KeyVault.Store/exampleVault.myCertificate`) to ensure they always access the most current certificate version.

When you use custom certificate store locations and the `customSymbolicLinkName` setting, the structure follows this pattern:

```
/path/to/custom/store/
├── customLinkName -> exampleVault.myCertificate.1234567890abcdef
```

```
└── exampleVault.myCertificate.1234567890abcdef    # Full chain PEM file
```

Frequently Asked Questions

- Is there's a limit on the number of observedCertificates you can configure? No, Key Vault VM Extension doesn't have limit on the number of observedCertificates.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 02/26/2026

Azure Key Vault virtual machine extension for Windows

The Azure Key Vault virtual machine (VM) extension provides automatic refresh of certificates stored in an Azure key vault. The extension monitors a list of observed certificates stored in key vaults. When it detects a change, the extension retrieves and installs the corresponding certificates. This article describes the supported platforms, configurations, and deployment options for the Key Vault VM extension for Windows.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Operating systems

The Key Vault VM extension supports Windows Server 2019 and later. The Key Vault VM extension is also supported on a custom local VM. The VM should be uploaded and converted into a specialized image for use in Azure by using Windows Server 2019 core install.

Supported certificates

The Key Vault VM extension supports the following certificate content types:

- PKCS #12
- PEM

> **Note**
>
> The Key Vault VM extension downloads all certificates to the Windows certificate store or to the location specified in the `certificateStoreLocation` property in the VM extension settings.

Features

The Key Vault VM extension for Windows version 3.0 supports:

- Add ACL permissions to downloaded certificates
- Enable Certificate Store configuration per certificate

- Export private keys
- IIS Certificate Rebind support

Prerequisites

Review the following prerequisites for using the Key Vault VM extension for Windows:

- An Azure Key Vault instance with a certificate. For more information, see Create a key vault by using the Azure portal.
- A VM with an assigned managed identity.
- The **Key Vault Secrets User** role must be assigned at the Key Vault scope level for VMs and Azure Virtual Machine Scale Sets managed identity. This role retrieves a secret's portion of a certificate. For more information, see the following articles:
  - Authentication in Azure Key Vault
  - Use Azure RBAC secret, key, and certificate permissions with Azure Key Vault
  - Key Vault scope role assignment
- Virtual Machine Scale Sets should have the following `identity` configuration:

  JSON

  ```
  "identity": {
     "type": "UserAssigned",
     "userAssignedIdentities": {
        "[parameters('userAssignedIdentityResourceId')]": {}
     }
  }
  ```

- The Key Vault VM extension should have the following `authenticationSettings` configuration:

  JSON

  ```
  "authenticationSettings": {
      "msiEndpoint": "[parameters('userAssignedIdentityEndpoint')]",
      "msiClientId": "[reference(parameters('userAssignedIdentityResourceId'),
  variables('msiApiVersion')).clientId]"
  }
  ```

> **Note**
>
> The old access policy permission model can also be used to provide access to VMs and Virtual Machine Scale Sets. This method requires policy with **get** and **list** permissions on secrets. For more information, see Assign a Key Vault access policy.

Extension schema

The following JSON shows the schema for the Key Vault VM extension. Before you consider the schema implementation options, review the following important notes.

- The extension doesn't require protected settings. All settings are considered public information.
- Observed certificates URLs should be of the form `https://myVaultName.vault.azure.net/secrets/myCertName`.

  This form is preferred because the `/secrets` path returns the full certificate, including the private key, but the `/certificates` path doesn't. For more information about certificates, see Azure Key Vault keys, secrets and certificates overview.
- The `authenticationSettings` property is **required** for VMs with any **user assigned identities**.

  This property specifies the identity to use for authentication to Key Vault. Define this property with a system-assigned identity to avoid issues with a VM extension with multiple identities.

JSON

```
{
   "type": "Microsoft.Compute/virtualMachines/extensions",
   "name": "KVVMExtensionForWindows",
   "apiVersion": "2022-08-01",
   "location": "<location>",
   "dependsOn": [
      "[concat('Microsoft.Compute/virtualMachines/', <vmName>)]"
   ],
   "properties": {
      "publisher": "Microsoft.Azure.KeyVault",
      "type": "KeyVaultForWindows",
      "typeHandlerVersion": "3.0",
      "autoUpgradeMinorVersion": true,
      "settings": {
         "secretsManagementSettings": {
             "pollingIntervalInS": <A string that specifies the polling interval in seconds. Example: "3600">,
             "linkOnRenewal": <Windows only. Ensures s-channel binding when the certificate renews without necessitating redeployment. Example: true>,
             "requireInitialSync": <Initial synchronization of certificates. Example: true>,
             "observedCertificates": <An array of KeyVault URIs that represent monitored certificates, including certificate store location and ACL permission to certificate private key. Example:
             [
                {
                    "url": <A Key Vault URI to the secret portion of the certificate. Example: "https://myvault.vault.azure.net/secrets/mycertificate1">,
                    "certificateStoreName": <The certificate store name. Example: "MY">,
                    "certificateStoreLocation": <The certificate store location, which currently works locally only. Example: "LocalMachine">,
                    "accounts": <Optional. An array of preferred accounts with read access to certificate private keys. Administrators and SYSTEM get Full Control by default. Example: ["Network Service", "Local Service"]>
                },
                {
                    "url": <Example:
```

```
"https://myvault.vault.azure.net/secrets/mycertificate2">,
                    "certificateStoreName": <Example: "MY">,
                    "certificateStoreLocation": <Example: "CurrentUser">,
                    "keyExportable": <Optional. Lets the private key be exportable. Example:
"false">,
                    "accounts": <Example: ["Local Service"]>
                }
            ]>
        },
        "authenticationSettings": {
            "msiEndpoint":  <Required when the msiClientId property is used. Specifies the
MSI endpoint. Example for most Azure VMs:
"http://169.254.169.254/metadata/identity/oauth2/token">,
            "msiClientId":  <Required when the VM has any user assigned identities.
Specifies the MSI identity. Example:  "00001111-aaaa-2222-bbbb-3333cccc4444">
        }
    }
  }
}
```

Property values

The JSON schema includes the following properties.

Expand table

Name	Value/Example	Data type
`apiVersion`	2022-08-01	date
`publisher`	Microsoft.Azure.KeyVault	string
`type`	KeyVaultForWindows	string
`typeHandlerVersion`	"3.0"	string
`pollingIntervalInS`	"3600"	string
`linkOnRenewal` (optional)	true	boolean
`requireInitialSync` (optional)	false	boolean
`observedCertificates`	[{...}, {...}]	string array
`observedCertificates/url`	"https://myvault.vault.azure.net/secrets/mycertificate"	string
`observedCertificates/certificateStoreName`	MY	string
`observedCertificates/certificateStoreLocation`	LocalMachine or CurrentUser (case sensitive)	string
`observedCertificates/keyExportable` (optional)	false	boolean

Name	Value/Example	Data type
`observedCertificates/accounts` (optional)	["Network Service", "Local Service"]	string array
`msiEndpoint`	"http://169.254.169.254/metadata/identity/oauth2/token"	string
`msiClientId`	00001111-aaaa-2222-bbbb-3333cccc4444	string

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager (ARM) templates. Templates are ideal when deploying one or more virtual machines that require post deployment refresh of certificates. The extension can be deployed to individual VMs or Virtual Machine Scale Sets instances. The schema and configuration are common to both template types.

The JSON configuration for a key vault extension is nested inside the VM or Virtual Machine Scale Sets template. For a VM resource extension, the configuration is nested under the `"resources": []` virtual machine object. For a Virtual Machine Scale Sets instance extension, the configuration is nested under the `"virtualMachineProfile":"extensionProfile":{"extensions" :[]` object.

The following JSON snippets provide example settings for an ARM template deployment of the Key Vault VM extension.

JSON

```
{
   "type": "Microsoft.Compute/virtualMachines/extensions",
   "name": "KeyVaultForWindows",
   "apiVersion": "2022-08-01",
   "location": "<location>",
   "dependsOn": [
      "[concat('Microsoft.Compute/virtualMachines/', <vmName>)]"
   ],
   "properties": {
      "publisher": "Microsoft.Azure.KeyVault",
      "type": "KeyVaultForWindows",
      "typeHandlerVersion": "3.0",
      "autoUpgradeMinorVersion": true,
      "settings": {
         "secretsManagementSettings": {
             "pollingIntervalInS": <A string that specifies the polling interval in seconds.
Example: "3600">,
             "linkOnRenewal": <Windows only. Ensures s-channel binding when the certificate
renews without necessitating redeployment. Example: true>,
             "observedCertificates": <An array of KeyVault URIs that represent monitored
certificates, including certificate store location and ACL permission to certificate private
key. Example:
             [
                {
                    "url": <A Key Vault URI to the secret portion of the certificate.
Example: "https://myvault.vault.azure.net/secrets/mycertificate1">,
```

```
                    "certificateStoreName": <The certificate store name. Example: "MY">,
                    "certificateStoreLocation": <The certificate store location, which
currently works locally only. Example: "LocalMachine">,
                    "accounts": <Optional. An array of preferred accounts with read access
to certificate private keys. Administrators and SYSTEM get Full Control by default. Example:
["Network Service", "Local Service"]>
                },
                {
                    "url": <Example:
"https://myvault.vault.azure.net/secrets/mycertificate2">,
                    "certificateStoreName": <Example: "MY">,
                    "certificateStoreLocation": <Example: "CurrentUser">,
                    "keyExportable": <Optional. Lets the private key be exportable. Example:
"false">,
                    "accounts": <Example: ["Local Service"]>
                },
                {
                    "url": <Example:
"https://myvault.vault.azure.net/secrets/mycertificate3">,
                    "certificateStoreName": <Example: "TrustedPeople">,
                    "certificateStoreLocation": <Example: "LocalMachine">
                }
             ]>
         },
         "authenticationSettings": {
            "msiEndpoint":  <Required when the msiClientId property is used. Specifies the
MSI endpoint. Example for most Azure VMs:
"http://169.254.169.254/metadata/identity/oauth2/token">,
            "msiClientId":  <Required when the VM has any user assigned identities.
Specifies the MSI identity. Example: "00001111-aaaa-2222-bbbb-3333cccc4444">
         }
      }
   }
}
```

Extension dependency ordering

You can enable the Key Vault VM extension to support extension dependency ordering. By default, the Key Vault VM extension reports a successful start as soon as polling begins. However, you can configure the extension to report a successful start only after the extension downloads and installs all certificates.

If you use other extensions that require installation of all certificates before they start, you can enable extension dependency ordering in the Key Vault VM extension. This feature allows other extensions to declare a dependency on the Key Vault VM extension.

You can use this feature to prevent other extensions from starting until all dependent certificates are installed. When the feature is enabled, the Key Vault VM extension will retry download and install of certificates up to 25 times with increasing backoff periods, during which it remains in a **Transitioning** state. If the retries are exhausted, the extension will report an **Error** state. After all certificates are successfully installed, the Key Vault VM extension reports a successful start.

To enable the extension dependency ordering feature in the Key Vault VM extension, set the `secretsManagementSettings` property:

JSON

```
"secretsManagementSettings": {
   "requireInitialSync": true,
   ...
}
```

For more information on how to set up dependencies between extensions, see Sequence extension provisioning in Virtual Machine Scale Sets.

> **Important**
>
> The extension dependency ordering feature isn't compatible with an ARM template that creates a system-assigned identity and updates a Key Vault access policy with that identity. If you attempt to use the feature in this scenario, a deadlock occurs because the Key Vault access policy can't update until after all extensions start. Instead, use a *single-user-assigned MSI identity* and pre-ACL your key vaults with that identity before you deploy.

Azure PowerShell deployment

The Azure Key Vault VM extension can be deployed with Azure PowerShell. Save Key Vault VM extension settings to a JSON file (settings.json).

The following JSON snippets provide example settings for deploying the Key Vault VM extension with PowerShell.

JSON

```
{
   "secretsManagementSettings": {
   "pollingIntervalInS": "3600",
   "linkOnRenewal": true,
   "observedCertificates":
   [
      {
          "url": "https://<examplekv>.vault.azure.net/secrets/certificate1",
          "certificateStoreName": "MY",
          "certificateStoreLocation": "LocalMachine",
          "accounts": [
             "Network Service"
          ]
      },
      {
          "url": "https://<examplekv>.vault.azure.net/secrets/certificate2",
          "certificateStoreName": "MY",
          "certificateStoreLocation": "LocalMachine",
          "keyExportable": true,
          "accounts": [
             "Network Service",
             "Local Service"
          ]
```

```
      }
   ]},
   "authenticationSettings": {
      "msiEndpoint":  "http://169.254.169.254/metadata/identity/oauth2/token",
      "msiClientId":  "00001111-aaaa-2222-bbbb-3333cccc4444"
   }
}
```

Deploy on a VM

PowerShell

```
# Build settings
$settings = (get-content -raw ".\settings.json")
$extName =  "KeyVaultForWindows"
$extPublisher = "Microsoft.Azure.KeyVault"
$extType = "KeyVaultForWindows"

# Start the deployment
Set-AzVmExtension -TypeHandlerVersion "3.0" -ResourceGroupName <ResourceGroupName> -Location
<Location> -VMName <VMName> -Name $extName -Publisher $extPublisher -Type $extType -
SettingString $settings
```

Deploy on a Virtual Machine Scale Sets instance

PowerShell

```
# Build settings
$settings = ".\settings.json"
$extName = "KeyVaultForWindows"
$extPublisher = "Microsoft.Azure.KeyVault"
$extType = "KeyVaultForWindows"

# Add extension to Virtual Machine Scale Sets
$vmss = Get-AzVmss -ResourceGroupName <ResourceGroupName> -VMScaleSetName <VmssName>
Add-AzVmssExtension -VirtualMachineScaleSet $vmss  -Name $extName -Publisher $extPublisher -
Type $extType -TypeHandlerVersion "3.0" -Setting $settings

# Start the deployment
Update-AzVmss -ResourceGroupName <ResourceGroupName> -VMScaleSetName <VmssName> -
VirtualMachineScaleSet $vmss
```

Azure CLI deployment

The Azure Key Vault VM extension can be deployed by using the Azure CLI. Save Key Vault VM extension settings to a JSON file (settings.json).

The following JSON snippets provide example settings for deploying the Key Vault VM extension with the Azure CLI.

JSON

```
  {
       "secretsManagementSettings": {
         "pollingIntervalInS": "3600",
         "linkOnRenewal": true,
         "observedCertificates": [
           {
               "url": "https://<examplekv>.vault.azure.net/secrets/certificate1",
               "certificateStoreName": "MY",
               "certificateStoreLocation": "LocalMachine",
               "accounts": [
                   "Network Service"
               ]
           },
           {
               "url": "https://<examplekv>.vault.azure.net/secrets/certificate2",
               "certificateStoreName": "MY",
               "certificateStoreLocation": "LocalMachine",
               "keyExportable": true,
               "accounts": [
                   "Network Service",
                   "Local Service"
               ]
           }
       ]
       },
         "authenticationSettings": {
         "msiEndpoint":  "http://169.254.169.254/metadata/identity/oauth2/token",
         "msiClientId":  "00001111-aaaa-2222-bbbb-3333cccc4444"
       }
    }
```

Deploy on a VM

Azure CLI

```
# Start the deployment
az vm extension set --name "KeyVaultForWindows" `
 --publisher Microsoft.Azure.KeyVault `
 --resource-group "<resourcegroup>" `
 --vm-name "<vmName>" `
 --settings "@settings.json"
```

Deploy on a Virtual Machine Scale Sets instance

```
# Start the deployment
az vmss extension set --name "KeyVaultForWindows" `
 --publisher Microsoft.Azure.KeyVault `
 --resource-group "<resourcegroup>" `
 --vmss-name "<vmssName>" `
 --settings "@settings.json"
```

Troubleshoot issues

Here are some suggestions for how to troubleshoot deployment issues.

Check frequently asked questions

Is there a limit on the number of observed certificates?

No. The Key Vault VM extension doesn't limit the number of observed certificates (`observedCertificates`).

What's the default permission when no account is specified?

By default, Administrators and SYSTEM receive Full Control.

How do you determine if a certificate key is CAPI1 or CNG?

The extension relies on the default behavior of the PFXImportCertStore API. By default, if a certificate has a Provider Name attribute that matches with CAPI1, then the certificate is imported by using CAPI1 APIs. Otherwise, the certificate is imported by using CNG APIs.

Does the extension support certificate auto-rebinding?

Yes, the Azure Key Vault VM extension supports certificate auto-rebinding. The Key Vault VM extension does support S-channel binding on certificate renewal when the `linkOnRenewal` property is set to true.

For IIS, you can configure auto-rebind by enabling automatic rebinding of certificate renewals in IIS. The Azure Key Vault VM extension generates Certificate Lifecycle Notifications when a certificate with a matching SAN is installed. IIS uses this event to auto-rebind the certificate. For more information, see Certifcate Rebind in IIS.

View extension status

Check the status of your extension deployment in the Azure portal, or by using PowerShell or the Azure CLI.

To see the deployment state of extensions for a given VM, run the following commands.

- Azure PowerShell:

  PowerShell

  ```
  Get-AzVMExtension -ResourceGroupName <myResourceGroup> -VMName <myVM> -Name
  <myExtensionName>
  ```

- The Azure CLI:

  Azure CLI

  ```
  az vm get-instance-view --resource-group <myResourceGroup> --name <myVM> --query "instanceView.extensions"
  ```

Review logs and configuration

The Key Vault VM extension logs exist only locally on the VM. Review the log details to help with troubleshooting.

Expand table

Log file	Description
C:\WindowsAzure\Logs\WaAppAgent.log`	Shows when updates occur to the extension.
C:\WindowsAzure\Logs\Plugins\Microsoft.Azure.KeyVault.KeyVaultForWindows<*most recent version*>\	Shows the status of certificate download. The download location is always the Windows computer's MY store (certlm.msc).
C:\Packages\Plugins\Microsoft.Azure.KeyVault.KeyVaultForWindows<*most recent version*>\RuntimeSettings\	The Key Vault VM Extension service logs show the status of the akvvm_service service.
C:\Packages\Plugins\Microsoft.Azure.KeyVault.KeyVaultForWindows<*most recent version*>\Status\	The configuration and binaries for the Key Vault VM Extension service.

Certificate Installation on Windows

The Key Vault VM extension for Windows installs certificates into the Windows certificate store. When a certificate is downloaded from Key Vault, the extension:

1. Installs all intermediate and leaf certificates, regardless of how many intermediate certificates are present. Root certificates are not installed, as the extension is not authorized to perform root installation. It is the responsibility of the service owner to ensure that the root certificate is trusted on the system.

   - Leaf certificates are installed in the specified certificate store (`certificateStoreName`) and location (`certificateStoreLocation`)
   - Intermediate CA certificates are installed in the Intermediate Certificate Authorities store

2. Places the certificates in the specified certificate store (`certificateStoreName`) and location (`certificateStoreLocation`)
3. Applies appropriate permissions to the private key based on the `accounts` specified in the configuration
4. Sets the `linkOnRenewal` property (if enabled) to ensure certificate bindings in applications like IIS automatically update when certificates are renewed

Default Certificate Stores

If not specified, certificates are installed in the following locations by default:

- Store Name: MY (Personal)
- Store Location: LocalMachine

Certificate Access Control

By default, Administrators and SYSTEM receive Full Control permissions on installed certificates. You can customize access using the `accounts` array in the certificate configuration:

JSON

```
"accounts": ["Network Service", "Local Service"]
```

This grants read access to the specified accounts, allowing applications running under those identities to use the certificates.

Certificate Renewal

When certificates are renewed in Key Vault, the extension automatically:

1. Downloads the new certificate version
2. Installs it in the configured certificate store
3. Maintains existing bindings through the `linkOnRenewal` feature if enabled

Managing Certificate Lifecycle

For applications like IIS that support Certificate Lifecycle Notifications, the extension generates events when certificates with matching Subject Alternative Names (SANs) are installed, allowing automatic rebinding without service interruption.

Get support

Here are some other options to help you resolve deployment issues:

- For assistance, contact the Azure experts on the Q&A and Stack Overflow forums.

- If you don't find an answer on the site, you can post a question for input from Microsoft or other members of the community.
- You can also Contact Microsoft Support. For information about using Azure support, read the Azure support FAQ.

Last updated on 11/10/2025

Azure Machine Configuration extension

09/04/2025

The Machine Configuration extension performs audit and configuration operations inside virtual machines (VMs).

To check policies inside VMs, such as Azure compute security baseline definitions for Linux and Windows, the Machine Configuration extension must be installed.

Prerequisites

To enable your VM to authenticate to the Machine Configuration service, your VM must have a system-assigned managed identity. You can satisfy the identity requirement for your VM by setting the `"type": "SystemAssigned"` property:

JSON

```
"identity": {
   "type": "SystemAssigned"
}
```

Operating systems

Operating system support for the Machine Configuration extension is the same as documented operating system support for the end-to-end solution.

Internet connectivity

The agent installed by the Machine Configuration extension must be able to reach content packages listed by guest configuration assignments, and report status to the Machine Configuration service. The VM can connect by using outbound HTTPS over TCP port 443, or a connection provided through private networking.

To learn more about private networking, see the following articles:

- Azure Machine Configuration, Communicate over Azure Private Link
- Use private endpoints for Azure Storage

Install the extension

You can install and deploy the Azure Machine Configuration extension directly from the Azure portal, Azure CLI or PowerShell.

Azure Portal

1. Open the Azure portal.
2. In the search box, enter **Virtual machines** and then select **Virtual machines** to display the list of available VMs.
3. Select the virtual machines you want to use.
4. In the search box of virtual machine page, enter **Extensions+applications** and then select it.
5. Click on Add in the extensions page.
6. In the search box of extension page, enter **Azure Machine Configuration extension for Windows** or **Azure Machine Configuration extension for Linux** based on the OS type and select it.
7. Click Next and then select **Review + create** to install the extension.
8. Once validation passes, select **Create**.
9. Once installation finishes, you can see **AzurePolicyforWindows** or **AzurePolicyforLinux** extension installed in the extension page.

Using Application Health extension with Azure Virtual Machines

Article • 09/12/2024

Monitoring your application health is an important signal for managing your VMs. Azure Virtual Machines provides support for Automatic VM Guest Patching, which rely on health monitoring of the individual instances to safely update your VMs.

This article describes how you can use the two types of Application Health extension, **Binary Health States** or **Rich Health States**, to monitor the health of your applications deployed on Azure virtual machines.

Application health monitoring is also available on virtual machine scale sets and helps enable functionalities such as Rolling Upgrades, Automatic OS-Image Upgrades, and Automatic Instance Repairs. To experience these capabilities with the added benefits of scale, availability, and flexibility on scale sets, you can attach your VM to an existing scale set or create a new scale set.

Prerequisites

This article assumes that you're familiar with Azure virtual machine extensions.

> **⊗ Caution**
>
> Application Health Extension expects to receive a consistent probe response at the configured port `tcp` or request path `http/https` in order to label a VM as *Healthy*. If no application is running on the VM, or you're unable to configure a probe response, your VM is going to show up as *Unhealthy* (Binary Health States) or *Unknown* (Rich Health States). See **application health samples** for examples of health probe responses being emitted to a local endpoint.

When to use the Application Health extension

Application Health Extension reports on application health from inside the Virtual Machine. The extension probes on a local application endpoint and updates the health status based on TCP/HTTP(S) responses received from the application. This health status is used by Azure to monitor and detect patching failures during Automatic VM Guest Patching.

The extension reports health from within a VM and can be used in situations where an external probe such as the Azure Load Balancer health probes can't be used.

Application health is a customer-provided signal on the status of your application running inside the VM. Application health is different from resource health, which is a platform-provided signal used to report service-level events impacting the performance of your VM.

Binary versus Rich Health States

Application Health Extensions has two options available: **Binary Health States** and **Rich Health States**. The following table highlights some key differences between the two options. See the end of this section for general recommendations.

Expand table

Features	Binary Health States	Rich Health States
Available Health States	Two available states: *Healthy*, *Unhealthy*	Four available states: *Healthy*, *Unhealthy*, *Initializing*, *Unknown*[1]
Sending Health Signals	Health signals are sent through HTTP/HTTPS response codes or TCP connections.	Health signals on HTTP/HTTPS protocol are sent through the probe response code and response body. Health signals through TCP protocol remain unchanged from Binary Health States.
Identifying *Unhealthy* Instances	Instances automatically fall into *Unhealthy* state if a *Healthy* signal isn't received from the application. An *Unhealthy* instance can indicate either an issue with the extension configuration (for example, unreachable endpoint) or an issue with the application (for example, non-200 status code).	Instances only go into an *Unhealthy* state if the application emits an *Unhealthy* probe response. Users are responsible for implementing custom logic to identify and flag instances with *Unhealthy* applications[2]. Instances with incorrect extension settings (for example, unreachable endpoint) or invalid health probe responses will fall under the *Unknown* state[2].
Initializing state for newly created instances	*Initializing* state isn't available. Newly created instances may take some time before settling into a steady state.	*Initializing* state allows newly created instances to settle into a steady Health State before surfacing the health state as *Healthy*, *Unhealthy*, or *Unknown*.
HTTP/HTTPS protocol	Supported	Supported

Features	Binary Health States	Rich Health States
TCP protocol	Supported	Limited Support – *Unknown* state is unavailable on TCP protocol. See Rich Health States protocol table for Health State behaviors on TCP.

[1] The *Unknown* state is unavailable on TCP protocol. [2] Only applicable for HTTP/HTTPS protocol. TCP protocol follows the same process of identifying *Unhealthy* instances as in Binary Health States.

Use **Binary Health States** if:

- You're not interested in configuring custom logic to identify and flag an unhealthy instance
- You don't require an *initializing* grace period for newly created instances

Use **Rich Health States** if:

- You send health signals through HTTP/HTTPS protocol and can submit health information through the probe response body
- You would like to use custom logic to identify and mark unhealthy instances
- You would like to set an *initializing* grace period allowing newly created instances to settle into a steady health state

Binary Health States

Binary Health State reporting contains two Health States, *Healthy* and *Unhealthy*. The following tables provide a brief description for how the Health States are configured.

HTTP/HTTPS Protocol

Expand table

Protocol	Health State	Description
http/https	Healthy	To send a *Healthy* signal, the application is expected to return a 200 response code.
http/https	Unhealthy	The instance is marked as *Unhealthy* if a 200 response code isn't received from the application.

TCP Protocol

Expand table

Protocol	Health State	Description
TCP	Healthy	To send a *Healthy* signal, a successful handshake must be made with the provided application endpoint.
TCP	Unhealthy	The instance is marked as *Unhealthy* if a failed or incomplete handshake occurred with the provided application endpoint.

Some common scenarios that result in an *Unhealthy* state include:

- When the application endpoint returns a non-200 status code
- When there's no application endpoint configured inside the virtual machine to provide application health status
- When the application endpoint is incorrectly configured
- When the application endpoint isn't reachable

Rich Health States

Rich Health States reporting contains four Health States, *Initializing*, *Healthy*, *Unhealthy*, and *Unknown*. The following tables provide a brief description for how each Health State is configured.

HTTP/HTTPS Protocol

Expand table

Protocol	Health State	Description
http/https	Healthy	To send a *Healthy* signal, the application is expected to return a probe response with: **Probe Response Code**: Status 2xx, **Probe Response Body**: `{"ApplicationHealthState": "Healthy"}`
http/https	Unhealthy	To send an *Unhealthy* signal, the application is expected to return a probe response with: **Probe Response Code**: Status 2xx, **Probe Response Body**: `{"ApplicationHealthState": "Unhealthy"}`
http/https	Initializing	The instance automatically enters an *Initializing* state at extension start time. For more information, see Initializing state.
http/https	Unknown	An *Unknown* state may occur in the following scenarios: when a non-2xx status code is returned by the application, when the probe request times out, when the application endpoint is unreachable or incorrectly configured, when a missing or invalid value is provided for

Protocol	Health State	Description
		`ApplicationHealthState` in the response body, or when the grace period expires. For more information, see Unknown state.

TCP Protocol

Expand table

Protocol	Health State	Description
TCP	Healthy	To send a *Healthy* signal, a successful handshake must be made with the provided application endpoint.
TCP	Unhealthy	The instance is marked as *Unhealthy* if a failed or incomplete handshake occurred with the provided application endpoint.
TCP	Initializing	The instance automatically enters an *Initializing* state at extension start time. For more information, see Initializing state.

Initializing state

This state only applies to Rich Health States. The *Initializing* state only occurs once at extension start time and can be configured by the extension settings `gracePeriod` and `numberOfProbes`.

At extension startup, the application health remains in the *Initializing* state until one of two scenarios occurs:

- The same Health State (*Healthy* or *Unhealthy*) is reported a consecutive number of times as configured through *numberOfProbes*
- The `gracePeriod` expires

If the same Health State (*Healthy* or *Unhealthy*) is reported consecutively, the application health transitions out of the *Initializing* state and into the reported Health State (*Healthy* or *Unhealthy*).

Example

If `numberOfProbes` = 3, that would mean:

- To transition from *Initializing* to *Healthy* state: Application health extension must receive three consecutive *Healthy* signals via HTTP/HTTPS or TCP protocol

- To transition from *Initializing* to *Unhealthy* state: Application health extension must receive three consecutive *Unhealthy* signals via HTTP/HTTPS or TCP protocol

If the `gracePeriod` expires before a consecutive health status is reported by the application, the instance health is determined as follows:

- HTTP/HTTPS protocol: The application health transitions from *Initializing* to *Unknown*
- TCP protocol: The application health transitions from *Initializing* to *Unhealthy*

Unknown state

The *Unknown* state only applies to Rich Health States. This state is only reported for `http` or `https` probes and occurs in the following scenarios:

- When a non-2xx status code is returned by the application
- When the probe request times out
- When the application endpoint is unreachable or incorrectly configured
- When a missing or invalid value is provided for `ApplicationHealthState` in the response body
- When the grace period expires

Extension schema for Binary Health States

The following JSON shows the schema for the Application Health extension. The extension requires at a minimum either a "tcp", "http" or "https" request with an associated port or request path respectively.

JSON

```
{
  "extensionProfile" : {
     "extensions" : [
      {
        "name": "HealthExtension",
        "properties": {
          "publisher": "Microsoft.ManagedServices",
          "type": "<ApplicationHealthLinux or ApplicationHealthWindows>",
          "autoUpgradeMinorVersion": true,
          "typeHandlerVersion": "1.0",
          "settings": {
            "protocol": "<protocol>",
            "port": <port>,
            "requestPath": "</requestPath>",
            "intervalInSeconds": 5,
```

```
            "numberOfProbes": 1
          }
        }
      }
    ]
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	`2018-10-01` or above	date
publisher	`Microsoft.ManagedServices`	string
type	`ApplicationHealthLinux` (Linux), `ApplicationHealthWindows` (Windows)	string
typeHandlerVersion	`1.0`	string

Settings

Expand table

Name	Value / Example	Data Type
protocol	`http` or `https` or `tcp`	string
port	Optional when protocol is `http` or `https`, mandatory when protocol is `tcp`	int
requestPath	Mandatory when protocol is `http` or `https`, not allowed when protocol is `tcp`	string
intervalInSeconds	Optional, default is 5 seconds. This setting is the interval between each health probe. For example, if intervalInSeconds == 5, a probe is sent to the local application endpoint once every 5 seconds.	int
numberOfProbes	Optional, default is 1. This setting is the number of consecutive probes required for the health status to change. For example, if numberOfProbles == 3, you will need 3 consecutive "Healthy" signals to change the health status from "Unhealthy" into "Healthy"	int

Name	Value / Example	Data Type
	state. The same requirement applies to change health status into "Unhealthy" state.	

Extension schema for Rich Health States

The following JSON shows the schema for the Rich Health States extension. The extension requires at a minimum either an "http" or "https" request with an associated port or request path respectively. TCP probes are also supported, but cannot set the `ApplicationHealthState` through the probe response body and do not have access to the *Unknown* state.

JSON

```
{
  "extensionProfile" : {
     "extensions" : [
      {
        "name": "HealthExtension",
        "properties": {
          "publisher": "Microsoft.ManagedServices",
          "type": "<ApplicationHealthLinux or ApplicationHealthWindows>",
          "autoUpgradeMinorVersion": true,
          "typeHandlerVersion": "2.0",
          "settings": {
            "protocol": "<protocol>",
            "port": <port>,
            "requestPath": "</requestPath>",
            "intervalInSeconds": 5,
            "numberOfProbes": 1,
            "gracePeriod": 600
          }
        }
      }
    ]
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	`2018-10-01` or above	date
publisher	`Microsoft.ManagedServices`	string
type	`ApplicationHealthLinux` (Linux), `ApplicationHealthWindows` (Windows)	string
typeHandlerVersion	`2.0`	string

Settings

Expand table

Name	Value / Example	Data Type
protocol	`http` or `https` or `tcp`	string
port	Optional when protocol is `http` or `https`, mandatory when protocol is `tcp`	int
requestPath	Mandatory when protocol is `http` or `https`, not allowed when protocol is `tcp`	string
intervalInSeconds	Optional, default is 5 seconds. This setting is the interval between each health probe. For example, if intervalInSeconds == 5, a probe is sent to the local application endpoint once every 5 seconds.	int
numberOfProbes	Optional, default is 1. This setting is the number of consecutive probes required for the health status to change. For example, if numberOfProbles == 3, you will need 3 consecutive "Healthy" signals to change the health status from "Unhealthy"/"Unknown" into "Healthy" state. The same requirement applies to change health status into "Unhealthy" or "Unknown" state.	int
gracePeriod	Optional, default = `intervalInSeconds` * `numberOfProbes`; maximum grace period is 7200 seconds	int

Deploy the Application Health extension

There are multiple ways of deploying the Application Health extension to your VMs as detailed in the following examples.

Binary Health States

REST API

The following example adds the Application Health extension named *myHealthExtension* to a Windows-based virtual machine.

You can also use this example to change an existing extension from Rich Health States to Binary Health by making a PATCH call instead of a PUT.

```
PUT on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers
/Microsoft.Compute/virtualMachines/myVM/extensions/myHealthExtension?
api-version=2018-10-01`
```

JSON

```
{
  "name": "myHealthExtension",
  "location": "<location>",
  "properties": {
    "publisher": "Microsoft.ManagedServices",
    "type": "ApplicationHealthWindows",
    "autoUpgradeMinorVersion": true,
    "typeHandlerVersion": "1.0",
    "settings": {
      "protocol": "<protocol>",
      "port": <port>,
      "requestPath": "</requestPath>"
    }
  }
}
```

Use `PATCH` to edit an already deployed extension.

Rich Health States

REST API

The following example adds the **Application Health - Rich States** extension (with name myHealthExtension) to a Windows-based virtual machine.

You can also use this example to upgrade an existing extension from Binary to Rich Health States by making a PATCH call instead of a PUT.

```
PUT on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers
/Microsoft.Compute/virtualMachines/myVM/extensions/myHealthExtension?
api-version=2018-10-01`
```

JSON

```
{
  "name": "myHealthExtension",
  "location": "<location>",
  "properties": {
    "publisher": "Microsoft.ManagedServices",
    "type": "ApplicationHealthWindows",
    "autoUpgradeMinorVersion": true,
    "typeHandlerVersion": "2.0",
    "settings": {
      "requestPath": "</requestPath>",
      "intervalInSeconds": <intervalInSeconds>,
      "numberOfProbes": <numberOfProbes>,
      "gracePeriod": <gracePeriod>
    }
  }
}
```

Use `PATCH` to edit an already deployed extension.

Troubleshoot

Need help configuring a probe response

See application health samples for examples of health probe responses being emitted to a local endpoint.

View VMHealth

REST API

```
GET https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMa
chines/myVM/instanceView?api-version=2023-07-01
```

Sample Response (see "vmHealth" object for the latest VM health status)

```
"vmHealth": {
    "status": {
      "code": "HealthState/unknown",
      "level": "Warning",
      "displayStatus": "The VM health is unknown",
      "time": "2023-12-04T22:25:39+00:00"
    }
}
```

Extension execution output log

Extension execution output is logged to files found in the following directories:

Windows

```
C:\WindowsAzure\Logs\Plugins\Microsoft.ManagedServices.ApplicationHealthWind
ows\<version>\
```

Linux

```
/var/lib/waagent/Microsoft.ManagedServices.ApplicationHealthLinux-
<extension_version>/status
/var/log/azure/applicationhealth-extension
```

The logs also periodically capture the application health status.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use the Azure Custom Script Extension Version 2 with Linux virtual machines

08/18/2025

The Custom Script Extension Version 2 downloads and runs scripts on Azure virtual machines (VMs). Use this extension for post-deployment configuration, software installation, or any other configuration or management task. You can download scripts from Azure Storage or another accessible internet location, or you can provide them to the extension runtime.

The Custom Script Extension integrates with Azure Resource Manager templates. You can also run it by using the Azure CLI, Azure PowerShell, or the Azure Virtual Machines REST API.

This article describes how to use the Custom Script Extension from the Azure CLI, and how to run the extension by using an Azure Resource Manager template. This article also provides troubleshooting steps for Linux systems.

There are two versions of the Custom Script Extension:

- Version 1: Microsoft.OSTCExtensions.CustomScriptForLinux
- Version 2: Microsoft.Azure.Extensions.CustomScript

Use Version 2 for new and existing deployments. The new version is a drop-in replacement. The migration is as easy as changing the name and version. You don't need to change your extension configuration.

Prerequisites

Supported Linux distributions

Expand table

Publisher	Distribution	x64	ARM64
Alma Linux Community	Alma Linux	8.x+, 9.x+	8.x+, 9.x+
Credativ	Debian	10+	11.x+
Kinvolk	Flatcar Linux	3374.2.x+	3374.2.x+
Microsoft	Azure Linux	2.x	2.x
openSUSE Project	openSUSE	12.3+	*Not supported*
Oracle	Oracle Linux	6.4+, 7.x+, 8.x+	*Not supported*
Red Hat	Red Hat Enterprise Linux	6.7+, 7.x+, 8.x+, 9.x+, 10.x+	8.6+, 9.0+, 10.x+
CIQ	Rocky Linux	9.x+	9.x+
SUSE	SLES	12.x+, 15.x+	15.x SP4+
Canonical	Ubuntu (LTS releases)	18.04+, 20.04+, 22.04+, 24.04+	20.04+, 22.04+, 24.04+

Script location

You can set the extension to use your Azure Blob Storage credentials so that it can access Azure Blob Storage. The script location can be anywhere, as long as the VM can route to that endpoint, for example, GitHub or an internal file server.

Internet connectivity

To download a script externally, such as from GitHub or Azure Storage, you need to open other firewall or network security group (NSG) ports. For example, if your script is located in Azure Storage, you can allow access by using Azure NSG service tags for Storage.

If your script is on a local server, you might still need to open other firewall or NSG ports.

Tips

- The highest failure rate for this extension is due to syntax errors in the script. Verify that the script runs without errors. Put more logging into the script to make it easier to find failures.
- Write scripts that are idempotent, so that running them more than once accidentally doesn't cause system changes.
- Ensure that the scripts don't require user input when they run.
- The script is allowed 90 minutes to run. Anything longer results in a failed provision of the extension.
- Don't put reboots inside the script. Restarting causes problems with other extensions that are being installed, and the extension doesn't continue after the reboot.
- If you have a script that causes a reboot before installing applications and running scripts, schedule the reboot by using a cron job or by using tools such as DSC, Chef, or Puppet extensions.
- Don't run a script that causes a stop or update of the Azure Linux Agent. It might leave the extension in a transitioning state and lead to a time-out.
- The extension runs a script only once. If you want to run a script on every startup, you can use a cloud-init image and use a Scripts Per Boot module. Alternatively, you can use the script to create a systemd service unit.
- You can have only one version of an extension applied to the VM. To run a second custom script, update the existing extension with a new configuration. Alternatively, you can remove the Custom Script Extension and reapply it with the updated script.
- If you want to schedule when a script runs, use the extension to create a cron job.
- When the script is running, you only see a *transitioning* extension status from the Azure portal or CLI. If you want more frequent status updates for a running script, create your own solution.
- The Custom Script Extension doesn't natively support proxy servers. However, you can use a file transfer tool, such as `Curl`, that supports proxy servers within your script.
- Be aware of nondefault directory locations that your scripts or commands might rely on. Have logic to handle this situation.

Extension schema

The Custom Script Extension configuration specifies things like script location and the command to be run. You can store this information in configuration files, specify it on the command line, or specify it in an Azure Resource Manager template.

You can store sensitive data in a protected configuration, which is encrypted and only decrypted on the target VM. The protected configuration is useful when the execution command includes secrets such as a password. Here's an example:

JSON

```
{
  "name": "config-app",
  "type": "Extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2019-03-01",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', concat(variables('vmName'),copyindex()))]"
  ],
  "tags": {
    "displayName": "config-app"
  },
  "properties": {
    "publisher": "Microsoft.Azure.Extensions",
    "type": "CustomScript",
    "typeHandlerVersion": "2.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "skipDos2Unix":false,
      "timestamp":123456789
    },
    "protectedSettings": {
       "commandToExecute": "<command-to-execute>",
       "script": "<base64-script-to-execute>",
       "storageAccountName": "<storage-account-name>",
       "storageAccountKey": "<storage-account-key>",
       "fileUris": ["https://.."],
       "managedIdentity" : "<managed-identity-identifier>"
    }
  }
}
```

> **Note**
>
> The `managedIdentity` property *must not* be used in conjunction with the `storageAccountName` or `storageAccountKey` property.

Property values

Expand table

Name	Value or example	Data type
apiVersion	`2019-03-01`	date
publisher	`Microsoft.Azure.Extensions`	string
type	`CustomScript`	string
typeHandlerVersion	`2.1`	int

Name	Value or example	Data type
fileUris	`https://github.com/MyProject/Archive/MyPythonScript.py`	array
commandToExecute	`python MyPythonScript.py \<my-param1>`	string
script	`IyEvYmluL3NoCmVjaG8gIlVwZGF0aW5nIHBhY2thZ2VzIC4uLiIKYXB0IHVwZGF0ZQphcHQgdXBncmFkZSAteQo=`	string
skipDos2Unix	`false`	boolean
timestamp	`123456789`	32-bit integer
storageAccountName	`examplestorageacct`	string
storageAccountKey	`TmJK/1N3AbAZ3q/+hOXoi/l73zOqsaxXDhqa9Y83/v5UpXQp2DQIBuv2Tifp60cE/OaHsJZmQZ7teQfczQj8hg==`	string
managedIdentity	`{ }` or `{ "clientId": "00001111-aaaa-2222-bbbb-3333cccc4444" }` or `{ "objectId": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb" }`	JSON object

Property value details

Expand table

Property	Optional or required	Details
apiVersion	Not applicable	You can find the most up-to-date API version by using Resource Explorer or by using the command `az provider list -o json` in the Azure CLI.
fileUris	Optional	URLs for files to be downloaded.
commandToExecute	Required if `script` isn't set	The entry point script to run. Use this property instead of `script` if your command contains secrets such as passwords.
script	Required if `commandToExecute` isn't set	A Base64-encoded and optionally gzip'ed script run by `/bin/sh`.
skipDos2Unix	Optional	Set this value to `false` if you want to skip dos2unix conversion of script-based file URLs or scripts.
timestamp	Optional	Change this value only to trigger a rerun of the script. Any integer value is acceptable, as long as it's different from the previous value.
storageAccountName	Optional	The name of storage account. If you specify storage credentials, all `fileUris` values must be URLs for Azure blobs.
storageAccountKey	Optional	The access key of the storage account.
managedIdentity	Optional	The managed identity for downloading files. Values are `clientId` (optional, string), which is the client ID of the managed identity, and `objectId` (optional, string), which is the object ID of the managed identity.

Public settings are sent in clear text to the VM where the script runs. *Protected settings* are encrypted through a key known only to Azure and the VM. The settings are saved to the VM as they were sent. That is, if the settings

were encrypted, they're saved encrypted on the VM. The certificate that's used to decrypt the encrypted values is stored on the VM. The certificate is also used to decrypt settings, if necessary, at runtime.

Using public settings might be useful for debugging, but we strongly recommend that you use protected settings.

You can set the following values in either public or protected settings. The extension rejects any configuration where these values are set in both public and protected settings.

- `commandToExecute`
- `script`
- `fileUris`

Property: skipDos2Unix

The previous version of the Custom Script Extension, `Microsoft.OSTCExtensions.CustomScriptForLinux`, automatically converts DOS files to UNIX files by translating `\r\n` to `\n`. This translation still exists and is on by default. This conversion is applied to all files downloaded from `fileUris` or the script setting based on either of the following criteria:

- The extension is *.sh*, *.txt*, *.py*, or *.pl*. The script setting always matches this criterion because it's assumed to be a script run with */bin/sh*. The script setting is saved as *script.sh* on the VM.
- The file starts with `#!`.

The default value is `false`, which means dos2unix conversion *is* executed. You can skip the dos2unix conversion by setting `skipDos2Unix` to `true`:

JSON

```
{
  "fileUris": ["<url>"],
  "commandToExecute": "<command-to-execute>",
  "skipDos2Unix": true
}
```

Property: script

The Custom Script Extension supports execution of a user-defined script. The script settings combine `commandToExecute` and `fileUris` into a single setting. Instead of having to set up a file for download from Azure Storage or a GitHub gist, you can encode the script as a setting. You can use the script to replace `commandToExecute` and `fileUris`.

Here are some requirements:

- The script must be Base64 encoded.
- The script can optionally be gzip'ed.
- You can use the script setting in public or protected settings.
- The maximum size of the script parameter's data is 256 KB. If the script exceeds this size, it doesn't run.

For example, the following script is saved to the file */script.sh/*:

```sh
#!/bin/sh
echo "Creating directories ..."
mkdir /data
chown user:user /data
mkdir /appdata
chown user:user /appdata
```

You would construct the correct Custom Script Extension script setting by taking the output of the following command:

```Bash
cat script.sh | base64 -w0
```

```JSON
{
  "script":
"IyEvYmluL3NoCmVjaG8gIlVwZGF0aW5nIHBhY2thZ2VzIC4uLiIKYXB0IHVwZGF0ZQphcHQgdXBncmFkZSAteQo="
}
```

In most cases, the script can optionally be gzip'ed to further reduce size. The Custom Script Extension automatically detects the use of gzip compression.

```Bash
cat script | gzip -9 | base64 -w 0
```

The Custom Script Extension uses the following algorithm to run a script:

1. Assert that the length of the script's value doesn't exceed 256 KB.
2. Base64 decode the script's value.
3. *Try* to gunzip the Base64-decoded value.
4. Write the decoded and optionally decompressed value to disk: */var/lib/waagent/custom-script/#/script.sh*.
5. Run the script by using `_/bin/sh -c /var/lib/waagent/custom-script/#/script.sh`.

Property: managedIdentity

> **Note**
>
> This property *must* be specified in protected settings only.

The Custom Script Extension, version 2.1 and later, supports managed identities for downloading files from URLs provided in the `fileUris` setting. This approach allows the Custom Script Extension to access Azure Storage private blobs or containers without the user having to pass secrets like shared access signature (SAS) tokens or storage account keys.

To use this feature, add a system-assigned or user-assigned identity to the VM or Virtual Machine Scale Set where the Custom Script Extension is expected to run. Then grant the managed identity access to the Azure Storage container or blob.

To use the system-assigned identity on the target VM or Virtual Machine Scale Set, set `managedidentity` to an empty JSON object.

JSON

```
{
  "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.sh"],
  "commandToExecute": "sh script1.sh",
  "managedIdentity" : {}
}
```

To use the user-assigned identity on the target VM or Virtual Machine Scale Set, configure `managedidentity` with the client ID or the object ID of the managed identity.

JSON

```
{
  "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.sh"],
  "commandToExecute": "sh script1.sh",
  "managedIdentity" : { "clientId": "00001111-aaaa-2222-bbbb-3333cccc4444" }
}
```

JSON

```
{
  "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.sh"],
  "commandToExecute": "sh script1.sh",
  "managedIdentity" : { "objectId": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb" }
}
```

> **Note**
>
> The `managedIdentity` property *must not* be used in conjunction with the `storageAccountName` or `storageAccountKey` property.

Template deployment

You can deploy Azure VM extensions by using Azure Resource Manager templates. The JSON schema detailed in the previous section can be used in an Azure Resource Manager template to run the Custom Script Extension during the template's deployment. You can find a sample template that includes the Custom Script Extension on GitHub.

JSON

```
{
  "name": "config-app",
  "type": "extensions",
```

```
  "location": "[resourceGroup().location]",
  "apiVersion": "2019-03-01",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', concat(variables('vmName'),copyindex()))]"
  ],
  "tags": {
    "displayName": "config-app"
  },
  "properties": {
    "publisher": "Microsoft.Azure.Extensions",
    "type": "CustomScript",
    "typeHandlerVersion": "2.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
      },
    "protectedSettings": {
      "commandToExecute": "sh hello.sh <param2>",
      "fileUris": ["https://github.com/MyProject/Archive/hello.sh"
      ]
    }
  }
}
```

> **Note**
>
> These property names are case-sensitive. To avoid deployment problems, use the names as shown here.

Azure CLI

When you use the Azure CLI to run the Custom Script Extension, create a configuration file or files. At a minimum, the configuration file must contain `commandToExecute`. The `az vm extension set` command refers to the configuration file:

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM --name customScript \
  --publisher Microsoft.Azure.Extensions \
  --protected-settings ./script-config.json
```

Alternatively, you can specify the settings in the command as a JSON-formatted string. This approach allows the configuration to be specified during execution and without a separate configuration file.

Azure CLI

```
az vm extension set \
  --resource-group exttest \
  --vm-name exttest \
  --name customScript \
  --publisher Microsoft.Azure.Extensions \
  --protected-settings '{"fileUris": ["https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-linux/scripts/config-music.sh"],"commandToExecute": "./config-music.sh"}'
```

Example: Public configuration with script file

This example uses the following script file named *script-config.json*:

JSON

```
{
  "fileUris": ["https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-linux/scripts/config-music.sh"],
  "commandToExecute": "./config-music.sh"
}
```

1. Create the script file by using the text editor of your choice or by using the following CLI command:

   Azure CLI

   ```
   cat <<EOF > script-config.json
   {
     "fileUris": ["https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-linux/scripts/config-music.sh"],
     "commandToExecute": "./config-music.sh"
   }
   EOF
   ```

2. Run the following command:

   Azure CLI

   ```
   az vm extension set \
     --resource-group myResourceGroup \
     --vm-name myVM --name customScript \
     --publisher Microsoft.Azure.Extensions \
     --settings ./script-config.json
   ```

Example: Public configuration with no script file

This example uses the following JSON-formatted content:

JSON

```
{
  "commandToExecute": "apt-get -y update && apt-get install -y apache2"
}
```

Run the following command:

Azure CLI

```
az vm extension set \
  --resource-group tim0329vmRG \
  --vm-name tim0329vm --name customScript \
  --publisher Microsoft.Azure.Extensions \
  --settings '{"commandToExecute": "apt-get -y update && apt-get install -y apache2"}'
```

Example: Public and protected configuration files

Use a public configuration file to specify the script file's URI:

JSON

```
{
  "fileUris": ["https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-linux/scripts/config-music.sh"]
}
```

Use a protected configuration file to specify the command to be run:

JSON

```
{
  "commandToExecute": "./config-music.sh"
}
```

1. Create the public configuration file by using the text editor of your choice or by using the following CLI command:

   Azure CLI

   ```
   cat <<EOF > script-config.json
   {
     "fileUris": ["https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-linux/scripts/config-music.sh"]
   }
   EOF
   ```

2. Create the protected configuration file by using the text editor of your choice or by using the following CLI command:

   Azure CLI

   ```
   cat <<EOF > protected-config.json
   {
     "commandToExecute": "./config-music.sh"
   }
   EOF
   ```

3. Run the following command:

   Azure CLI

   ```
   az vm extension set \
     --resource-group myResourceGroup \
     --vm-name myVM \
     --name customScript \
     --publisher Microsoft.Azure.Extensions \
     --settings ./script-config.json \
     --protected-settings ./protected-config.json
   ```

Virtual Machine Scale Sets

If you deploy the Custom Script Extension from the Azure portal, you don't have control over the expiration of the SAS token to access the script in your storage account. The initial deployment works, but when the storage account's SAS token expires, any subsequent scaling operation fails because the Custom Script Extension can no longer access the storage account.

We recommend that you use PowerShell, the Azure CLI, or an Azure Resource Manager template when you deploy the Custom Script Extension on a Virtual Machine Scale Set. This way, you can choose to use a managed identity or have direct control of the expiration of the SAS token for accessing the script in your storage account for as long as you need.

Troubleshooting

When the Custom Script Extension runs, the script is created or downloaded into a directory that's similar to the following example. The command output is also saved into this directory in `stdout` and `stderr` files.

Bash

```
sudo ls -l /var/lib/waagent/custom-script/download/0/
```

To troubleshoot, first check the Linux Agent log and ensure that the extension ran:

Bash

```
sudo cat /var/log/waagent.log
```

Look for the extension execution. It looks something like:

Output

```
2018/04/26 17:47:22.110231 INFO [Microsoft.Azure.Extensions.customScript-2.0.6] [Enable] current
handler state is: notinstalled
2018/04/26 17:47:22.306407 INFO Event: name=Microsoft.Azure.Extensions.customScript, op=Download,
message=Download succeeded, duration=167
2018/04/26 17:47:22.339958 INFO [Microsoft.Azure.Extensions.customScript-2.0.6] Initialize
extension directory
2018/04/26 17:47:22.368293 INFO [Microsoft.Azure.Extensions.customScript-2.0.6] Update settings
file: 0.settings
2018/04/26 17:47:22.394482 INFO [Microsoft.Azure.Extensions.customScript-2.0.6] Install extension
[bin/custom-script-shim install]
2018/04/26 17:47:23.432774 INFO Event: name=Microsoft.Azure.Extensions.customScript, op=Install,
message=Launch command succeeded: bin/custom-script-shim install, duration=1007
2018/04/26 17:47:23.476151 INFO [Microsoft.Azure.Extensions.customScript-2.0.6] Enable extension
[bin/custom-script-shim enable]
2018/04/26 17:47:24.516444 INFO Event: name=Microsoft.Azure.Extensions.customScript, op=Enable,
message=Launch command succeeded: bin/custom-sc
```

In the preceding output:

- `Enable` is when the command starts running.

- `Download` relates to the downloading of the Custom Script Extension package from Azure, not the script files specified in `fileUris`.

The Azure Script Extension produces a log, which you can find here:

Bash

```
sudo cat /var/log/azure/custom-script/handler.log
```

Look for the individual execution. It looks something like:

Output

```
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event=start
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event=pre-check
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="comparing
seqnum" path=mrseq
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="seqnum
saved" path=mrseq
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="reading
configuration"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="read
configuration"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0
event="validating json schema"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="json
schema valid"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="parsing
configuration json"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="parsed
configuration json"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0
event="validating configuration logically"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="validated
configuration"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="creating
output directory" path=/var/lib/waagent/custom-script/download/0
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="created
output directory"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 files=1
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 file=0
event="download start"
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 file=0
event="download complete" output=/var/lib/waagent/custom-script/download/0
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="executing
command" output=/var/lib/waagent/custom-script/download/0
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="executing
protected commandToExecute" output=/var/lib/waagent/custom-script/download/0
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event="executed
command" output=/var/lib/waagent/custom-script/download/0
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event=enabled
time=2018-04-26T17:47:23Z version=v2.0.6/git@1008306-clean operation=enable seq=0 event=end
```

Here you can see:

- The `enable` command that starts this log.
- The settings passed to the extension.
- The extension downloading the file and the result of that action.

- The command being run and the result.

You can also retrieve the execution state of the Custom Script Extension, including the actual arguments passed as `commandToExecute`, by using the Azure CLI:

Azure CLI

```
az vm extension list -g myResourceGroup --vm-name myVM
```

The output looks like the following text:

Output

```
[
  {
    "autoUpgradeMinorVersion": true,
    "forceUpdateTag": null,
    "id":
"/subscriptions/subscriptionid/resourceGroups/rgname/providers/Microsoft.Compute/virtualMachines/v
mname/extensions/customscript",
    "resourceGroup": "rgname",
    "settings": {
      "commandToExecute": "sh script.sh > ",
      "fileUris": [
        "https://catalogartifact.azureedge.net/publicartifacts/scripts/script.sh",
        "https://catalogartifact.azureedge.net/publicartifacts/scripts/script.sh"
      ]
    },
    "tags": null,
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "typeHandlerVersion": "2.0",
    "virtualMachineExtensionType": "CustomScript"
  },
  {
    "autoUpgradeMinorVersion": true,
    "forceUpdateTag": null,
    "id":
"/subscriptions/subscriptionid/resourceGroups/rgname/providers/Microsoft.Compute/virtualMachines/v
mname/extensions/OmsAgentForLinux",
    "instanceView": null,
    "location": "eastus",
    "name": "OmsAgentForLinux",
    "protectedSettings": null,
    "provisioningState": "Succeeded",
    "publisher": "Microsoft.EnterpriseCloud.Monitoring",
    "resourceGroup": "rgname",
    "settings": {
      "workspaceId": "workspaceid"
    },
    "tags": null,
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "typeHandlerVersion": "1.0",
    "virtualMachineExtensionType": "OmsAgentForLinux"
  }
]
```

Azure CLI syntax issues

The Azure CLI can run in several shell environments, but with slight format variations. If you have unexpected results with Azure CLI commands, see How to use the Azure CLI successfully.

Next steps

To see the code, current issues, and versions, see custom-script-extension-linux.

Custom Script Extension for Windows

08/18/2025

The Custom Script Extension downloads and runs scripts on Azure virtual machines (VMs). Use this extension for post-deployment configuration, software installation, or any other configuration or management task. You can download scripts from Azure Storage or GitHub, or provide them to the Azure portal at extension runtime.

The Custom Script Extension integrates with Azure Resource Manager templates. You can also run it by using the Azure CLI, Azure PowerShell, the Azure portal, or the Azure Virtual Machines REST API.

This article describes how to use the Custom Script Extension by using the Azure PowerShell module and Azure Resource Manager templates. It also provides troubleshooting steps for Windows systems.

Prerequisites

> **Note**
>
> Don't use the Custom Script Extension to run `Update-AzVM` with the same VM as its parameter. The extension will wait for itself.

Supported Windows operating systems

Expand table

OS Version	x64	ARM64
Windows 10	Supported	Supported
Windows 11	Supported	Supported
Windows Server 2016	Supported	Supported
Windows Server 2016 Core	Supported	Supported
Windows Server 2019	Supported	Supported
Windows Server 2019 Core	Supported	Supported
Windows Server 2022	Supported	Supported
Windows Server 2022 Core	Supported	Supported
Windows Server 2025	Supported	Supported
Windows Server 2025 Core	Supported	Supported

Script location

You can set the extension to use your Azure Blob Storage credentials so that it can access Azure Blob Storage. The script location can be anywhere, as long as the VM can route to that endpoint, for example, GitHub or an internal file server.

Internet connectivity

To download a script externally, such as from GitHub or Azure Storage, you need to open other firewall or network security group (NSG) ports. For example, if your script is located in Azure Storage, you can allow access by using Azure NSG service tags for Storage.

The Custom Script Extension doesn't have any way to bypass certificate validation. If you're downloading from a secured location with, for example, a self-signed certificate, you might get errors like *The remote certificate is invalid according to the validation procedure*. Make sure that the certificate is correctly installed in the *Trusted Root Certification Authorities* store on the VM.

If your script is on a local server, you might still need to open other firewall or NSG ports.

Tips

- Output is limited to the last 4,096 bytes.

- Properly escaping characters will help ensure that strings are parsed correctly. For example, you always need two backslashes to escape a single literal backslash when dealing with file paths. Sample: `{"commandToExecute": "C:\\Windows\\System32\\systeminfo.exe >> D:\\test.txt"}`
- The highest failure rate for this extension is due to syntax errors in the script. Verify that the script runs without errors. Put more logging into the script to make it easier to find failures.
- Write scripts that are idempotent, so that running them more than once accidentally doesn't cause system changes.
- Ensure that the scripts don't require user input when they run.
- The script is allowed 90 minutes to run. Anything longer results in a failed provision of the extension.
- Don't put restarts inside the script. This action causes problems with other extensions that are being installed, and the extension doesn't continue after the restart.
- If you have a script that causes a restart before installing applications and running scripts, schedule the restart by using a Windows Scheduled Task or by using tools such as DSC, Chef, or Puppet extensions.
- Don't run a script that causes a stop or update of the VM agent. It might leave the extension in a transitioning state and lead to a time-out.
- The extension runs a script only once. If you want to run a script on every startup, use the extension to create a Windows Scheduled Task.
- If you want to schedule when a script runs, use the extension to create a Windows Scheduled Task.
- When the script is running, you only see a *transitioning* extension status from the Azure portal or Azure CLI. If you want more frequent status updates for a running script, create your own solution.
- The Custom Script Extension doesn't natively support proxy servers. However, you can use a file transfer tool, such as Invoke-WebRequest, that supports proxy servers within your script.
- Be aware of nondefault directory locations that your scripts or commands might rely on. Have logic to handle this situation.
- Ensure you don't have any custom setting in the registry key `HKLM\SOFTWARE\Microsoft\Command Processor\AutoRun` (*detailed here*). This would trigger during the Custom Script Extension install or enable phases and cause an error like `'XYZ is not recognized as an internal or external command, operable program or batch file'`.
- The Custom Script Extension runs under the `LocalSystem` account.
- If you plan to use the `storageAccountName` and `storageAccountKey` properties, these properties must be collocated in `protectedSettings`.
- You can have only one version of an extension applied to the VM. To run a second custom script, you can update the existing extension with a new configuration. Alternatively, you can remove the custom script extension and reapply it with the updated script

Extension schema

The Custom Script Extension configuration specifies things like script location and the command to be run. You can store this configuration in configuration files, specify it on the command line, or specify it in an Azure Resource Manager template.

You can store sensitive data in a protected configuration, which is encrypted and only decrypted inside the VM. The protected configuration is useful when the execution command includes secrets such as a password or a shared access signature (SAS) file reference. Here's an example:

JSON

```
{
    "apiVersion": "2018-06-01",
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "name": "virtualMachineName/config-app",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'),copyindex())]",
        "[variables('musicstoresqlName')]"
    ],
    "tags": {
        "displayName": "config-app"
    },
    "properties": {
        "publisher": "Microsoft.Compute",
        "type": "CustomScriptExtension",
        "typeHandlerVersion": "1.10",
        "autoUpgradeMinorVersion": true,
        "settings": {
            "timestamp":123456789
        },
        "protectedSettings": {
            "commandToExecute": "myExecutionCommand",
            "storageAccountName": "myStorageAccountName",
            "storageAccountKey": "myStorageAccountKey",
            "managedIdentity" : {},
            "fileUris": [
                "script location"
            ]
        }
```

```
    }
}
```

> **Note**
>
> The `managedIdentity` property *must not* be used in conjunction with the `storageAccountName` or `storageAccountKey` property.

Only one version of an extension can be installed on a VM at a time. Specifying a custom script twice in the same Azure Resource Manager template for the same VM fails.

You can use this schema inside the VM resource or as a standalone resource. If this extension is used as a standalone resource in the Azure Resource Manager template, the name of the resource has to be in the format *virtualMachineName/extensionName*.

Property values

Expand table

Name	Value or example	Data type
apiVersion	`2018-06-01`	date
publisher	`Microsoft.Compute`	string
type	`CustomScriptExtension`	string
typeHandlerVersion	`1.10`	int
fileUris	`https://raw.githubusercontent.com/Microsoft/dotnet-core-sample-templates/master/dotnet-core-music-windows/scripts/configure-music-app.ps1`	array
timestamp	`123456789`	32-bit integer
commandToExecute	`powershell -ExecutionPolicy Unrestricted -File configure-music-app.ps1`	string
storageAccountName	`examplestorageacct`	string
storageAccountKey	`TmJK/1N3AbAZ3q/+hOXoi/l73zOqsaxXDhqa9Y83/v5UpXQp2DQIBuv2Tifp60cE/OaHsJZmQZ7teQfczQj8hg==`	string
managedIdentity	`{ }` or `{ "clientId": "00001111-aaaa-2222-bbbb-3333cccc4444" }` or `{ "objectId": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb" }`	JSON object

> **Note**
>
> These property names are case-sensitive. To avoid deployment problems, use the names as shown here.

Property value details

Expand table

Property	Optional or required	Details
fileUris	Optional	URLs for files to be downloaded. If URLs are sensitive, for example, if they contain keys, this field should be specified in `protectedSettings`.
commandToExecute	Required	The entry point script to run. Use this property if your command contains secrets such as passwords or if your file URIs are sensitive.
timestamp	Optional	Change this value only to trigger a rerun of the script. Any integer value is acceptable, as long as it's different from the previous value.
storageAccountName	Optional	The name of storage account. If you specify storage credentials, all `fileUris` values must be URLs for Azure blobs.
storageAccountKey	Optional	The access key of the storage account.
managedIdentity	Optional	The managed identity for downloading files. Valid values are `clientId` (optional, string), which is the client ID of the managed identity, and `objectId` (optional, string), which is the object ID of the managed identity.

Public settings are sent in clear text to the VM where the script runs. *Protected settings* are encrypted through a key known only to Azure and the VM. The settings are saved to the VM as they were sent. That is, if the settings were encrypted, they're saved encrypted on the VM. The certificate that's used to decrypt the encrypted values is stored on the VM. The certificate is also used to decrypt settings, if necessary, at runtime.

Using public settings might be useful for debugging, but we recommend that you use protected settings.

You can set the following values in either public or protected settings. The extension rejects any configuration where these values are set in both public and protected settings.

- `commandToExecute`
- `fileUris`

Property: managedIdentity

> ⓘ Note
>
> This property *must* be specified in protected settings only.

The Custom Script Extension, version 1.10 and later, supports managed identities for downloading files from URLs provided in the `fileUris` setting. The property allows the Custom Script Extension to access Azure Storage private blobs or containers without the user having to pass secrets like SAS tokens or storage account keys.

> ⓘ Note
>
> The Custom Script Extension does not currently support using Managed Identity authentication on Azure Arc–enabled servers.

To use this feature, add a system-assigned or user-assigned identity to the VM or Virtual Machine Scale Set where the Custom Script Extension runs. Then grant the managed identity access to the Azure Storage container or blob.

To use the system-assigned identity on the target VM or Virtual Machine Scale Set, set `managedidentity` to an empty JSON object.

JSON

```
{
  "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.ps1"],
  "commandToExecute": "powershell.exe script1.ps1",
  "managedIdentity" : {}
}
```

To use the user-assigned identity on the target VM or Virtual Machine Scale Set, configure `managedidentity` with the client ID or the object ID of the managed identity.

JSON

```
{
  "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.ps1"],
  "commandToExecute": "powershell.exe script1.ps1",
  "managedIdentity" : { "clientId": "00001111-aaaa-2222-bbbb-3333cccc4444" }
}
```

JSON

```
{
  "fileUris": ["https://mystorage.blob.core.windows.net/privatecontainer/script1.ps1"],
  "commandToExecute": "powershell.exe script1.ps1",
  "managedIdentity" : { "objectId": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb" }
}
```

> ⓘ Note
>
> The `managedIdentity` property *must not* be used in conjunction with the `storageAccountName` or `storageAccountKey` property.

Template deployment

You can deploy Azure VM extensions by using Azure Resource Manager templates. The JSON schema detailed in the previous section can be used in an Azure Resource Manager template to run the Custom Script Extension during the template's deployment. The following samples show how to use the Custom Script Extension:

- Deploy virtual machine extensions with Azure Resource Manager templates
- Deploy Two Tier Application on Windows and Azure SQL Database

PowerShell deployment

You can use the `Set-AzVMCustomScriptExtension` command to add the Custom Script Extension to an existing virtual machine. For more information, see Set-AzVMCustomScriptExtension.

PowerShell

```
Set-AzVMCustomScriptExtension -ResourceGroupName <resourceGroupName> `
    -VMName <vmName> `
    -Location myLocation `
    -FileUri <fileUrl> `
    -Run 'myScript.ps1' `
    -Name DemoScriptExtension
```

Examples

Using multiple scripts

This example uses three scripts to build your server. The `commandToExecute` property calls the first script. You then have options on how the others are called. For example, you can have a lead script that controls the execution, with the right error handling, logging, and state management. The scripts are downloaded to the local machine to run.

For example, in *1_Add_Tools.ps1*, you would call *2_Add_Features.ps1* by adding `.\2_Add_Features.ps1` to the script. Repeat this process for the other scripts that you define in `$settings`.

PowerShell

```
$fileUri = @("https://xxxxxxx.blob.core.windows.net/buildServer1/1_Add_Tools.ps1",
"https://xxxxxxx.blob.core.windows.net/buildServer1/2_Add_Features.ps1",
"https://xxxxxxx.blob.core.windows.net/buildServer1/3_CompleteInstall.ps1")

$settings = @{"fileUris" = $fileUri};

$storageAcctName = "xxxxxxx"
$storageKey = "1234ABCD"
$protectedSettings = @{"storageAccountName" = $storageAcctName; "storageAccountKey" = $storageKey; "commandToExecute" =
"powershell -ExecutionPolicy Unrestricted -File 1_Add_Tools.ps1"};

#run command
Set-AzVMExtension -ResourceGroupName <resourceGroupName> `
    -Location <locationName> `
    -VMName <vmName> `
    -Name "buildserver1" `
    -Publisher "Microsoft.Compute" `
    -ExtensionType "CustomScriptExtension" `
    -TypeHandlerVersion "1.10" `
    -Settings $settings `
    -ProtectedSettings $protectedSettings;
```

Running scripts from a local share

In this example, you might want to use a local Server Message Block (SMB) server for your script location. You then don't need to provide any other settings, except `commandToExecute`.

PowerShell

```
$protectedSettings = @{"commandToExecute" = "powershell -ExecutionPolicy Unrestricted -File \\filesvr\build\serverUpdate1.ps1"};

Set-AzVMExtension -ResourceGroupName <resourceGroupName> `
    -Location <locationName> `
```

```
    -VMName <vmName> `
    -Name "serverUpdate"
    -Publisher "Microsoft.Compute" `
    -ExtensionType "CustomScriptExtension" `
    -TypeHandlerVersion "1.10" `
    -ProtectedSettings $protectedSettings
```

Running a custom script more than once by using the CLI

The Custom Script Extension handler prevents rerunning a script if the *exact* same settings have been passed. This behavior prevents accidental rerunning, which might cause unexpected behaviors if the script isn't idempotent. To confirm whether the handler blocked the rerunning, look at `C:\WindowsAzure\Logs\Plugins\Microsoft.Compute.CustomScriptExtension\<HandlerVersion>\CustomScriptHandler.log*`. Searching for a warning like this one:

Output

```
Current sequence number, <SequenceNumber>, is not greater than the sequence number
of the most recently executed configuration. Exiting...
```

If you want to run the Custom Script Extension more than once, you can do that only under these conditions:

- The extension's `Name` parameter is the same as the previous deployment of the extension.
- You've updated the configuration. You can add a dynamic property to the command, such as a timestamp. If the handler detects a change in the configuration settings, it considers that change as an explicit desire to rerun the script.

Alternatively, you can set the ForceUpdateTag property to `true`.

Using Invoke-WebRequest

If you're using Invoke-WebRequest in your script, you must specify the parameter `-UseBasicParsing`. If you don't specify the parameter, you get the following error when checking the detailed status:

Output

```
The response content cannot be parsed because the Internet Explorer engine
is not available, or Internet Explorer's first-launch configuration
is not complete. Specify the UseBasicParsing parameter and try again.
```

Virtual Machine Scale Sets

If you deploy the Custom Script Extension from the Azure portal, you don't have control over the expiration of the SAS token for accessing the script in your storage account. The initial deployment works, but when the storage account's SAS token expires, any subsequent scaling operation fails because the Custom Script Extension can no longer access the storage account.

We recommend that you use PowerShell, the Azure CLI, or an Azure Resource Manager template when you deploy the Custom Script Extension on a Virtual Machine Scale Set. This way, you can choose to use a managed identity or have direct control of the expiration of the SAS token for accessing the script in your storage account for as long as you need.

Troubleshoot and support

You can retrieve data about the state of extension deployments from the Azure portal and by using the Azure PowerShell module. To see the deployment state of extensions for a VM, run the following command:

PowerShell

```
Get-AzVMExtension -ResourceGroupName <resourceGroupName> `
    -VMName <vmName> -Name myExtensionName
```

Extension output is logged to files found under the following folder on the target virtual machine:

Windows Command Prompt

```
C:\WindowsAzure\Logs\Plugins\Microsoft.Compute.CustomScriptExtension
```

The specified files are downloaded into the following folder on the target virtual machine:

Windows Command Prompt

```
C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.*\Downloads\<n>
```

In the preceding path, `<n>` is a decimal integer that might change between executions of the extension. The `1.*` value matches the actual, current `typeHandlerVersion` value of the extension. For example, the actual directory could be `C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.8\Downloads\2`.

When you run the `commandToExecute` command, the extension sets this directory, for example, `...\Downloads\2`, as the current working directory. This process enables the use of relative paths to locate the files downloaded by using the `fileURIs` property. Here are examples of downloaded files:

Expand table

URI in `fileUris`	Relative download location	Absolute download location
`https://someAcct.blob.core.windows.net/aContainer/scripts/myscript.ps1`	`./scripts/myscript.ps1`	`C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1`
`https://someAcct.blob.core.windows.net/aContainer/topLevel.ps1`	`./topLevel.ps1`	`C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1`

The absolute directory paths change over the lifetime of the VM, but not within a single execution of the Custom Script Extension.

Because the absolute download path might vary over time, it's better to opt for relative script/file paths in the `commandToExecute` string, whenever possible. For example:

JSON

```
"commandToExecute": "powershell.exe . . . -File \"./scripts/myscript.ps1\""
```

Path information after the first URI segment is kept for files downloaded by using the `fileUris` property list. As shown in the earlier table, downloaded files are mapped into download subdirectories to reflect the structure of the `fileUris` values.

Support

- If you need help with any part of this article, contact the Azure experts at Azure Community Support.
- To file an Azure support incident, go to the Azure support site and select **Get support**.
- For information about using Azure support, read the Microsoft Azure support FAQ.

Microsoft Antimalware Extension for Windows

Overview

The threat landscape in cloud environments is constantly evolving, making it essential for organizations to maintain strong security and compliance. Microsoft Antimalware for Azure provides free, real-time protection to help detect and remove viruses, spyware, and other malicious software. It offers configurable alerts when known threats or unwanted software attempt to install or run on your Azure systems. This solution is built on the same antimalware platform as Microsoft Security Essentials (MSE), Microsoft Forefront Endpoint Protection, Microsoft System Center Endpoint Protection, Windows Intune, and Windows Defender for Windows 8.0 and later.

Microsoft Antimalware for Azure is a single-agent solution for applications and tenant environments, designed to run in the background without human intervention. You can deploy protection based on the needs of your application workloads, with either basic secure-by-default or advanced custom configuration, including antimalware monitoring.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

Operating system

The Microsoft Antimalware for Azure solution includes the Microsoft Antimalware Client, and Service, Antimalware classic deployment model, Antimalware PowerShell cmdlets, and Azure Diagnostics Extension. The Microsoft Antimalware solution is supported on Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2 operating system families. It isn't supported on the Windows Server 2008 operating system, and also isn't supported in Linux.

Windows Defender is the built-in Antimalware enabled in Windows Server 2016. The Windows Defender Interface is also enabled by default on some Windows Server 2016 SKUs. The Azure VM Antimalware extension can still be added to a Windows Server 2016 and above Azure VM

with Windows Defender. In this scenario, the extension applies any optional configuration policies to be used by Windows Defender. The extension doesn't deploy any other antimalware service. For more information, see Microsoft Antimalware for Azure Cloud Services and Virtual Machines: Samples.

Internet connectivity

The Microsoft Antimalware for Windows requires that the target virtual machine is connected to the internet to receive regular engine and signature updates.

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. Templates are ideal when deploying one or more virtual machines that require post deployment configuration such as onboarding to Azure Antimalware.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource, or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the VM extension is nested inside the virtual machine resource. When the extension resource is nested, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "name": "[concat(parameters('vmName'),'/', parameters('vmExtensionName'))]",
      "apiVersion": "2019-07-01",
      "location": "[resourceGroup().location]",
      "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', parameters('vmName'))]"
      ],

      "properties": {
        "publisher": "Microsoft.Azure.Security",
        "type": "IaaSAntimalware",
        "typeHandlerVersion": "1.3",
        "autoUpgradeMinorVersion": true,
        "settings": {
          "AntimalwareEnabled": "true",
          "Exclusions": {
            "Extensions": ".ext1;.ext2",
            "Paths": "c:\excluded-path-1;c:\excluded-path-2",
```

```
            "Processes": "excludedproc1.exe;excludedproc2.exe"
          },

          "RealtimeProtectionEnabled": "true",
          "ScheduledScanSettings": {
            "isEnabled": "true",
            "scanType": "Quick",
            "day": "7",
            "time": "120"
          }
        },
        "protectedSettings": null
      }
}
```

You must include, at a minimum, the following content to enable the Microsoft Antimalware extension:

`{ "AntimalwareEnabled": true }`

Microsoft Antimalware JSON configuration sample:

JSON

```
{ "AntimalwareEnabled": true, "RealtimeProtectionEnabled": true,
"ScheduledScanSettings": { "isEnabled": true, "day": 1, "time": 120, "scanType":
"Full" },

"Exclusions": { "Extensions": ".ext1;.ext2", "Paths": "c:\excluded-path-
1;c:\excluded-path-2", "Processes": "excludedproc1.exe;excludedproc2.exe" }
}
```

AntimalwareEnabled:

- required parameter
- Values: true/false
  - true = Enable
  - false = Error out, as false isn't a supported value

RealtimeProtectionEnabled:

- Values: true/false, default is true
  - true = Enable
  - false = Disable

ScheduledScanSettings:

- isEnabled = true/false

- day = 0-8 (0-daily, 1-Sunday, 2-Monday, ..., 7-Saturday, 8-Disabled)
- time = 0-1440 (measured in minutes after midnight - 60->1AM, 120 -> 2AM, ...)
- scanType = Quick/Full, default is Quick
- If isEnabled = true is the only setting provided, the following defaults are set: day=7 (Saturday), time=120 (2 AM), scanType="Quick"

Exclusions

- Semicolons delimit multiple exclusions in the same list.
- If no exclusions are specified, the system overwrites existing exclusions with blanks.

PowerShell deployment

Use the appropriate commands to deploy the Azure Antimalware extension to an existing virtual machine, based on your deployment type.

- Azure Resource Manager based Virtual Machine
- Azure Service Fabric Clusters
- Azure Arc-enabled servers

Troubleshoot and support

Troubleshoot

Microsoft Antimalware extension logs are available at - %Systemdrive%\WindowsAzure\Logs\Plugins\Microsoft.Azure.Security.IaaSAntimalware(OrPaaSAntimalware)\1.5.5.x(version#)\CommandExecution.log

Error codes and their meanings

Expand table

Error code	Meaning	Possible action
-2147156224	MSI is busy with different installation	Try running installation later
-2147156221	MSE setup already running	Run only one instance at a time
-2147156208	Low disk space < 200 MB	Delete unused files, and retry installation

Error code	Meaning	Possible action
-2147156187	Last installation, upgrade, update, or uninstall requested reboot	Reboot, and retry installation
-2147156121	Setup tried to remove competitor product. But competitor product uninstall failed	Try to remove the competitor product manually, reboot, and retry installation
-2147156116	Policy file validation failed	Make sure you pass a valid policy XML file to set up
-2147156095	Setup couldn't start the Antimalware service	Verify all binaries are correctly signed, and right licensing file is installed
-2147023293	A fatal error occurred during installation. Epp.msi can't register\start\stop AM service or mini filter driver	MSI logs from EPP.msi are required here for future investigation
-2147023277	Installation package couldn't be opened	Verify that the package exists, and is accessible, or contact the application vendor to verify that it is a valid Windows Installer package
-2147156109	Windows Defender is required as a prerequisite	
-2147205073	The websso issuer isn't supported	
-2147024893	The system can't find the path specified	
-2146885619	Not a cryptographic message or the cryptographic message isn't formatted correctly	
-1073741819	The instruction at 0x%p referenced memory at 0x%p. The memory couldn't be %s	
1	Incorrect Function	

Support

If you need more help at any point in this article, you can contact the Azure experts on the Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site, and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 11/10/2025

VM Snapshot Linux extension for Azure Backup

Summarize this article for me

Azure Backup provides support for backing up workloads from on-premises to cloud and backing up cloud resources to Recovery Services vault. Azure Backup uses VM snapshot extension to take an application consistent backup of the Azure virtual machine without the need to shutdown the VM. VM Snapshot Linux extension is published and supported by Microsoft as part of Azure Backup service. Azure Backup will install the extension as part of first scheduled backup triggered post enabling backup. This document details the supported platforms, configurations, and deployment options for the VM Snapshot extension.

The VMSnapshot extension appears in the Azure portal only for non-managed VMs.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Operating system

For a list of supported operating systems, please refer to Operating Systems supported by Azure Backup

Extension schema

The following JSON shows the schema for the VM snapshot extension. The extension requires the task ID - this identifies the backup job which triggered snapshot on the VM, status blob uri - where status of the snapshot operation is written, scheduled start time of the snapshot, logs blob uri - where logs corresponding to snapshot task are written, objstr- representation of VM disks and meta data. Because these settings should be treated as sensitive data, it should be stored in a protected setting configuration. Azure VM extension protected setting data is encrypted, and only decrypted on the target virtual machine. Please note that these settings are recommended to be passed from Azure Backup service only as part of Backup job.

JSON

```
{
  "type": "extensions",
  "name": "VMSnapshot",
  "location":"<myLocation>",
  "properties": {
    "publisher": "Microsoft.RecoveryServices",
    "type": "VMSnapshot",
    "typeHandlerVersion": "1.9",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "locale":"<location>",
      "taskId":"<taskId used by Azure Backup service to communicate with extension>",
      "commandToExecute": "snapshot",
```

```
      "commandStartTimeUTCTicks": "<scheduled start time of the snapshot task>",
      "vmType": "microsoft.compute/virtualmachines"
    },
    "protectedSettings": {
      "objectStr": "<blob SAS uri representation of VM sent by Azure Backup service to extension>",
      "logsBlobUri": "<blob uri where logs of command execution by extension are written to>",
      "statusBlobUri": "<blob uri where status of the command executed by extension is written>"
    }
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	2015-06-15	date
taskId	e07354cf-041e-4370-929f-25a319ce8933_1	string
commandStartTimeUTCTicks	6.36458E+17	string
locale	en-us	string
objectStr	Encoding of sas uri array- "blobSASUri": ["https://sopattna5365.blob.core.windows.net/vhds/vmubuntu1404ltsc201652903941.vhd?sv=2014-02-14&sr=b&sig=TywkROXL1zvhXcLujtCut8g3jTpgbE6JpSWRLZxAdtA%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna8461.blob.core.windows.net/vhds/vmubuntu1404ltsc-20160629-122418.vhd?sv=2014-02-14&sr=b&sig=5S0A6YDWvVwqPAkzWXVy%2BS%2FqMwzFMbamT5upwx05v8Q%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna8461.blob.core.windows.net/bootdiagnostics-vmubuntu1-deb58392-ed5e-48be-9228-ff681b0cd3ee/vmubuntu1404ltsc-20160629-122541.vhd?sv=2014-02-14&sr=b&sig=X0Me2djByksBBMVXMGIUrcycvhQSfjYvqKLeRA7nBD4%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna5365.blob.core.windows.net/vhds/vmubuntu1404ltsc-20160701-163922.vhd?sv=2014-02-14&sr=b&sig=oXvtK2IXCNqWv7fpjc7TAzFDpc1GoXtT7r%2BC%2BNIAork%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna5365.blob.core.windows.net/vhds/vmubuntu1404ltsc-20170705-124311.vhd?sv=2014-02-14&sr=b&sig=ZUM9d28Mvvm%2FfrhJ71TFZh0Ni90m38bBs3zMl%2FQ9rs0%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw"]	string
logsBlobUri	["https://seapod01coord1exsapk732.blob.core.windows.net/bcdrextensionlogs-d45d8a1c-281e-4bc8-9d30-3b25176f68ea/sopattna-vmubuntu1404ltsc.v2.Logs.txt?sv=2014-02-14&sr=b&sig=DbwYhwfeAC5YJzISgxoKk%2FEWQq2AO1vS1E0rDW%2FlsBw%3D&st=2017-11-09T14%3A33%3A29Z&se=2017-11-09T17%3A38%3A29Z&sp=rw"]	string
statusBlobUri	["https://seapod01coord1exsapk732.blob.core.windows.net/bcdrextensionlogs-d45d8a1c-281e-4bc8-9d30-3b25176f68ea/sopattna-vmubuntu1404ltsc.v2.Status.txt?sv=2014-02-14&sr=b&sig=96RZBpTKCjmV7QFeXm5IduB%2FILktwGbLwbWg6Ih96Ao%3D&st=2017-11-09T14%3A33%3A29Z&se=2017-11-09T17%3A38%3A29Z&sp=rw"]	string

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. However, the recommended way of adding a VM snapshot extension to a virtual machine is by enabling backup on the virtual machine. This can be achieved through a Resource Manager template. A sample Resource Manager template that enables backup on a virtual machine can be found on the Azure Quick Start Gallery.

Azure CLI deployment

The Azure CLI can be used to enable backup on a virtual machine. Post enable backup, first scheduled backup job will install the Vm snapshot extension on the VM.

Azure CLI

```
az backup protection enable-for-vm \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --vm myVM \
    --policy-name DefaultPolicy
```

Azure PowerShell deployment

Azure PowerShell can be used to enable backup on a virtual machine. Once the backup is configured, first scheduled backup job will install the Vm snapshot extension on the VM.

Azure PowerShell

```
$targetVault = Get-AzRecoveryServicesVault -ResourceGroupName "myResourceGroup" -Name
"myRecoveryServicesVault"
$pol = Get-AzRecoveryServicesBackupProtectionPolicy Name DefaultPolicy -VaultId $targetVault.ID
Enable-AzRecoveryServicesBackupProtection -Policy $pol -Name "myVM" -ResourceGroupName
"myVMResourceGroup" -VaultId $targetVault.ID
```

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure CLI. To see the deployment state of extensions for a given VM, run the following command using the Azure CLI.

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension execution output is logged to the following file:

```
/var/log/waagent.log
```

Error codes and their meanings

Troubleshooting information can be found on the Azure VM backup troubleshooting guide.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 02/27/2026

VM Snapshot Windows extension for Azure Backup

Summarize this article for me

Azure Backup provides support for backing up workloads from on-premises to cloud and backing up cloud resources to Recovery Services vault. Azure Backup uses VM snapshot extension to take an application consistent backup of the Azure virtual machine without the need to shutdown the VM. VM Snapshot extension is published and supported by Microsoft as part of Azure Backup service. Azure Backup will install the extension as part of first scheduled backup triggered post enabling backup. This document details the supported platforms, configurations, and deployment options for the VM Snapshot extension.

The VMSnapshot extension appears in the Azure portal only for non-managed VMs.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Operating system

For a list of supported operating systems, refer to Operating Systems supported by Azure Backup

Extension schema

The following JSON shows the schema for the VM snapshot extension. The extension requires the task ID - this identifies the backup job which triggered snapshot on the VM, status blob uri - where status of the snapshot operation is written, scheduled start time of the snapshot, logs blob uri - where logs corresponding to snapshot task are written, objstr- representation of VM disks and meta data. Because these settings should be treated as sensitive data, it should be stored in a protected setting configuration. Azure VM extension protected setting data is encrypted, and only decrypted on the target virtual machine. Note that these settings are recommended to be passed from Azure Backup service only as part of Backup job.

JSON

```
{
  "type": "extensions",
  "name": "VMSnapshot",
  "location":"<myLocation>",
  "properties": {
```

```
    "publisher": "Microsoft.Azure.RecoveryServices",
    "type": "VMSnapshot",
    "typeHandlerVersion": "1.9",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "locale":"<location>",
      "taskId":"<taskId used by Azure Backup service to communicate with extension>",
      "commandToExecute": "snapshot",
      "commandStartTimeUTCTicks": "<scheduled start time of the snapshot task>",
      "vmType": "microsoft.compute/virtualmachines"
    },
    "protectedSettings": {
      "objectStr": "<blob SAS uri representation of VM sent by Azure Backup service to
extension>",
      "logsBlobUri": "<blob uri where logs of command execution by extension are written to>",
      "statusBlobUri": "<blob uri where status of the command executed by extension is written>"
    }
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	2015-06-15	date
taskId	e07354cf-041e-4370-929f-25a319ce8933_1	string
commandStartTimeUTCTicks	6.36458E+17	string
locale	en-us	string
objectStr	Encoding of sas uri array- "blobSASUri": ["https://sopattna5365.blob.core.windows.net/vhds/vmwin1404ltsc201652903941.vhd?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna8461.blob.core.windows.net/vhds/vmwin1404ltsc-20160629-122418.vhd?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna8461.blob.core.windows.net/bootdiagnostics-vmwintu1-deb58392-ed5e-48be-9228-ff681b0cd3ee/vmubuntu1404ltsc-20160629-122541.vhd?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna5365.blob.core.windows.net/vhds/vmwin1404ltsc-20160701-163922.vhd?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw", "https://sopattna5365.blob.core.windows.net/vhds/vmwin1404ltsc-20170705-124311.vhd?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A23%3A28Z&se=2017-11-09T17%3A38%3A28Z&sp=rw"]	string
logsBlobUri	["https://seapod01coord1exsapk732.blob.core.windows.net/bcdrextensionlogs-d45d8a1c-281e-4bc8-9d30-3b25176f68ea/sopattna-vmubuntu1404ltsc.v2.Logs.txt?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A33%3A29Z&se=2017-11-09T17%3A38%3A29Z&sp=rw"]	string

Name	Value / Example	Data Type
statusBlobUri	["https://seapod01coord1exsapk732.blob.core.windows.net/bcdrextensionlogs-d45d8a1c-281e-4bc8-9d30-3b25176f68ea/sopattna-vmubuntu1404ltsc.v2.Status.txt?sv=2014-02-14&sr=b&sig=abc123DEF456...%3D&st=2017-11-09T14%3A33%3A29Z&se=2017-11-09T17%3A38%3A29Z&sp=rw"]	string

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. However, the recommended way of adding a VM snapshot extension to a virtual machine is by enabling backup on the virtual machine. This can be achieved through a Resource Manager template. A sample Resource Manager template that enables backup on a virtual machine can be found on the Azure Quick Start Gallery.

Azure CLI deployment

The Azure CLI can be used to enable backup on a virtual machine. Post enable backup, first scheduled backup job will install the Vm snapshot extension on the VM.

Azure CLI

```
az backup protection enable-for-vm \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --vm myVM \
    --policy-name DefaultPolicy
```

Azure PowerShell deployment

Azure PowerShell can be used to enable backup on a virtual machine. Once the backup is configured, first scheduled backup job will install the Vm snapshot extension on the VM.

Azure PowerShell

```
$targetVault = Get-AzRecoveryServicesVault -ResourceGroupName "myResourceGroup" -Name
"myRecoveryServicesVault"
$pol = Get-AzRecoveryServicesBackupProtectionPolicy Name DefaultPolicy -VaultId $targetVault.ID
Enable-AzRecoveryServicesBackupProtection -Policy $pol -Name "myVM" -ResourceGroupName
"myVMResourceGroup" -VaultId $targetVault.ID
```

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure CLI. To see the deployment state of extensions for a given VM, run the following command using the

Azure CLI.

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension execution output is logged to the following file:

```
C:\Packages\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot
```

Error codes and their meanings

Troubleshooting information can be found on the Azure VM backup troubleshooting guide.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 02/27/2026

Manage Network Watcher Agent virtual machine extension for Linux

09/23/2025

The Network Watcher Agent virtual machine extension is a requirement for some of Azure Network Watcher features that capture network traffic to diagnose and monitor Azure virtual machines (VMs). For more information, see What is Azure Network Watcher?

In this article, you learn how to install and uninstall Network Watcher Agent for Linux. Installation of the agent doesn't disrupt, or require a reboot of the virtual machine. If the virtual machine is deployed by an Azure service, check the documentation of the service to determine whether or not it permits installing extensions in the virtual machine.

> **Note**
>
> Network Watcher Agent extension is not supported on AKS clusters.

Prerequisites

Portal

- An Azure Linux virtual machine (VM). For more information, see Supported Linux distributions and versions.
- Outbound TCP connectivity to `169.254.169.254` over `port 80` and `168.63.129.16` over `port 8037`. The agent uses these IP addresses to communicate with the Azure platform.
- Internet connectivity: Network Watcher Agent requires internet connectivity for some features to properly work. For example, it requires connectivity to your storage account to upload packet captures.

Supported operating systems

Network Watcher Agent extension for Linux can be installed on the following Linux distributions:

Expand table

Distribution	Version
AlmaLinux	9.2
Azure Linux	2.0
CentOS [1]	6.10 and 7
Debian	7 and 8
openSUSE Leap	42.3+
Oracle Linux	6.10 [2], 7 and 8+
Red Hat Enterprise Linux (RHEL)	6.10 [3], 7, 8 and 9.2
Rocky Linux	9.1
SUSE Linux Enterprise Server (SLES)	12 and 15 (SP2, SP3, and SP4)
Ubuntu	16+

[1] CentOS Linux reached its end-of-life (EOL) on June 30, 2024. For more information, see the CentOS End Of Life guidance.

[2] Extended life cycle (ELS) support for Oracle Linux version 6.X ended on July 1, 2024.

[3] Extended life cycle (ELS) support for Red Hat Enterprise Linux 6.X ended on June 30, 2024.

Extension schema

The following JSON shows the schema for the Network Watcher Agent extension. The extension doesn't require, or support, any user-supplied settings. The extension relies on its default configuration.

JSON

```
{
    "name": "[concat(parameters('vmName'), '/AzureNetworkWatcherExtension')]",
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "apiVersion": "2023-03-01",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', parameters('vmName'))]"
    ],
    "properties": {
```

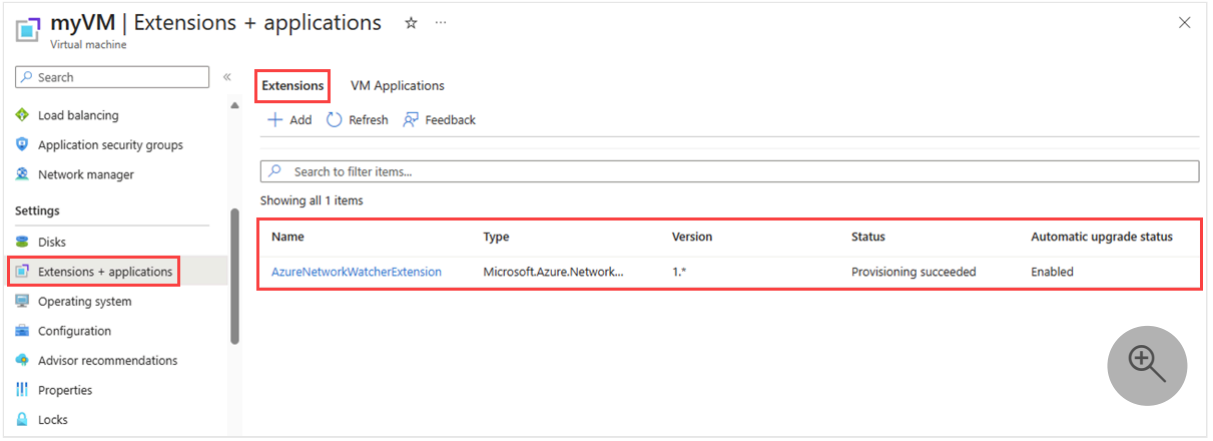
```
        "autoUpgradeMinorVersion": true,
        "publisher": "Microsoft.Azure.NetworkWatcher",
        "type": "NetworkWatcherAgentLinux",
        "typeHandlerVersion": "1.4"
    }
}
```

List installed extensions

Portal

From the virtual machine page in the Azure portal, you can view the installed extension by following these steps:

1. Under **Settings**, select **Extensions + applications**.
2. In the **Extensions** tab, you can see all installed extensions on the virtual machine. If the list is long, you can use the search box to filter the list.

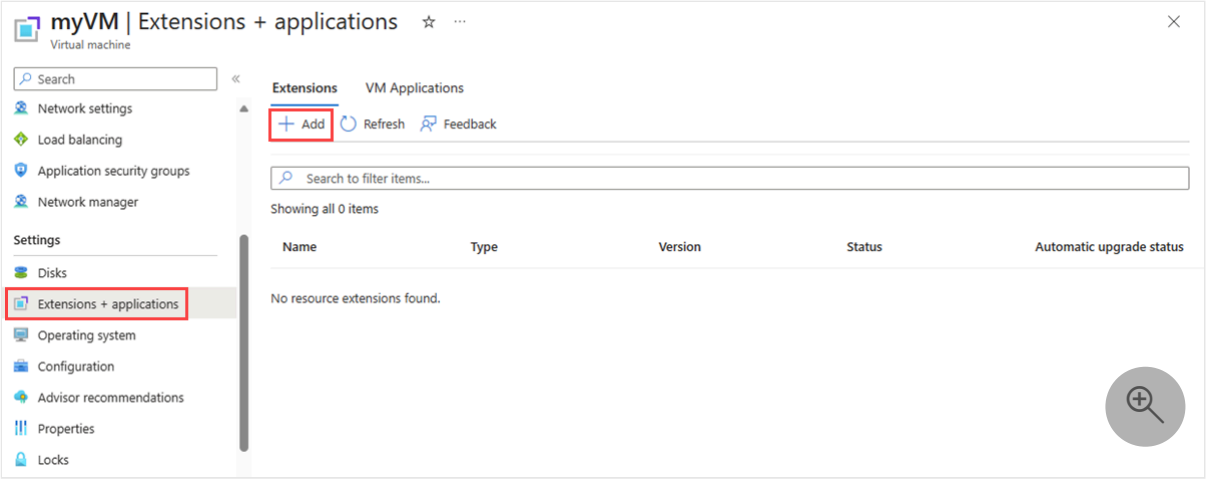


Install Network Watcher Agent VM extension

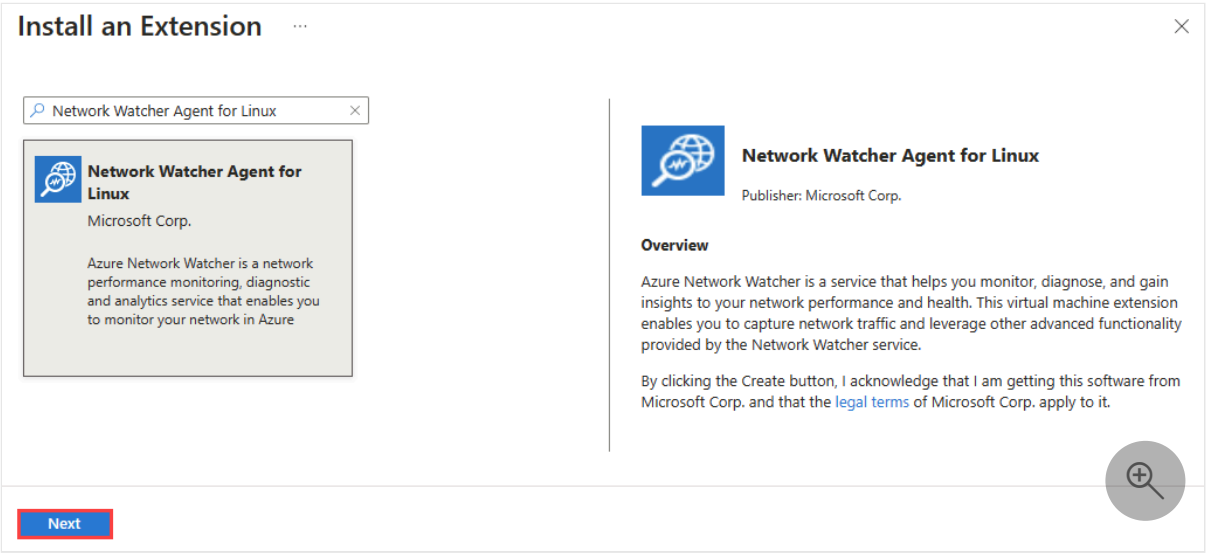
Portal

From the virtual machine page in the Azure portal, you can install the Network Watcher Agent VM extension by following these steps:

1. Under **Settings**, select **Extensions + applications**.
2. Select **+ Add** and search for **Network Watcher Agent** and install it. If the extension is already installed, you can see it in the list of extensions.



3. In the search box of **Install an Extension**, enter *Network Watcher Agent for Linux*. Select the extension from the list and select **Next**.



4. Select **Review + create** and then select **Create**.

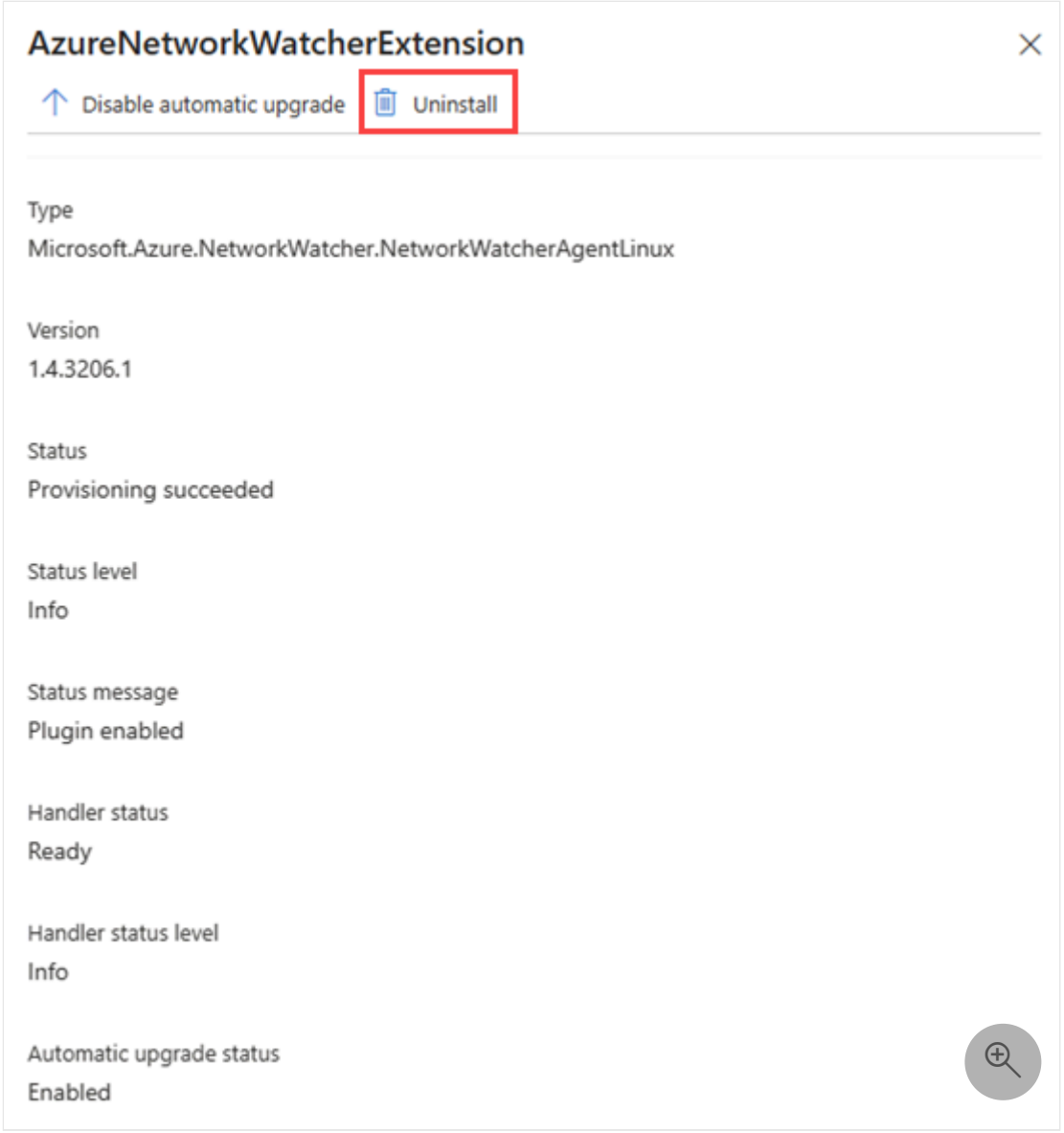
Uninstall Network Watcher Agent VM extension

Portal

From the virtual machine page in the Azure portal, you can uninstall the Network Watcher Agent VM extension by following these steps:

1. Under **Settings**, select **Extensions + applications**.

2. Select **AzureNetworkWatcherExtension** from the list of extensions, and then select **Uninstall**.



> **Note**
>
> You might see Network Watcher Agent VM extension named differently than **AzureNetworkWatcherExtension**.

Frequently asked questions (FAQ)

To get answers to most frequently asked questions about Network Watcher Agent, see Network Watcher Agent FAQ.

Related content

- Update Azure Network Watcher extension to the latest version.
- Network Watcher documentation.

- Microsoft Q&A - Network Watcher.

Manage Network Watcher Agent virtual machine extension for Windows

09/23/2025

The Network Watcher Agent virtual machine extension is a requirement for some of Azure Network Watcher features that capture network traffic to diagnose and monitor Azure virtual machines (VMs). For more information, see What is Azure Network Watcher?

In this article, you learn how to install and uninstall Network Watcher Agent for Windows. Installation of the agent doesn't disrupt, or require a reboot of the virtual machine. If the virtual machine is deployed by an Azure service, check the documentation of the service to determine whether or not it permits installing extensions in the virtual machine.

Prerequisites

Portal

- An Azure Windows virtual machine (VM). For more information, see Supported Windows versions.
- Outbound TCP connectivity to `169.254.169.254` over `port 80` and `168.63.129.16` over `port 8037`. The agent uses these IP addresses to communicate with the Azure platform.
- Internet connectivity: Network Watcher Agent requires internet connectivity for some features to properly work. For example, it requires connectivity to your storage account to upload packet captures.

Supported operating systems

Network Watcher Agent extension for Windows can be installed on:

- Windows Server 2012, 2012 R2, 2016, 2019 and 2022 releases.
- Windows 10 and 11 releases.

> ⓘ Note

Currently, Nano Server isn't supported.

Extension schema

The following JSON shows the schema for the Network Watcher Agent extension. The extension doesn't require, or support, any user-supplied settings, and relies on its default configuration.

JSON

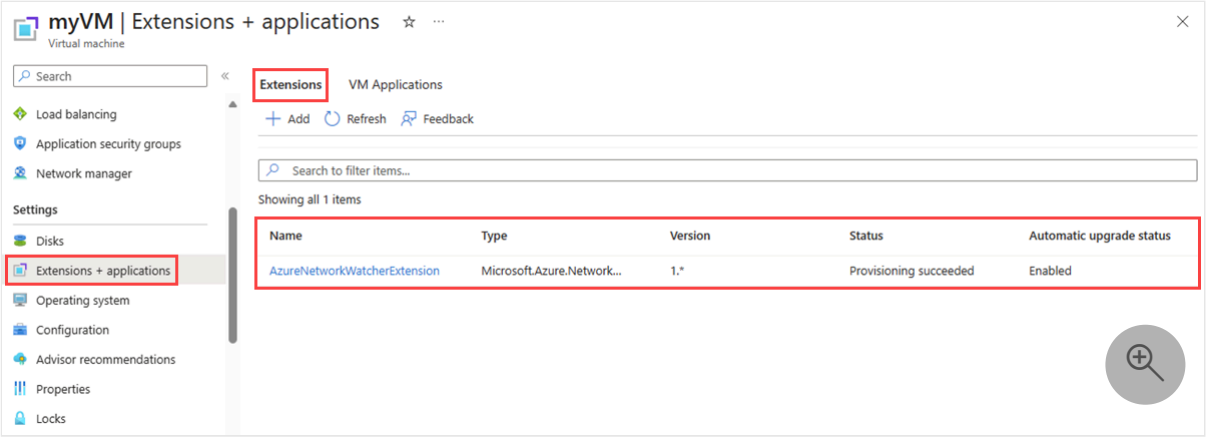
```
{
    "name": "[concat(parameters('vmName'), '/AzureNetworkWatcherExtension')]",
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "apiVersion": "2023-03-01",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', parameters('vmName'))]"
    ],
    "properties": {
        "autoUpgradeMinorVersion": true,
        "publisher": "Microsoft.Azure.NetworkWatcher",
        "type": "NetworkWatcherAgentWindows",
        "typeHandlerVersion": "1.4"
    }
}
```

List installed extensions

Portal

From the virtual machine page in the Azure portal, you can view the installed extension by following these steps:

1. Under **Settings**, select **Extensions + applications**.
2. In the **Extensions** tab, you can see all installed extensions on the virtual machine. If the list is long, you can use the search box to filter the list.



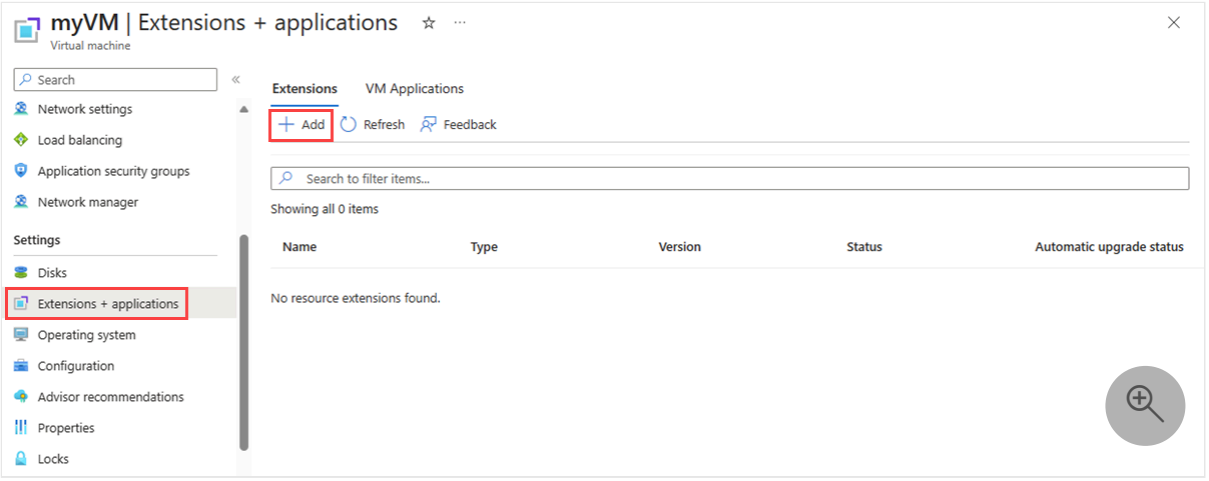
Install Network Watcher Agent VM extension

Portal

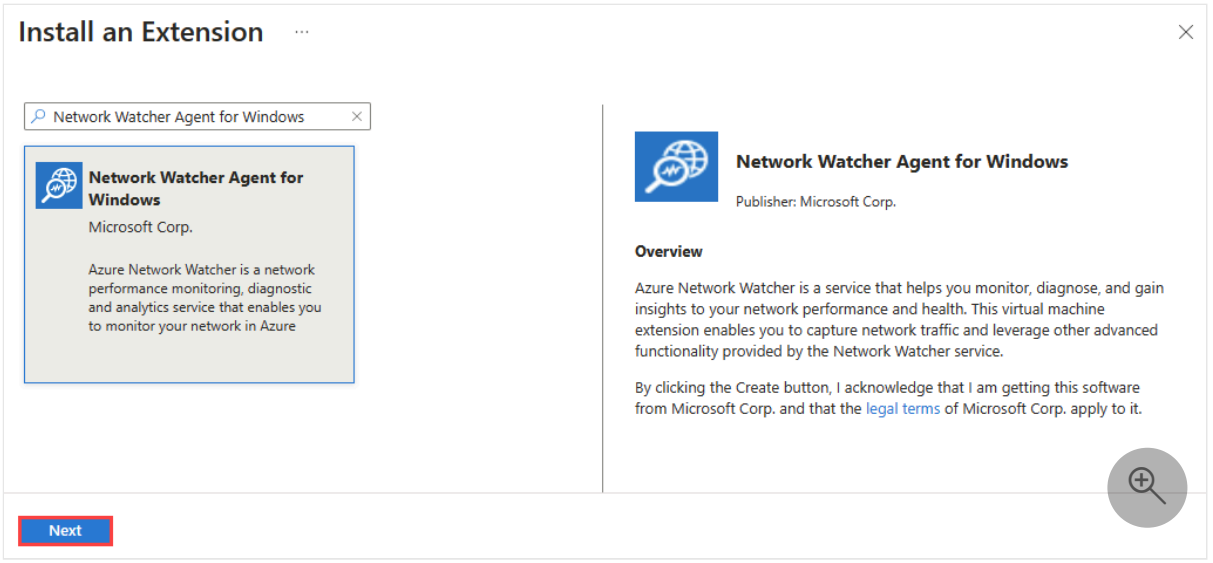
From the virtual machine page in the Azure portal, you can install the Network Watcher Agent VM extension by following these steps:

1. Under **Settings**, select **Extensions + applications**.

2. Select **+ Add** and search for **Network Watcher Agent** and install it. If the extension is already installed, you can see it in the list of extensions.



3. In the search box of **Install an Extension**, enter *Network Watcher Agent for Windows*. Select the extension from the list and select **Next**.



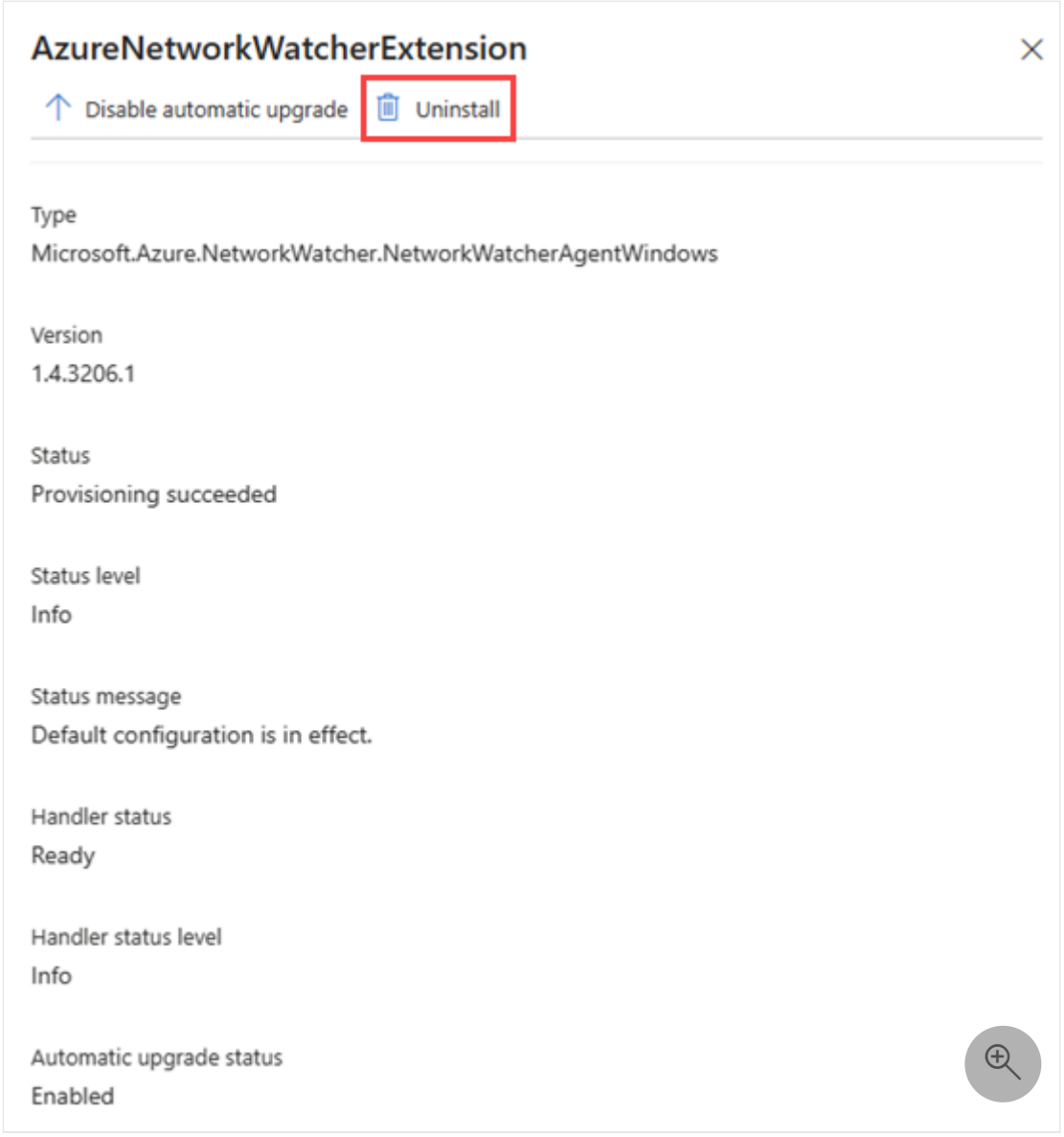
4. Select **Review + create** and then select **Create**.

Uninstall Network Watcher Agent VM extension

Portal

From the virtual machine page in the Azure portal, you can uninstall the Network Watcher Agent VM extension by following these steps:

1. Under **Settings**, select **Extensions + applications**.
2. Select **AzureNetworkWatcherExtension** from the list of extensions, and then select **Uninstall**.



> **Note**
>
> You might see Network Watcher Agent VM extension named differently than **AzureNetworkWatcherExtension**.

Frequently asked questions (FAQ)

To get answers to most frequently asked questions about Network Watcher Agent, see Network Watcher Agent FAQ.

Related content

- Update Azure Network Watcher extension to the latest version.
- Network Watcher documentation.

- Microsoft Q&A - Network Watcher.

Update Azure Network Watcher extension to the latest version

Summarize this article for me

Azure Network Watcher is a network performance monitoring, diagnostic, and analytics service that monitors Azure networks. The Network Watcher Agent virtual machine (VM) extension is a requirement for capturing network traffic on demand and using other advanced functionality on Azure VMs. It's used by connection monitor, connection troubleshoot, and packet capture.

Prerequisites

- An Azure account with an active subscription. If you don't have one, create a free account.
- An Azure virtual machine (VM) that has the Network Watcher extension installed.

Latest version

The latest version of the Network Watcher extension is `1.4.3783.1`.

> **Note**
>
> When automatic extension upgrade is enabled, there may be a delay of up to 30 days between the release of a new extension version and its automatic upgrade on your virtual machines and scale sets.

Identify latest version

Use az vm extension image list command to identify the latest version of the Network Watcher extension for your VM's operating system.

Windows

Azure CLI

```
# Identify latest version of Network Watcher extension for Windows.
az vm extension image list --name 'NetworkWatcherAgentWindows' --publisher
'Microsoft.Azure.NetworkWatcher' --latest --location 'eastus'
```

Update your extension using a PowerShell script

If you have large deployments, use a PowerShell script to update multiple VMs at once. The following PowerShell script updates Network Watcher extension of all Windows VMs in a subscription:

PowerShell

```
<#
    .SYNOPSIS
    This script will scan all VMs in the provided subscription and upgrade any out
of date AzureNetworkWatcherExtensions
    .DESCRIPTION
    This script should be no-op if AzureNetworkWatcherExtensions are up to date
    Requires Azure PowerShell 4.2 or higher to be installed (e.g. Install-Module
AzureRM).
    .EXAMPLE
    .\UpdateVMAgentsInSub.ps1 -SubID F4BC4873-5DAB-491E-B713-1358EF4992F2 -NoUpdate
#>

[CmdletBinding()]
param(
    [Parameter(Mandatory=$true)]
    [string] $SubID,
    [Parameter(Mandatory=$false)]
    [Switch] $NoUpdate = $false,
    [Parameter(Mandatory=$false)]
    [string] $MinVersion = "1.4.2.1"
)
function NeedsUpdate($version)
{
    if ([Version]$version -lt [Version]$MinVersion)
    {
        $lessThan = $true
    }else{
        $lessThan = $false
    }
    return $lessThan
}
Write-Host "Scanning all VMs in the subscription: $($SubID)"
Set-AzContext -SubscriptionId $SubID
$vms = Get-AzVM
$foundVMs = $false
Write-Host "Starting VM search, this may take a while"
foreach ($vmName in $vms)
{
    # Get Detailed VM info
    $vm = Get-AzVM -ResourceGroupName $vmName.ResourceGroupName -Name $vmName.name
-Status
    $isitWindows = $vm.OsName -like "*Windows*"

    foreach ($extension in $vm.Extensions)
```

```
    {
        if ($extension.Name -eq "AzureNetworkWatcherExtension")
        {
            if (NeedsUpdate($extension.TypeHandlerVersion))
            {
                $foundVMs = $true
                if (-not ($NoUpdate))
                {
                    Write-Host "Found VM that needs to be updated:
subscriptions/$($SubID)/resourceGroups/$($vm.ResourceGroupName)/providers/Microsoft
.Compute/virtualMachines/$($vm.Name) -> Updating " -NoNewline
                    Remove-AzVMExtension -ResourceGroupName $vm.ResourceGroupName -
VMName $vm.Name -Name "AzureNetworkWatcherExtension" -Force
                    Write-Host "... " -NoNewline
                    $type = if ($isitWindows) { "NetworkWatcherAgentWindows" } else
{ "NetworkWatcherAgentLinux" }
                    Set-AzVMExtension -ResourceGroupName $vm.ResourceGroupName -
Location $vmName.Location -VMName $vm.Name -Name "AzureNetworkWatcherExtension" -
Publisher "Microsoft.Azure.NetworkWatcher" -Type $type -typeHandlerVersion
$MinVersion
                    Write-Host "Done"
                }
                else
                {
                    Write-Host "Found $(if ($isitWindows) {"Windows"} else
{"Linux"}) VM that needs to be updated:
subscriptions/$($SubID)/resourceGroups/$($vm.ResourceGroupName)/providers/Microsoft
.Compute/virtualMachines/$($vm.Name)"
                }
            }
        }
    }
}

if ($foundVMs)
{
    Write-Host "Finished $(if ($NoUpdate) {"searching"} else {"updating"}) out of
date AzureNetworkWatcherExtension on VMs"
}
else
{
    Write-Host "All AzureNetworkWatcherExtensions up to date"
}
```

Update your extension manually

To update your extension, you need to know your extension version.

Check your extension version

You can check your extension version by using the Azure portal, the Azure CLI, or PowerShell.

Use the Azure portal

1. Go to the **Extensions** pane of your VM in the Azure portal.
2. Select the **AzureNetworkWatcher** extension to see the details pane.
3. Locate the version number in the **Version** field.

Use the Azure CLI

Run the following command from an Azure CLI prompt:

Azure CLI

```
az vm get-instance-view --resource-group  "SampleRG" --name "Sample-VM"
```

Locate "**AzureNetworkWatcherExtension**" in the output and identify the version number from the *"TypeHandlerVersion"* field in the output.

Information about the extension appears multiple times in the JSON output. The full version number of the extension is available under the Extensions block.

You should see something like the below:



Use PowerShell

Run the following commands from a PowerShell prompt:

PowerShell

```
Get-AzVM -ResourceGroupName "SampleRG" -Name "Sample-VM" -Status
```

Locate the Azure Network Watcher extension in the output and identify the version number from the *"TypeHandlerVersion"* field in the output.

You should see something like the below:

```
ExtensionHandlers[1]     :
  Type                   : Microsoft.Azure.NetworkWatcher.NetworkWatcherAgentWindows
  TypeHandlerVersion     : 1.4.1654.1
  Status                 :
    Code                 : ProvisioningState/succeeded
    Level                : Info
```

Update your extension

If your version is below the latest version mentioned above, update your extension by using any of the following options.

Option 1: Use PowerShell

Run the following commands:

PowerShell

```powershell
#Linux command
Set-AzVMExtension -ResourceGroupName "myResourceGroup1" -Location "WestUS" -VMName
"myVM1" -Name "AzureNetworkWatcherExtension" -Publisher
"Microsoft.Azure.NetworkWatcher" -Type "NetworkWatcherAgentLinux"

#Windows command
Set-AzVMExtension -ResourceGroupName "myResourceGroup1" -Location "WestUS" -VMName
"myVM1" -Name "AzureNetworkWatcherExtension" -Publisher
"Microsoft.Azure.NetworkWatcher" -Type "NetworkWatcherAgentWindows" -ForceRerun
"True"
```

If that doesn't work. Remove and install the extension again, using the steps below, to install latest version.

Removing the extension

PowerShell

```powershell
#Same command for Linux and Windows
Remove-AzVMExtension -ResourceGroupName "SampleRG" -VMName "Sample-VM" -Name
"AzureNetworkWatcherExtension"
```

Installing the extension again

PowerShell

```
#Linux command
Set-AzVMExtension -ResourceGroupName "SampleRG" -Location "centralus" -VMName
"Sample-VM" -Name "AzureNetworkWatcherExtension" -Publisher
"Microsoft.Azure.NetworkWatcher" -Type "NetworkWatcherAgentLinux" -
typeHandlerVersion "1.4"

#Windows command
Set-AzVMExtension -ResourceGroupName "SampleRG" -Location "centralus" -VMName
"Sample-VM" -Name "AzureNetworkWatcherExtension" -Publisher
"Microsoft.Azure.NetworkWatcher" -Type "NetworkWatcherAgentWindows" -
typeHandlerVersion "1.4"
```

Option 2: Use the Azure CLI

Force an upgrade.

Azure CLI

```
#Linux command
az vm extension set --resource-group "myResourceGroup1" --vm-name "myVM1" --name
"NetworkWatcherAgentLinux" --publisher "Microsoft.Azure.NetworkWatcher" --force-
update

#Windows command
az vm extension set --resource-group "myResourceGroup1" --vm-name "myVM1" --name
"NetworkWatcherAgentWindows" --publisher "Microsoft.Azure.NetworkWatcher" --force-
update
```

If that doesn't work, remove and install the extension again, and follow these steps to automatically add the latest version.

Remove the extension.

Azure CLI

```
#Same for Linux and Windows
az vm extension delete --resource-group "myResourceGroup1" --vm-name "myVM1" -n
"AzureNetworkWatcherExtension"
```

Install the extension again.

Azure CLI

```
#Linux command
az vm extension set --resource-group "DALANDEMO" --vm-name "Linux-01" --name
```

```
"NetworkWatcherAgentLinux" --publisher "Microsoft.Azure.NetworkWatcher"

#Windows command
az vm extension set --resource-group "DALANDEMO" --vm-name "Linux-01" --name
"NetworkWatcherAgentWindows" --publisher "Microsoft.Azure.NetworkWatcher"
```

Option 3: Reboot your VMs

If you have auto-upgrade set to true for the Network Watcher extension, reboot your VM installation to the latest extension.

Support

If you need more help at any point in this article, see the Network Watcher extension documentation for Linux or Windows. You can also contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, file an Azure support incident. Go to the Azure support site, and select **Get support**. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 02/25/2026

InfiniBand Driver Extension for Linux

This extension installs InfiniBand OFED drivers on InfiniBand and SR-IOV-enabled ('r' sizes) HB-series and N-series VMs running Linux. Depending on the VM family, the extension installs the appropriate drivers for the Connect-X NIC. It does not install the InfiniBand ND drivers on the non-SR-IOV enabled HB-series and N-series VMs.

Instructions on manual installation of the OFED drivers are available in Enable InfiniBand on HPC VMs.

An extension is also available to install InfiniBand drivers for Windows VMs.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

Operating system

This extension supports the following OS distros, depending on driver support for specific OS version. For latest list of supported OS and driver versions, refer to resources.json

Expand table

Distribution	Version	InfiniBand NIC drivers
Ubuntu	18.04 LTS, 20.04 LTS, 22.04 LTS	CX3-Pro, CX5, CX6
Red Hat Enterprise Linux	7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.1, 8,2	CX3-Pro, CX5, CX6

> **Important**
>
> This document references a release version of Linux that is nearing or at, End of Life (EOL). Please consider updating to a more current version.

Internet connectivity

The Microsoft Azure Extension for InfiniBand Drivers requires that the target VM is connected to and has access to the internet.

Extension schema

The following JSON shows the schema for the extension.

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "InfiniBandDriverLinux",
    "typeHandlerVersion": "1.2",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

Properties

Expand table

Name	Value / Example	Data Type
apiVersion	2015-06-15	date
publisher	Microsoft.HpcCompute	string
type	InfiniBandDriverLinux	string
typeHandlerVersion	1.2	int

Deployment

Azure Resource Manager Template

Azure VM extensions can be deployed with Azure Resource Manager templates. Templates are ideal when deploying one or more virtual machines that require post deployment configuration.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource, or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the extension is nested inside the virtual machine resource. When nesting the extension resource, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
  "name": "myExtensionName",
  "type": "extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2015-06-15",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', myVM)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "InfiniBandDriverLinux",
    "typeHandlerVersion": "1.2",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

PowerShell

PowerShell

```
Set-AzVMExtension
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Location "southcentralus" `
    -Publisher "Microsoft.HpcCompute" `
    -ExtensionName "InfiniBandDriverLinux" `
    -ExtensionType "InfiniBandDriverLinux" `
    -TypeHandlerVersion 1.2 `
    -SettingString '{ `
    }'
```

Azure CLI

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name InfiniBandDriverLinux \
  --publisher Microsoft.HpcCompute \
  --version 1.2
```

Add extension to a Virtual Machine Scale Set

The following example installs the latest version 1.2 InfiniBandDriverLinux extension on all RDMA-capable VMs in an existing virtual machine scale set named *myVMSS* deployed in the resource group named *myResourceGroup*:

PowerShell

```
$VMSS = Get-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "myVMSS"
Add-AzVmssExtension -VirtualMachineScaleSet $VMSS -Name "InfiniBandDriverLinux" -
Publisher "Microsoft.HpcCompute" -Type "InfiniBandDriverLinux" -TypeHandlerVersion
"1.2"
Update-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "MyVMSS" -
VirtualMachineScaleSet $VMSS
Update-AzVmssInstance -ResourceGroupName "myResourceGroup" -VMScaleSetName "myVMSS"
-InstanceId "*"
```

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using Azure PowerShell and Azure CLI. To see the deployment state of extensions for a given VM, run the following command.

PowerShell

```
Get-AzVMExtension -ResourceGroupName myResourceGroup -VMName myVM -Name
myExtensionName
```

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension execution output is logged to the following file. Refer to this file to track the status of the installation as well as for troubleshooting any failures.

```
Bash
/var/log/azure/ib-vmext-status
```

Exit codes

The following table describes the meaning and recommended action based on the exit codes of the extension installation process.

Expand table

Exit Code	Meaning	Possible Action
0	Operation successful	
1	Incorrect usage of extension	Check execution output log
10	Linux Integration Services for Hyper-V and Azure not available or installed	Check output of lspci
11	Mellanox InfiniBand not found on this VM size	Use a supported VM size and OS
12	Image offer not supported	
13	VM size not supported	Use an InfiniBand-enabled ('r' size) H-series and N-seriesN-series VM to deploy
14	Operation unsuccessful	Check execution output log

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file a support incident through the Azure support site. For information about using Azure Support, read the Microsoft Azure support FAQ.

Next steps

For more information about InfiniBand-enabled ('r' sizes), see HB-series and N-series VMs.

Learn more about Linux VMs extensions and features

Last updated on 11/10/2025

InfiniBand Driver Extension for Windows

This extension installs InfiniBand ND drivers (for non-SR-IOV enabled) and OFED drivers (for SR-IOV-enabled) ('r' sizes) H-series and N-series VMs running Windows. Depending on the VM family, the extension installs the appropriate drivers for the Connect-X NIC.

An extension is also available to install InfiniBand drivers for Linux VMs.

> ⓘ Note
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

Operating system

This extension supports the following OS distros, depending on driver support for specific OS version. Note the appropriate InfiniBand NIC for the H and N-series VM sizes of interest.

Expand table

Distribution	InfiniBand NIC drivers
Windows 10	CX5, CX6
Windows Server 2019	CX5, CX6
Windows Server 2016	CX5, CX6
Windows Server 2012 R2	CX5, CX6

For latest list of supported OS and driver versions, refer to resources.json

Internet connectivity

The Microsoft Azure Extension for InfiniBand Drivers requires that the target VM is connected to and has access to the internet.

Extension schema

The following JSON shows the schema for the extension.

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "InfiniBandDriverWindows",
    "typeHandlerVersion": "1.5",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

Properties

Expand table

Name	Value / Example	Data Type
apiVersion	2015-06-15	date
publisher	Microsoft.HpcCompute	string
type	InfiniBandDriverWindows	string
typeHandlerVersion	1.5	int

Deployment

Azure Resource Manager Template

Azure VM extensions can be deployed with Azure Resource Manager templates. Templates are ideal when deploying one or more virtual machines that require post deployment configuration.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource, or placed at the root or top level of a Resource Manager JSON template.

The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the extension is nested inside the virtual machine resource. When nesting the extension resource, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
  "name": "myExtensionName",
  "type": "extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2015-06-15",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', myVM)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "InfiniBandDriverWindows",
    "typeHandlerVersion": "1.5",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

PowerShell

PowerShell

```
Set-AzVMExtension
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Location "southcentralus" `
    -Publisher "Microsoft.HpcCompute" `
    -ExtensionName "InfiniBandDriverWindows" `
    -ExtensionType "InfiniBandDriverWindows" `
    -TypeHandlerVersion 1.5 `
    -SettingString '{ `
    }'
```

Azure CLI

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
```

```
  --vm-name myVM \
  --name InfiniBandDriverWindows \
  --publisher Microsoft.HpcCompute \
  --version 1.5
```

Add extension to a Virtual Machine Scale Set

The following example installs the latest version 1.5 InfiniBandDriverWindows extension on all RDMA-capable VMs in an existing virtual machine scale set named *myVMSS* deployed in the resource group named *myResourceGroup*:

PowerShell

```
$VMSS = Get-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "myVMSS"
Add-AzVmssExtension -VirtualMachineScaleSet $VMSS -Name "InfiniBandDriverWindows" -
Publisher "Microsoft.HpcCompute" -Type "InfiniBandDriverWindows" -TypeHandlerVersion
"1.5"
Update-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "MyVMSS" -
VirtualMachineScaleSet $VMSS
Update-AzVmssInstance -ResourceGroupName "myResourceGroup" -VMScaleSetName "myVMSS"
-InstanceId "*"
```

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using Azure PowerShell and Azure CLI. To see the deployment state of extensions for a given VM, run the following command.

PowerShell

```
Get-AzVMExtension -ResourceGroupName myResourceGroup -VMName myVM -Name
myExtensionName
```

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension execution output is logged to the following file. Refer to this file to track the status of the installation as well as for troubleshooting any failures.

Windows Command Prompt

```
C:\WindowsAzure\Logs\Plugins\Microsoft.HpcCompute.InfiniBandDriverWindows\
```

Exit codes

The following table describes the meaning and recommended action based on the exit codes of the extension installation process.

Expand table

Error Code	Meaning	Possible Action
0	Operation successful	
3010	Operation successful. Reboot required.	
100	Operation not supported or could not be completed.	Possible causes: PowerShell version not supported, VM size is not an InfiniBand-enabled VM, Failure downloading data. Check the log files to determine cause of error.
240, 840	Operation timeout.	Retry operation.
-1	Exception occurred.	Check the log files to determine cause of exception.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file a support incident through the Azure support site. For information about using Azure Support, read the Microsoft Azure support FAQ.

Next steps

For more information about InfiniBand-enabled ('r' sizes), see H-series and N-series VMs.

Last updated on 11/10/2025

Enable InfiniBand

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

RDMA capable HB-series and N-series VMs communicate over the low latency and high bandwidth InfiniBand network. The RDMA capability over such an interconnect is critical to boost the scalability and performance of distributed-node HPC and AI workloads. The InfiniBand enabled HB-series and N-series VMs are connected in a non-blocking fat tree with a low-diameter design for optimized and consistent RDMA performance.

There are various ways to enable InfiniBand on the capable VM sizes.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

VM Images with InfiniBand drivers

See VM Images for a list of supported VM Images on the Marketplace, which come pre-loaded with InfiniBand drivers (for SR-IOV or non-SR-IOV VMs) or can be configured with the appropriate drivers for RDMA capable VMs. The Ubuntu-HPC and AlmaLinux-HPC VM images in the marketplace are the easiest way to get started.

InfiniBand Driver VM Extensions

On Linux, the InfiniBandDriverLinux VM extension can be used to install the Mellanox OFED drivers and enable InfiniBand on the SR-IOV enabled HB-series and N-series VMs.

On Windows, the InfiniBandDriverWindows VM extension installs Windows Network Direct drivers (on non-SR-IOV VMs) or Mellanox OFED drivers (on SR-IOV VMs) for RDMA connectivity. In certain deployments of A8 and A9 instances, the HpcVmDrivers extension is added automatically. Note that the HpcVmDrivers VM extension is being deprecated; it will not be updated.

To add the VM extension to a VM, you can use Azure PowerShell cmdlets. For more information, see Virtual machine extensions and features. You can also work with extensions for

VMs deployed in the classic deployment model.

Manual installation

Mellanox OpenFabrics drivers (OFED) can be manually installed on the SR-IOV enabled HB-series and N-series VMs.

Linux

The OFED drivers for Linux can be installed with the example below. Though the example here is for RHEL, but the steps are general and can be used for any compatible Linux operating system such as Ubuntu (22.04, 24.04) and SLES (12 SP4+ and 15). More examples for other distros are on the azhpc-images repo. The inbox drivers also work as well, but the Mellanox OFED drivers provide more features.

Bash

```
MLNX_OFED_DOWNLOAD_URL=http://content.mellanox.com/ofed/MLNX_OFED-5.0-
2.1.8.0/MLNX_OFED_LINUX-5.0-2.1.8.0-rhel7.7-x86_64.tgz
# Optionally verify checksum
wget --retry-connrefused --tries=3 --waitretry=5 $MLNX_OFED_DOWNLOAD_URL
tar zxvf MLNX_OFED_LINUX-5.0-2.1.8.0-rhel7.7-x86_64.tgz

KERNEL=( $(rpm -q kernel | sed 's/kernel\-//g') )
KERNEL=${KERNEL[-1]}
# Uncomment the lines below if you are running this on a VM
#RELEASE=( $(cat /etc/redhat-release | awk '{print $4}') )
#yum -y install
http://olcentgbl.trafficmanager.net/redhat/${RELEASE}/updates/x86_64/kernel-
devel-${KERNEL}.rpm
sudo yum install -y kernel-devel-${KERNEL}
sudo ./MLNX_OFED_LINUX-5.0-2.1.8.0-rhel7.7-x86_64/mlnxofedinstall --kernel $KERNEL -
-kernel-sources /usr/src/kernels/${KERNEL} --add-kernel-support --skip-repo
```

Windows

For Windows, download and install the Mellanox OFED for Windows drivers.

Enable IP over InfiniBand (IB)

If you plan to run MPI jobs, you typically don't need IPoIB. The MPI library will use the verbs interface for IB communication (unless you explicitly use the TCP/IP channel of MPI library). But if you have an app that uses TCP/IP for communication and you want to run over IB, you can

use IPoIB over the IB interface. Use the following commands (for RHEL) to enable IP over InfiniBand.

> **Important**
>
> To avoid issues, ensure you aren't running older versions of Microsoft Azure Linux Agent (waagent). We recommend using at least version 2.4.0.2 before enabling IP over IB.

Bash

```
sudo sed -i -e 's/# OS.EnableRDMA=n/OS.EnableRDMA=y/g' /etc/waagent.conf
sudo systemctl restart waagent
```

Next steps

- Learn more about installing and running various supported MPI libraries on the VMs.
- Review the HBv3-series overview and HC-series overview.
- Read about the latest announcements, HPC workload examples, and performance results at the Azure Compute Tech Community Blogs.
- For a higher level architectural view of running HPC workloads, see High Performance Computing (HPC) on Azure.

Last updated on 12/03/2025

NVIDIA GPU Driver Extension for Linux

This extension installs NVIDIA GPU drivers on Linux N-series virtual machines (VMs). Depending on the VM family, the extension installs CUDA or GRID drivers. When you install NVIDIA drivers by using this extension, you're accepting and agreeing to the terms of the NVIDIA End-User License Agreement. During the installation process, the VM might reboot to complete the driver setup.

Instructions on manual installation of the drivers and the current supported versions are available. An extension is also available to install NVIDIA GPU drivers on Windows N-series VMs.

> **Note**
>
> With Secure Boot enabled, all OS boot components (boot loader, kernel, kernel drivers) must be signed by trusted publishers (key trusted by the system). Secure Boot is not supported using Windows or Linux extensions. For more information on manually installing GPU drivers with Secure Boot enabled, see Azure N-series GPU driver setup for Linux.

> **Note**
>
> The GPU driver extensions do not automatically update the driver after the extension is installed. If you need to move to a newer driver version then either manually download and install the driver or remove and add the extension again.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Operating system

This extension supports the following OS distros, depending on driver support for the specific OS version:

Expand table

Driver	Linux: Ubuntu	Linux: Red Hat Enterprise Linux
CUDA	20.04 LTS	7.9
GRID	22.04 LTS 24.04 LTS	8.2

> **Note**
>
> The latest supported CUDA drivers for NC-series VMs are currently 470.82.01. Later driver versions aren't supported on the K80 cards in NC. While the extension is being updated with this end of support for NC, install CUDA drivers manually for K80 cards on the NC-series.

> **Important**
>
> This document references a release version of Linux that is nearing or at, End of Life (EOL). Please consider updating to a more current version.

Internet connectivity

The Microsoft Azure Extension for NVIDIA GPU Drivers requires that the target VM is connected to the internet and has access.

Extension schema

The following JSON shows the schema for the extension:

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
```

```
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverLinux",
    "typeHandlerVersion": "1.6",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

Properties

Expand table

Name	Value/Example	Data type
apiVersion	2015-06-15	date
publisher	Microsoft.HpcCompute	string
type	NvidiaGpuDriverLinux	string
typeHandlerVersion	1.6	int

Settings

All settings are optional. The default behavior is to not update the kernel if not required for driver installation and install the latest supported driver and the CUDA toolkit (as applicable).

Expand table

Name	Description	Default value	Valid values	Data type
updateOS	Update the kernel even if not required for driver installation.	false	true, false	boolean
driverVersion	NV: GRID driver version. NC/ND: CUDA toolkit version. The latest drivers for the chosen CUDA are installed automatically.	latest	List of supported driver versions	string
installCUDA	Install CUDA toolkit. Only relevant for NC/ND series VMs.	true	true, false	boolean

Deployment

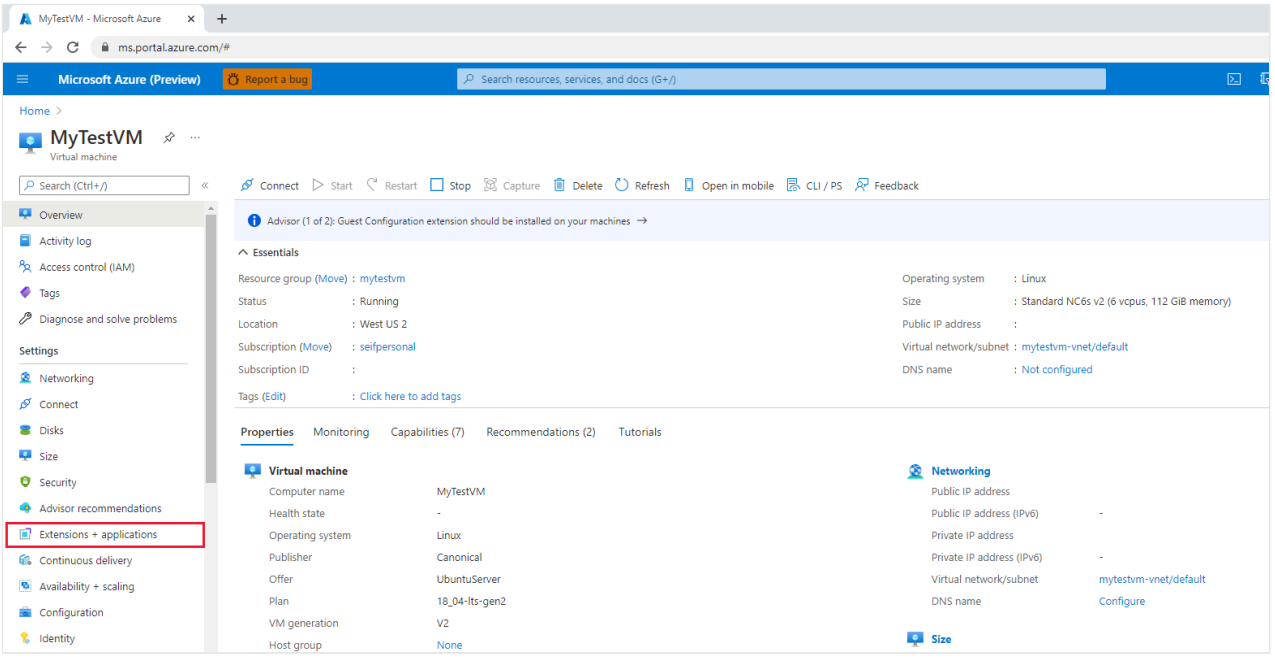
Azure portal

You can deploy Azure NVIDIA VM extensions in the Azure portal.

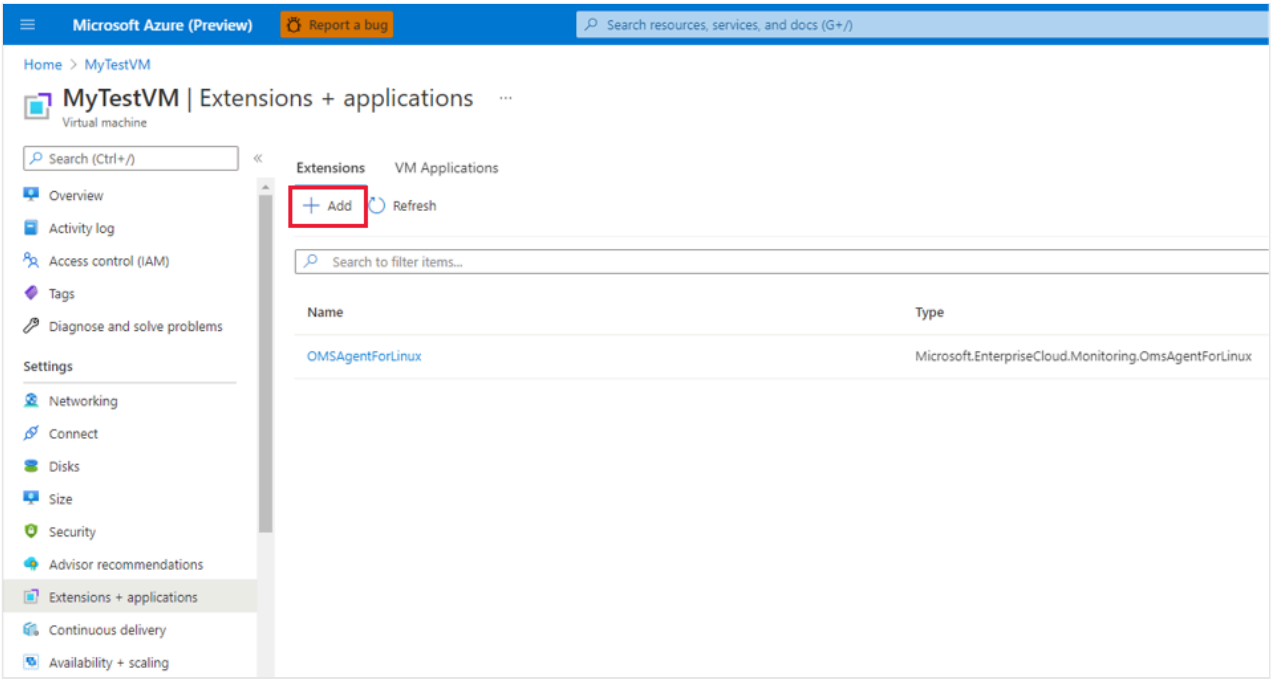
1. In a browser, go to the Azure portal.

2. Go to the virtual machine on which you want to install the driver.

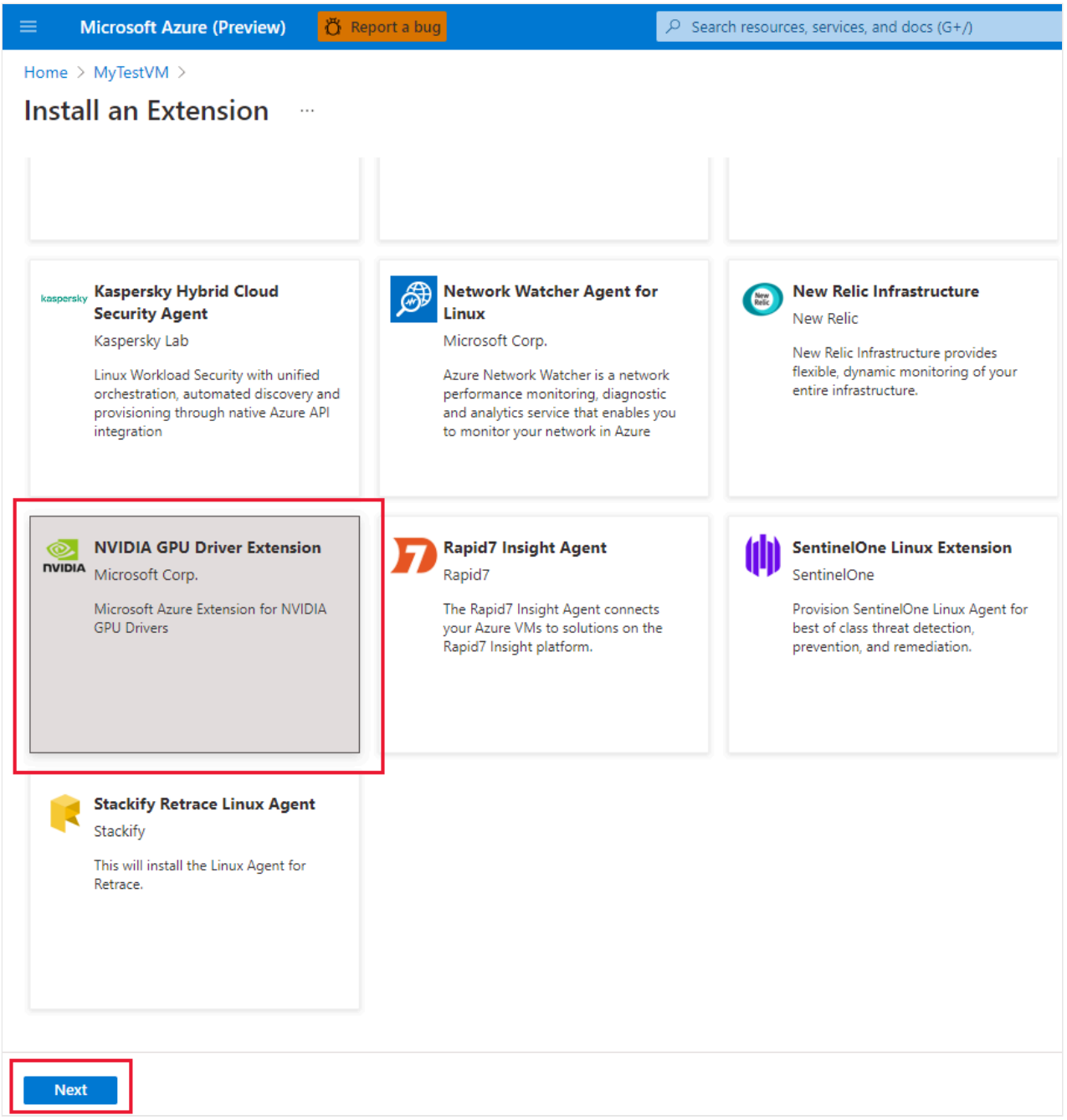
3. On the left menu, select **Extensions**.



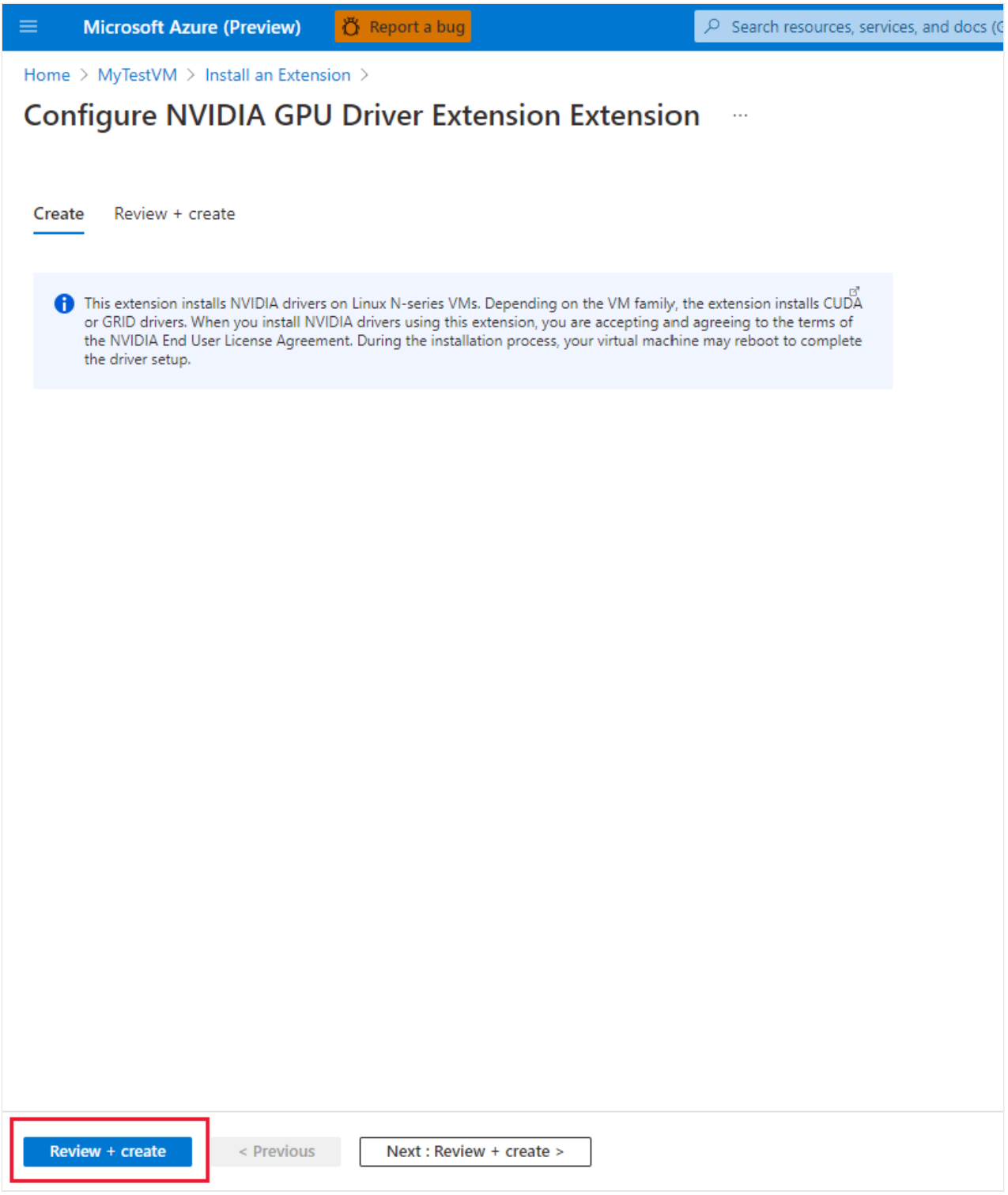
4. Select **Add**.



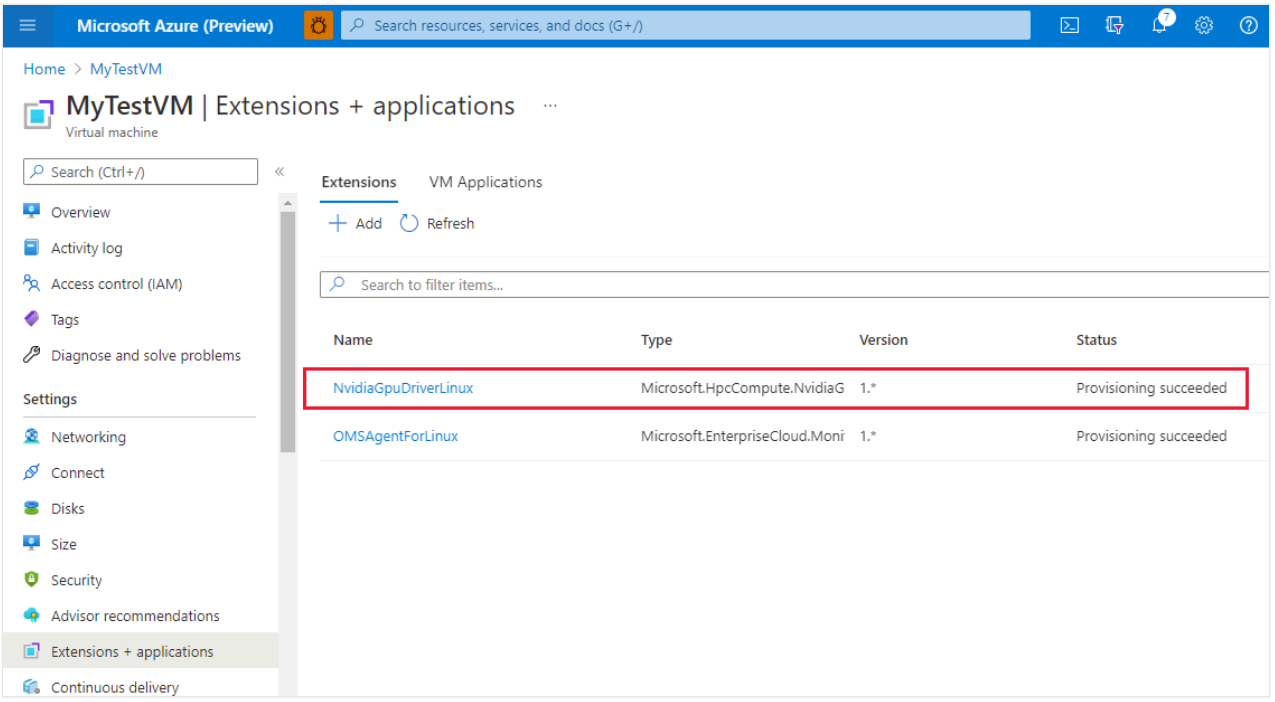
5. Scroll to find and select **NVIDIA GPU Driver Extension**, and then select **Next**.



6. Select **Review + create**, and select **Create**. Wait a few minutes for the driver to deploy.



7. Verify that the extension was added to the list of installed extensions.



Azure Resource Manager template

You can use Azure Resource Manager templates to deploy Azure VM extensions. Templates are ideal when you deploy one or more virtual machines that require post-deployment configuration.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the extension is nested inside the virtual machine resource. When the extension resource is nested, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
  "name": "myExtensionName",
  "type": "extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2015-06-15",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', myVM)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverLinux",
    "typeHandlerVersion": "1.6",
    "autoUpgradeMinorVersion": true,
```

```
    "settings": {
    }
  }
}
```

PowerShell

PowerShell

```
Set-AzVMExtension
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Location "southcentralus" `
    -Publisher "Microsoft.HpcCompute" `
    -ExtensionName "NvidiaGpuDriverLinux" `
    -ExtensionType "NvidiaGpuDriverLinux" `
    -TypeHandlerVersion 1.6 `
    -SettingString '{ `
    }'
```

Azure CLI

The following example mirrors the preceding Resource Manager and PowerShell examples:

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name NvidiaGpuDriverLinux \
  --publisher Microsoft.HpcCompute \
  --version 1.6
```

The following example also adds two optional custom settings as an example for nondefault driver installation. Specifically, it updates the OS kernel to the latest and installs a specific CUDA toolkit version driver. Again, note the `--settings` are optional and default. Updating the kernel might increase the extension installation times. Also, choosing a specific (older) CUDA toolkit version might not always be compatible with newer kernels.

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name NvidiaGpuDriverLinux \
  --publisher Microsoft.HpcCompute \
  --version 1.6 \
```

```
  --settings '{ \
    "updateOS": true, \
    "driverVersion": "10.0.130" \
  }'
```

Troubleshoot and support

Troubleshoot

You can retrieve data about the state of extension deployments from the Azure portal and by using Azure PowerShell and the Azure CLI. To see the deployment state of extensions for a given VM, run the following command:

PowerShell

```
Get-AzVMExtension -ResourceGroupName myResourceGroup -VMName myVM -Name
myExtensionName
```

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension execution output is logged to the following file. Refer to this file to track the status of any long-running installation and for troubleshooting any failures.

Bash

```
/var/log/azure/nvidia-vmext-status
```

Exit codes

Expand table

Exit code	Meaning	Possible action
0	Operation successful	
1	Incorrect usage of extension	Check the execution output log.
10	Linux Integration Services for Hyper-V and Azure not available or installed	Check the output of lspci.

Exit code	Meaning	Possible action
11	NVIDIA GPU not found on this VM size	Use a supported VM size and OS.
12	Image offer not supported	
13	VM size not supported	Use an N-series VM to deploy.
14	Operation unsuccessful	Check the execution output log.

Known issues

1. GRID driver 16.x and 17.x are having installation issues on Azure kernel 6.11. Nvidia is working on solving this issue, meanwhile, downgrade the Azure kernel to 6.8 by following these steps. Try to reinstall the drivers manually or by using an extension after downgrading the kernel to 6.8.

```
// Get the installed kernel. If kernel 6.11 is installed,  downgrade it to 6.8.
uname -a

// Install  kernel 6.8. Note that kernel  6.11  is not supported.
$ sudo apt install linux-image-6.8.0-1015-azure

// Get the list of installed kernels.
dpkg --list | egrep -i --color 'linux-image|linux-headers|linux-modules' | awk '{
print $2 }'

// Uninstall any 6.11 kernels.
sudo apt purge linux-headers-6.11.0-1013-azure  linux-image-6.11.0-1013-azure
linux-modules-6.11.0-1013-azure

// Run the following command to ensure only 6.8 images, headers, and modules are
installed and no other versions are present.
dpkg --list | egrep -i --color 'linux-image|linux-headers|linux-modules' | awk '{
print $2 }'

// Results from the previous command:
linux-headers-6.8.0-1015-azure
linux-image-6.8.0-1015-azure
linux-modules-6.8.0-1015-azure

// Open the grub settings and modify the GRUB_DEFAULT="0" to GRUB_DEFAULT="Advanced
options for Ubuntu>Ubuntu, with Linux 6.8.0-1015-azure".
$ sudo vim /etc/default/grub
```

```
// The grub file will look like the following:
GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 6.8.0-1015-azure"
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
//////////////////

// Update GRUB and reboot.
$ sudo update-grub && sudo update-grub2
$ sudo reboot

// Reinstall the driver after reboot.
```

2. `NvidiaGpuDriverLinux` currently installs the latest `17.5` GRID drivers, which is having issues with CUDA on A10 series. NVIDIA is working on solving this issue, meanwhile, use GRID driver `16.5` by passing a runtime setting to the extension.

Azure CLI

```
az vm extension set  --resource-group <rg-name> --vm-name <vm-name>  --name
NvidiaGpuDriverLinux --publisher Microsoft.HpcCompute --settings "
{'driverVersion':'535.161'}"
```

ARM

```
{
  "name": "NvidiaGpuDriverLinux",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverLinux",
    "typeHandlerVersion": "1.11",
    "autoUpgradeMinorVersion": true,
    "settings": {
         "driverVersion": "535.161"
    }
  }
}
```

3. GRID Driver version `17.x` is incompatible on NVv3 (NVIDIA Tesla M60). GRID drivers up to version `16.5` are supported. `NvidiaGpuDriverLinux` installs the latest drivers which are incompatible on NVv3 SKU. Instead, use the following runtime settings to force the

extension to install an older version of the driver. For more information on driver versions, see NVIDIA GPU resources.

Azure CLI

```
az vm extension set  --resource-group <rg-name> --vm-name <vm-name>  --name
NvidiaGpuDriverLinux --publisher Microsoft.HpcCompute --settings "
{'driverVersion':'535.161'}"
```

ARM

```
{
  "name": "NvidiaGpuDriverLinux",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverLinux",
    "typeHandlerVersion": "1.11",
    "autoUpgradeMinorVersion": true,
    "settings": {
         "driverVersion": "535.161"
    }
  }
}
```

4. Grid 17.5 linux driver has a bug where it impacts CUDA related workload. Error signature typically involves CUDA devices unavailable. While Azure is working to resolve this issue, use GRID driver 16.5 to continue running your workload.

Support

If you need more help at any point in this article, contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to Azure support and select **Get support**. For information about using Azure support, read the Azure support FAQ.

Next steps

- For more information about extensions, see Virtual machine extensions and features for Linux.
- For more information about N-series VMs, see GPU optimized virtual machine sizes.

Last updated on 12/02/2025

NVIDIA GPU Driver Extension for Windows

The NVIDIA GPU Driver Extension for Windows installs NVIDIA GPU drivers on Windows N-series virtual machines (VMs). Depending on the VM family, the extension installs CUDA or GRID drivers. When you install NVIDIA drivers by using this extension, you accept and agree to the terms of the NVIDIA End-User License Agreement. During the installation process, the VM might reboot to complete the driver setup.

The instructions for manual installation of the drivers, and the list of current supported versions are available for review. For more information, see Install NVIDIA GPU drivers on N-series VMs running Windows.

The NVIDIA GPU Driver Extension can also be deployed on Linux N-series VMs. For more information, see NVIDIA GPU Driver Extension for Linux.

> **Note**
>
> For NVadsA10v5-series VMs, this extension will continue to use v17.x until we finish the v18.x rollout in sovereign clouds in January 2026.

> **Note**
>
> The GPU driver extensions do not automatically update the driver after the extension is installed. If you need to move to a newer driver version then either manually download and install the driver or remove and add the extension again.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Confirm your virtual machine satisfies the prerequisites for using the NVIDIA GPU Driver Extension.

Operating system support

The NVIDIA GPU Driver Extension supports the following Windows versions:

Expand table

Distribution	Version
Windows 11	Core
Windows 10	Core
Windows Server 2022	Core
Windows Server 2019	Core
Windows Server 2016	Core
Windows Server 2012 R2	Core

Internet connection required

The NVIDIA GPU Driver Extension requires that the target VM is connected to the internet and has access.

Review the extension schema

The following JSON snippet shows the schema for the extension:

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverWindows",
    "typeHandlerVersion": "1.10",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

Properties

The JSON schema includes values for the following parameters.

Expand table

Name	Value/Example	Data type
`apiVersion`	2015-06-15	date
`publisher`	Microsoft.HpcCompute	string
`type`	NvidiaGpuDriverWindows	string
`typeHandlerVersion`	1.10	int

Deploy the extension

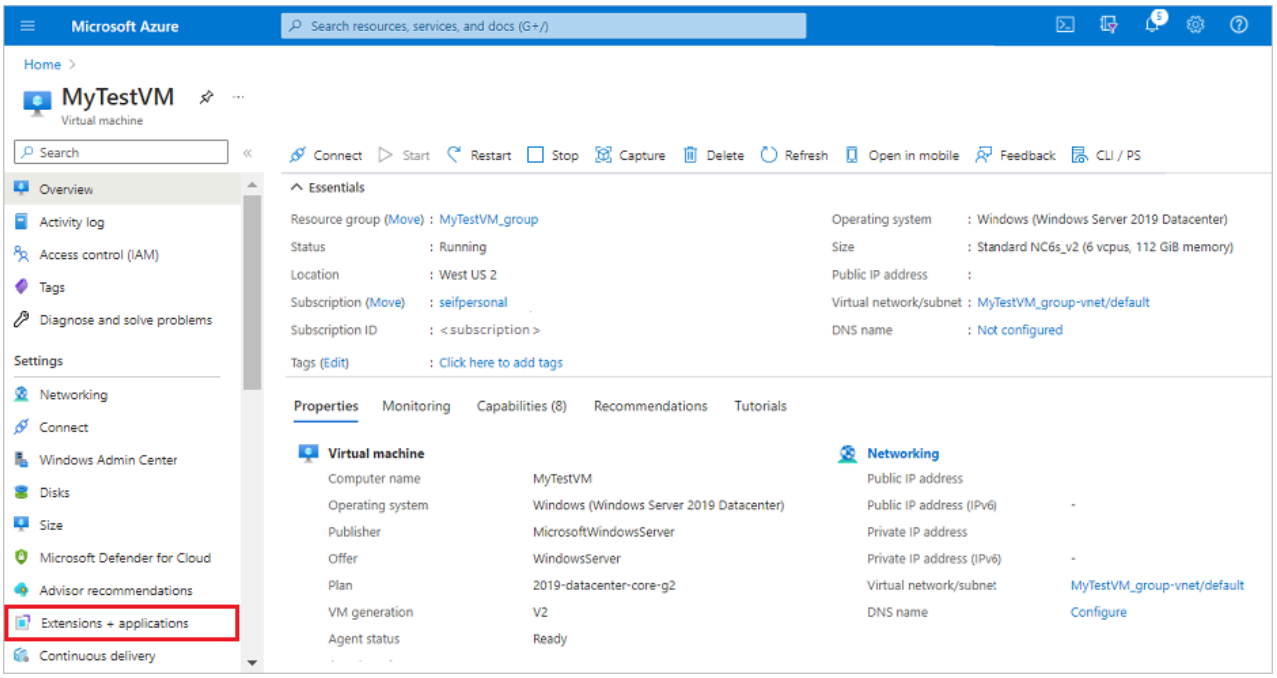
Azure VM extensions can be managed by using the Azure CLI, PowerShell, Azure Resource Manager (ARM) templates, and the Azure portal.

> **Note**
>
> Some of the following examples use `<placeholder>` parameter values in the commands. Before you run each command, make sure to replace any placeholder values with specific values for your configuration.

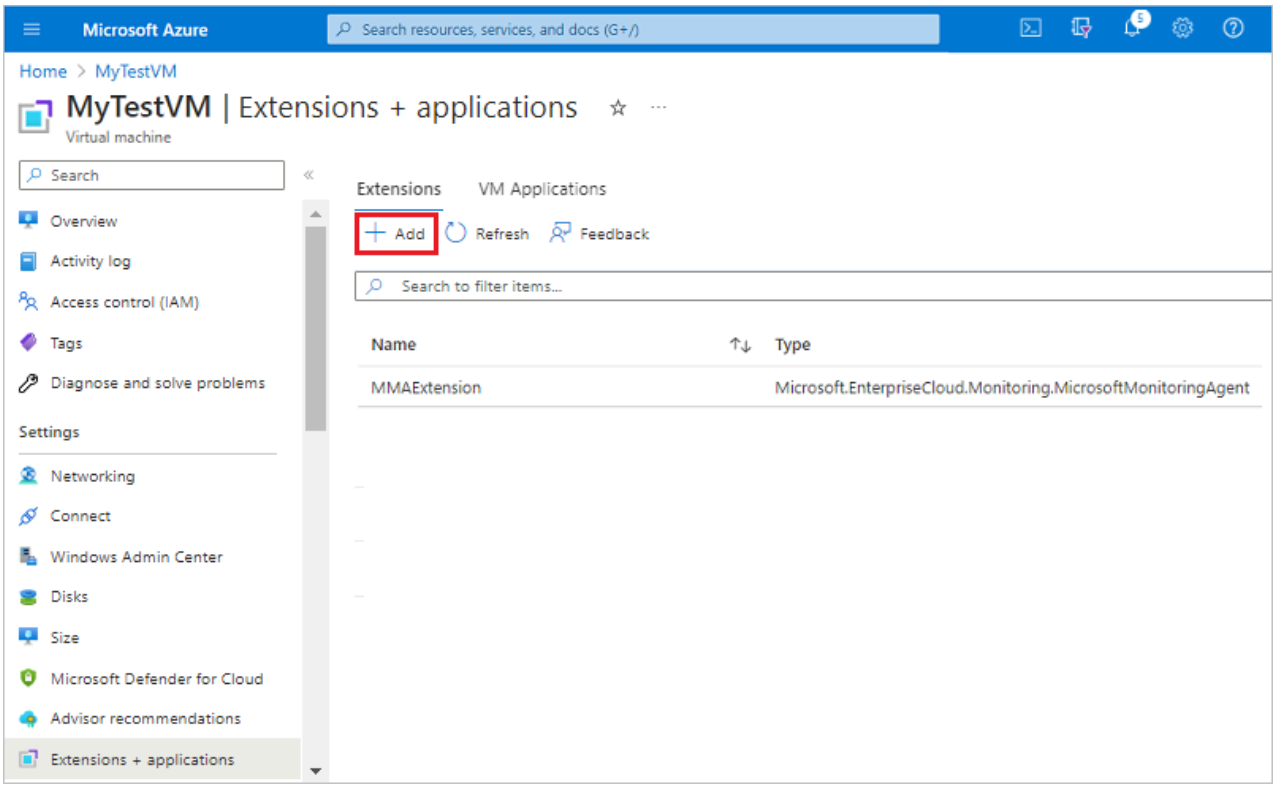
Azure portal

To install the NVIDIA GPU Driver Extension in the Azure portal, follow these steps:

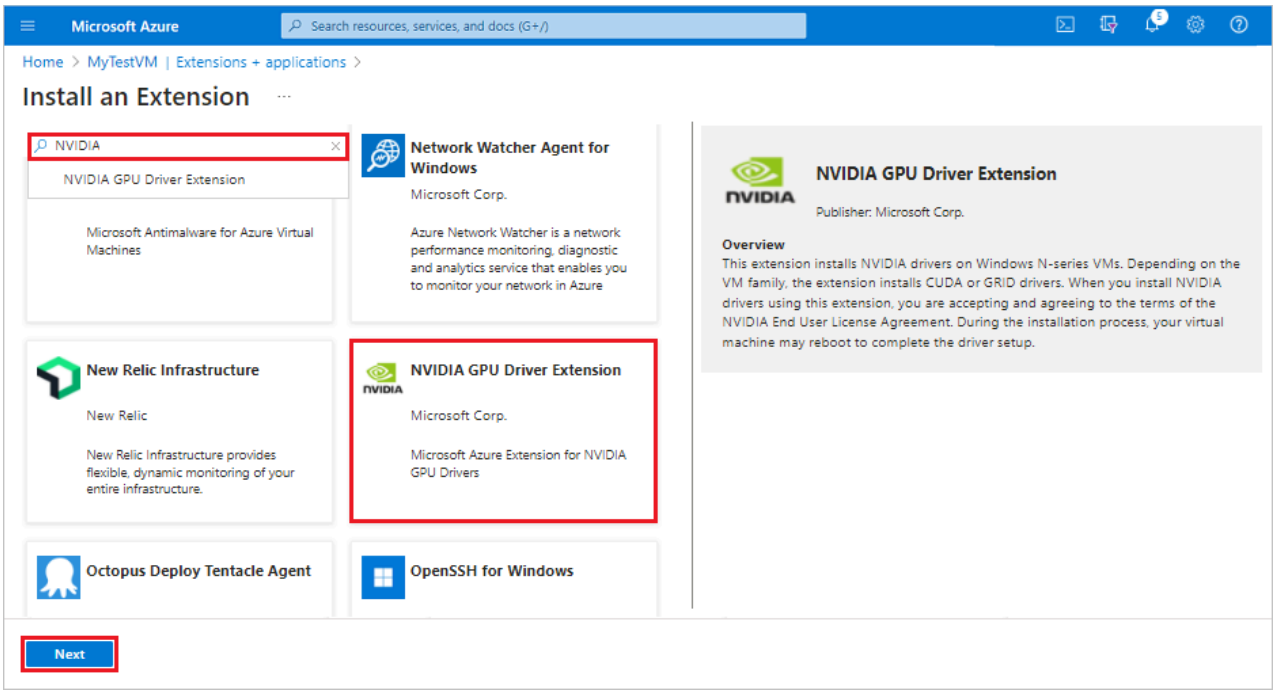
1. In the Azure portal, go to the virtual machine on which you want to install the extension.
2. Under **Settings**, select **Extensions + Applications**.



3. Under **Extensions**, select **+ Add**.

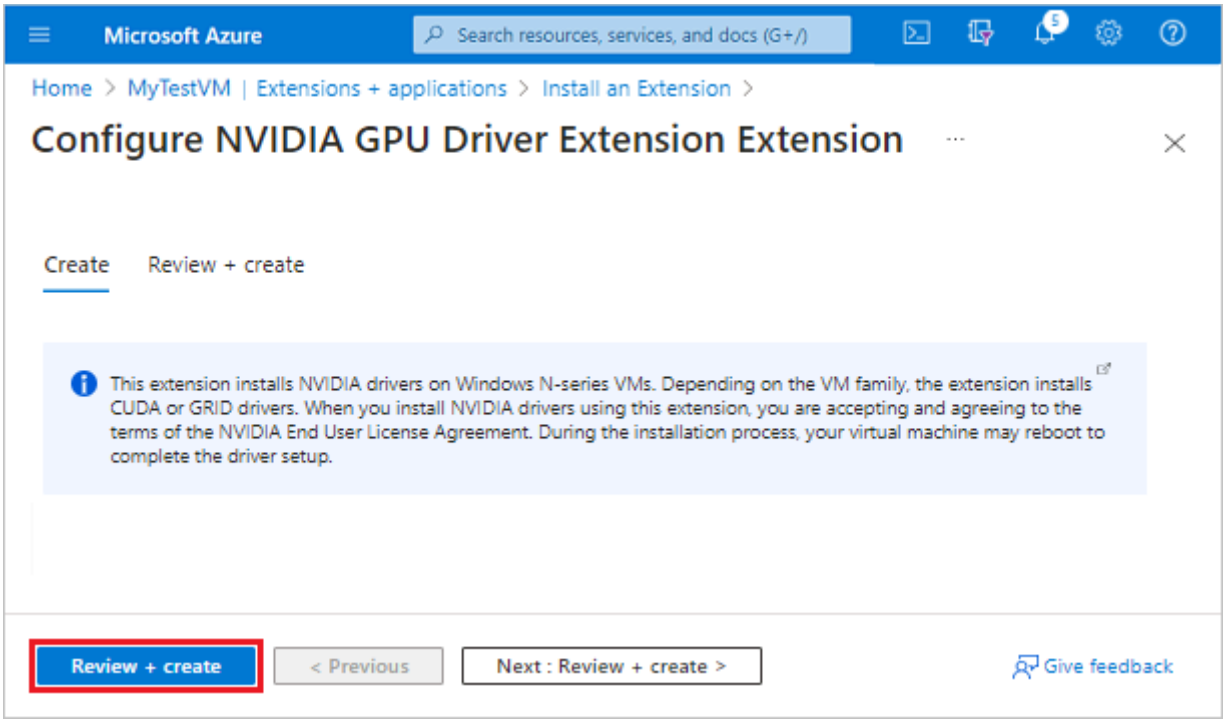


4. Locate and select **NVIDIA GPU Driver Extension**, then select **Next**.

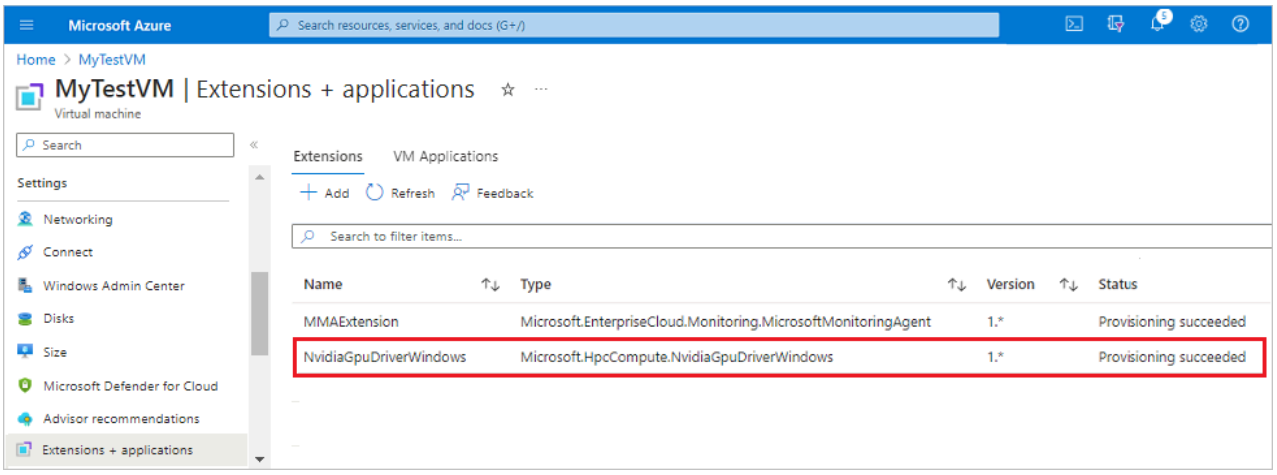


5. Select **Review + create**. Confirm the deployment action, and select **Create**.

   Wait a few minutes for the extension to deploy.



6. Confirm the extension is listed as an installed extension for the virtual machine.



ARM template

ARM templates are ideal when you deploy one or more virtual machines that require post-deployment configuration.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource or placed at the root or top level of a JSON ARM template. The placement of the JSON configuration affects the value of the resource `name` and `type`. For more information, see Set name and type for child resources.

The following example assumes the extension is nested inside the virtual machine resource. When the extension resource is nested, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "location": "[<resourceGroup().location>]",
  "apiVersion": "2015-06-15",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverWindows",
    "typeHandlerVersion": "1.10",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

PowerShell

Use the following PowerShell command to deploy the NVIDIA GPU Driver Extension to a virtual machine.

PowerShell

```
Set-AzVMExtension
    -ResourceGroupName "<myResourceGroup>" `
    -VMName "<myVM>" `
    -Location "<location>" `
    -Publisher "Microsoft.HpcCompute" `
    -ExtensionName "NvidiaGpuDriverWindows" `
    -ExtensionType "NvidiaGpuDriverWindows" `
    -TypeHandlerVersion 1.10 `
    -SettingString '{ `
    }'
```

Azure CLI

Run the following command in the Azure CLI to deploy the NVIDIA GPU Driver Extension to a virtual machine.

Azure CLI

```
az vm extension set \
  --resource-group <myResourceGroup> \
  --vm-name <myVM> \
  --name NvidiaGpuDriverWindows \
  --publisher Microsoft.HpcCompute \
  --version 1.10 \
  --settings '{ \
  }'
```

Troubleshoot issues

Here are some suggestions for how to troubleshoot deployment issues.

Check extension status

Check the status of your extension deployment in the Azure portal, or by using PowerShell or the Azure CLI.

To see the deployment state of extensions for a given VM, run the following commands:

PowerShell

```
Get-AzVMExtension -ResourceGroupName <myResourceGroup> -VMName <myVM> -Name
<myExtensionName>
```

Azure CLI

```
az vm extension list --resource-group <myResourceGroup> --vm-name <myVM> -o table
```

Review output logs

View output logs for the NVIDIA GPU Driver Extension deployment under `C:\WindowsAzure\Logs\Plugins\Microsoft.HpcCompute.NvidiaGpuDriverWindows\`.

Respond to error codes

The following table lists common error codes for deployment and potential follow-up actions.

Expand table

Error	Description	Action
0	Operation successful.	No required action.
1	Operation successful.	Reboot.
100	Operation not supported or couldn't be completed.	Check log files to determine cause of error, such as: - PowerShell version isn't supported. - VM size isn't an N-series VM. - Failure during data download.
240, 840	Operation timeout.	Retry operation.
-1	Exception occurred.	Check log files to determine cause of exception.
-5x	Operation interrupted due to pending reboot.	Reboot the VM. Installation continues after reboot. Uninstall should be invoked manually.

Known issues

GRID Driver version `17.x` is incompatible on NVv3 (NVIDIA Tesla M60). GRID drivers up to version `16.5` are supported. `NvidiaGpuDriverWindows` installs the latest drivers which are incompatible on NVv3 SKU. Instead, use the following runtime settings to force the extension

to install an older version of the driver. For more information on driver versions, see NVIDIA GPU resources.

Azure CLI

```
az vm extension set  --resource-group <rg-name> --vm-name <vm-name>  --name
NvidiaGpuDriverWindows --publisher Microsoft.HpcCompute --settings "
{'driverVersion':'538.46'}"
```

ARM

```
{
  "name": "NvidiaGpuDriverWindows",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "NvidiaGpuDriverWindows",
    "typeHandlerVersion": "1.10",
    "autoUpgradeMinorVersion": true,
    "settings": {
         "driverVersion": "538.46"
    }
  }
}
```

Get support

Here are some other options to help you resolve deployment issues:

- For assistance, contact the Azure experts on the Q&A and Stack Overflow forums.
- If you don't find an answer on the site, you can post a question for input from Microsoft or other members of the community.
- You can also Contact Microsoft Support. For information about using Azure support, read the Azure support FAQ.

Next steps

- For more information about extensions, see Virtual machine extensions and features for Windows.
- For more information about N-series VMs, see GPU optimized virtual machine sizes.

Last updated on 12/17/2025

AMD GPU Driver Extension for Windows

This article provides an overview of the virtual machine (VM) extension to deploy AMD GPU drivers on Windows N-series VMs. When you install AMD drivers by using this extension, you're accepting and agreeing to the terms of the AMD End-User License Agreement. During the installation process, the VM might reboot to complete the driver setup.

Instructions on manual installation of the drivers and the current supported versions are available. For more information, see Azure N-series AMD GPU driver setup for Windows.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

Internet connectivity

The Microsoft Azure Extension for AMD GPU Drivers requires that the target VM is connected to the internet and has access.

Extension schema

The following JSON shows the schema for the extension:

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "AmdGpuDriverWindows",
    "typeHandlerVersion": "1.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
```

```
  }
}
```

Properties

Expand table

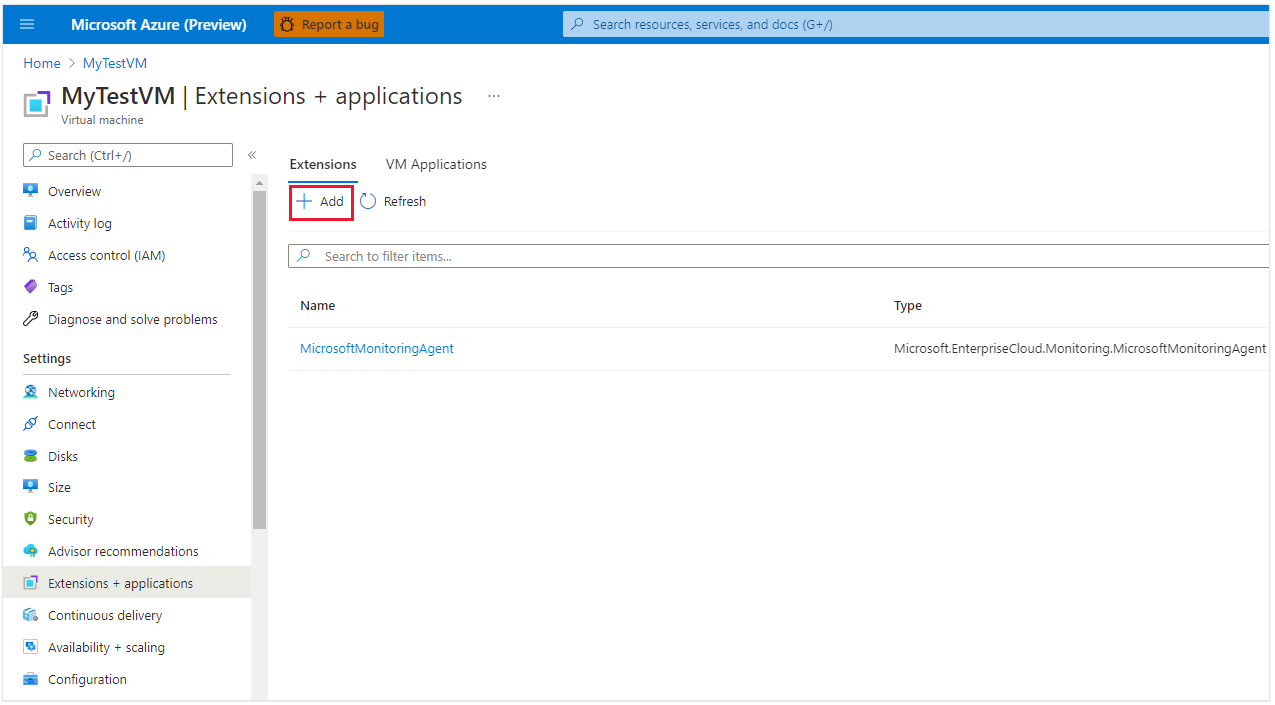
Name	Value/Example	Data type
apiVersion	2015-06-15	date
publisher	Microsoft.HpcCompute	string
type	AmdGpuDriverWindows	string
typeHandlerVersion	1.1	int

Deployment

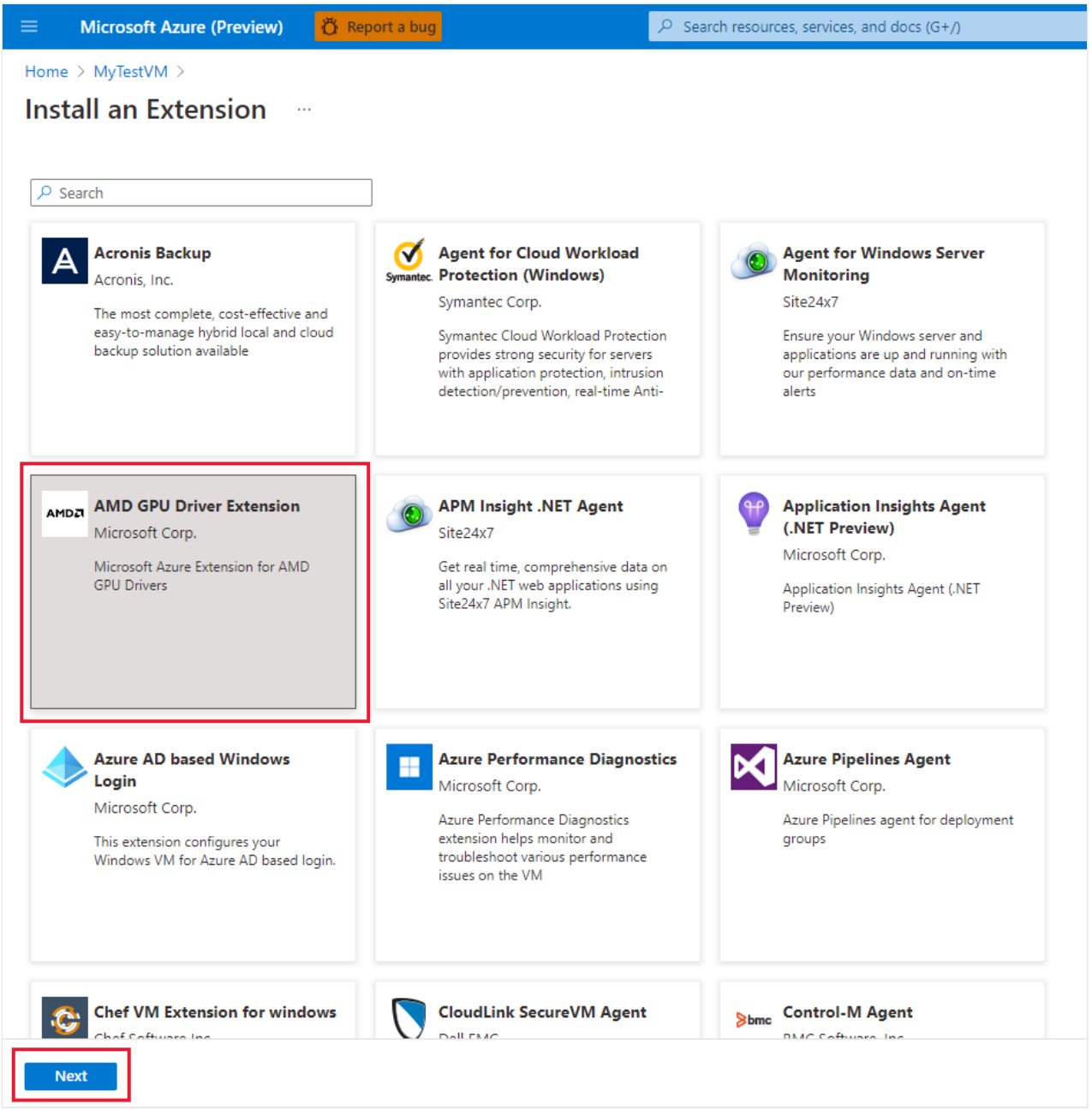
Azure portal

You can deploy Azure AMD VM extensions in the Azure portal.

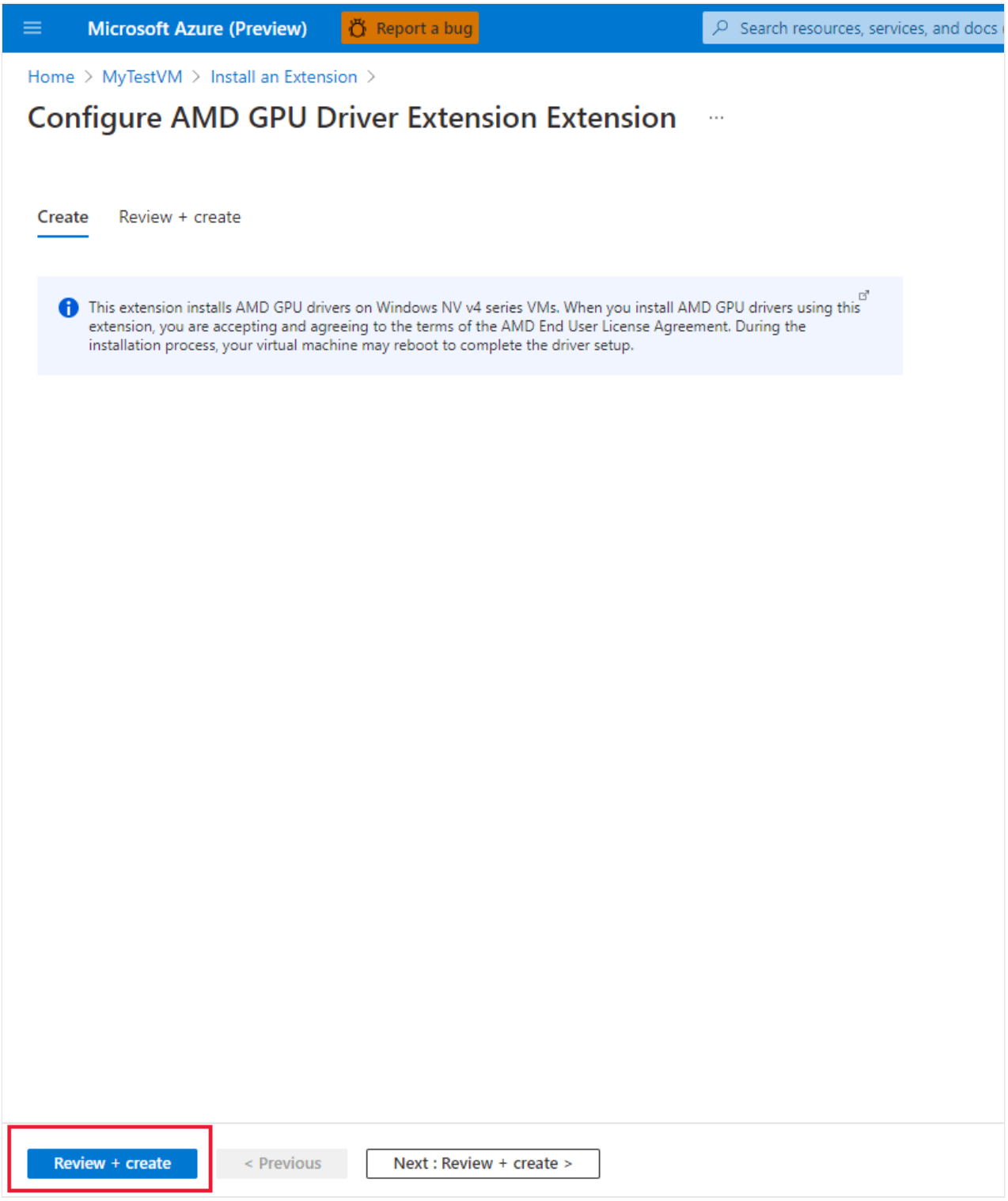
1. In a browser, go to the Azure portal.
2. Go to the virtual machine on which you want to install the driver.
3. On the left menu, select **Extensions**.
4. Select **Add**.



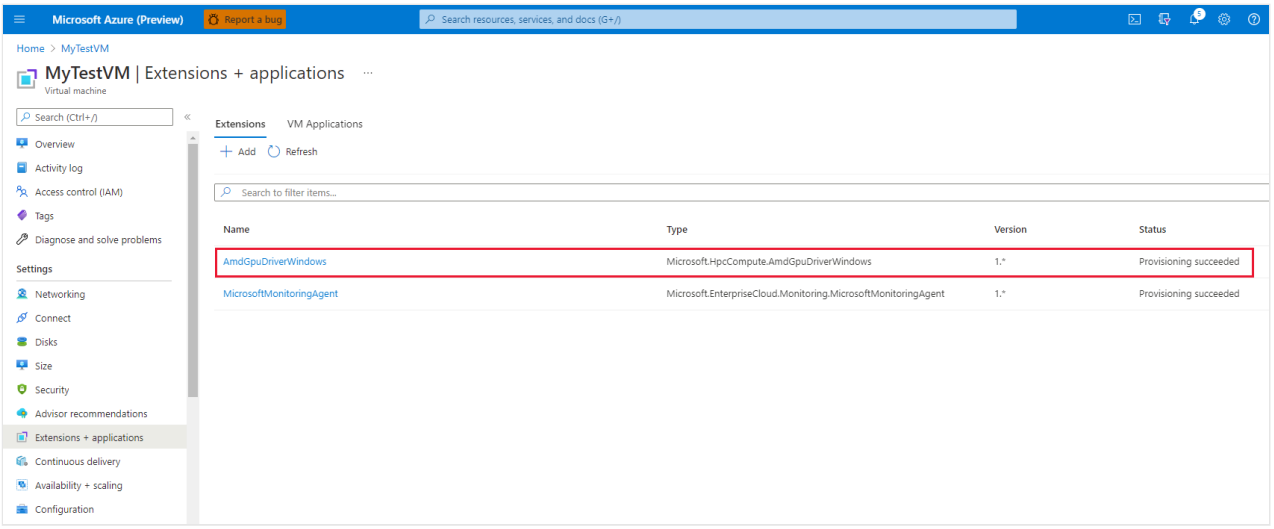
5. Scroll to find and select **AMD GPU Driver Extension**, and then select **Next**.



6. Select **Review + create**, and select **Create**. Wait a few minutes for the driver to deploy.



7. Verify that the extension was added to the list of installed extensions.



Azure Resource Manager template

You can use Azure Resource Manager templates to deploy Azure VM extensions. Templates are ideal when you deploy one or more virtual machines that require post-deployment configuration.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the extension is nested inside the virtual machine resource. When the extension resource is nested, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
  "name": "myExtensionName",
  "type": "extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2015-06-15",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', myVM)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "AmdGpuDriverWindows",
    "typeHandlerVersion": "1.1",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

PowerShell

PowerShell

```
Set-AzVMExtension `
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Location "southcentralus" `
    -Publisher "Microsoft.HpcCompute" `
    -ExtensionName "AmdGpuDriverWindows" `
    -ExtensionType "AmdGpuDriverWindows" `
    -TypeHandlerVersion 1.1 `
    -SettingString '{ `
    }'
```

Azure CLI

Azure CLI

```
az vm extension set `
  --resource-group myResourceGroup `
  --vm-name myVM `
  --name AmdGpuDriverWindows `
  --publisher Microsoft.HpcCompute `
  --version 1.1 `
  --settings '{ `
  }'
```

Troubleshoot and support

Troubleshoot

After extension is installed successfully, to run 'amd-smi' install WMIC using

PowerShell

```
DISM /Online /Add-Capability /CapabilityName:WMIC
```

You can retrieve data about the state of extension deployments from the Azure portal and by using Azure PowerShell and the Azure CLI. To see the deployment state of extensions for a given VM, run the following command:

PowerShell

```
Get-AzVMExtension -ResourceGroupName myResourceGroup -VMName myVM -Name
myExtensionName
```

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension execution output is logged to the following directory:

Windows Command Prompt

```
C:\WindowsAzure\Logs\Plugins\Microsoft.HpcCompute.AmdGpuDriverMicrosoft\
```

Error codes

Expand table

Error Code	Meaning	Possible action
0	Operation successful.	
1	Operation successful. Reboot required.	
100	Operation not supported or couldn't be completed.	Possible causes are that the PowerShell version isn't supported, the VM size isn't an N-series VM, and a failure occurred in downloading data. Check the log files to determine the cause of the error.
240, 840	Operation timeout.	Retry operation.
-1	Exception occurred.	Check the log files to determine the cause of the exception.
-5x	Operation interrupted due to pending reboot.	Reboot VM. Installation continues after the reboot. Uninstall should be invoked manually.

Support

If you need more help at any point in this article, contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to Azure support and select **Get support**. For information about using Azure support, read the Azure support FAQ.

Next steps

- For more information about extensions, see Virtual machine extensions and features for Windows.
- For more information about N-series VMs, see GPU optimized virtual machine sizes.

Last updated on 11/10/2025

AMD GPU Driver Extension for Linux

This extension installs AMD GPU drivers on Linux N-series virtual machines (VMs). When you install AMD drivers by using this extension, you're accepting and agreeing to the terms of the AMD End-User License Agreement. During the installation process, the VM might reboot to complete the driver setup.

Instructions on manual installation of the drivers and the current supported versions are available. An extension is also available to install AMD GPU drivers on Linux N-series VMs.

> **Note**
>
> With Secure Boot enabled, all OS boot components, including the boot loader, kernel, and kernel drivers, must be signed by trusted publishers whose keys are trusted by the system. For more information on manually installing GPU drivers with Secure Boot enabled, see Azure N-series GPU driver setup for Linux.
>
> The GPU driver extensions do not automatically update the driver once the extension is installed. To upgrade to a newer driver version, you will need to either uninstall and reinstall the extension or manually install the driver.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

Operating system

This extension supports the following OS distros, depending on driver support for the specific OS version:

Expand table

Distribution	Version
Linux: Ubuntu	22.04 24.04

> **Note**
>
> For installation instructions on other Linux distributions, please visit AMD's documentation

Internet connectivity

The AMD GPU Drivers Extension requires that the target VM is connected to the internet and has access.

Review the extension schema

The following JSON snippet shows the schema for the extension:

JSON

```
{
  "name": "<myExtensionName>",
  "type": "extensions",
  "apiVersion": "2015-06-15",
  "location": "<location>",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', <myVM>)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "AmdGpuDriverLinux",
    "typeHandlerVersion": "1.0",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

Properties

The JSON schema includes values for the following parameters.

Expand table

Name	Value/Example	Data type
apiVersion	2015-06-15	date
publisher	Microsoft.HpcCompute	string

Name	Value/Example	Data type
type	AmdGpuDriverLinux	string
typeHandlerVersion	1.0	int

Settings

All settings are optional. The default behavior is to not update the kernel if not required for driver installation and install the latest supported driver.

Expand table

Name	Default value	Valid values	Data type
driverVersion	Latest	List of supported driver versions	string

Deploy the extension

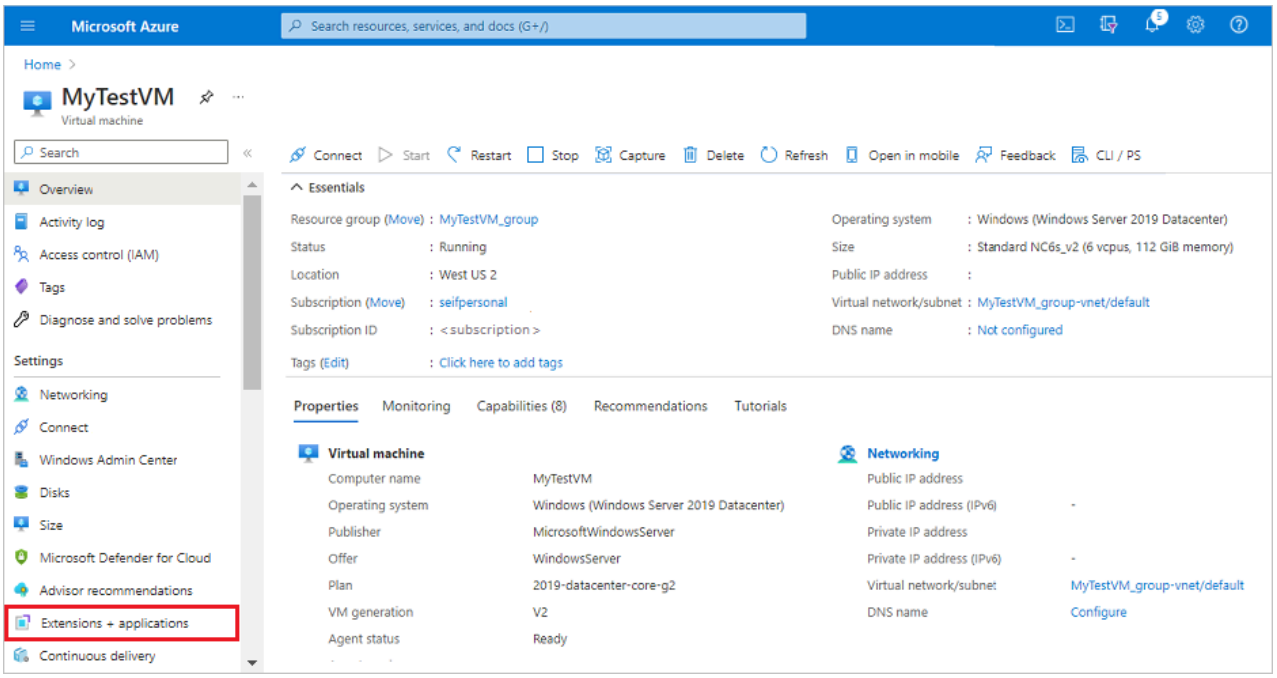
Azure VM extensions can be managed by using the Azure CLI, PowerShell, Azure Resource Manager (ARM) templates, and the Azure portal.

Azure portal

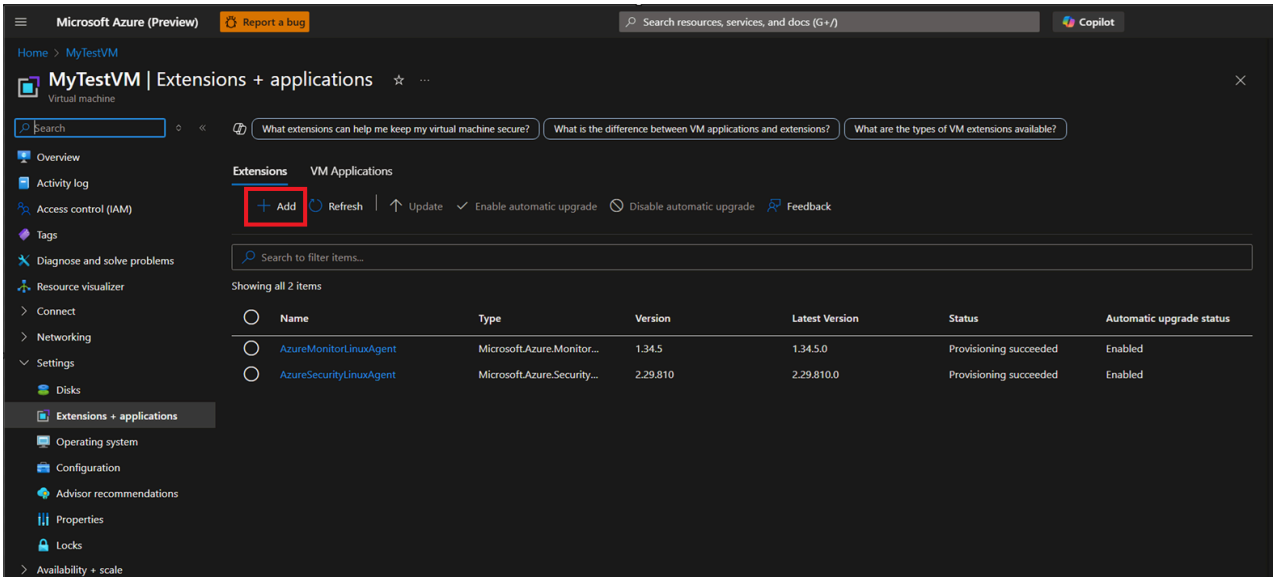
To install the Azure AMD VM extension in the Azure portal, follow these steps:

1. In the Azure portal, go to the virtual machine on which you want to install the extension.

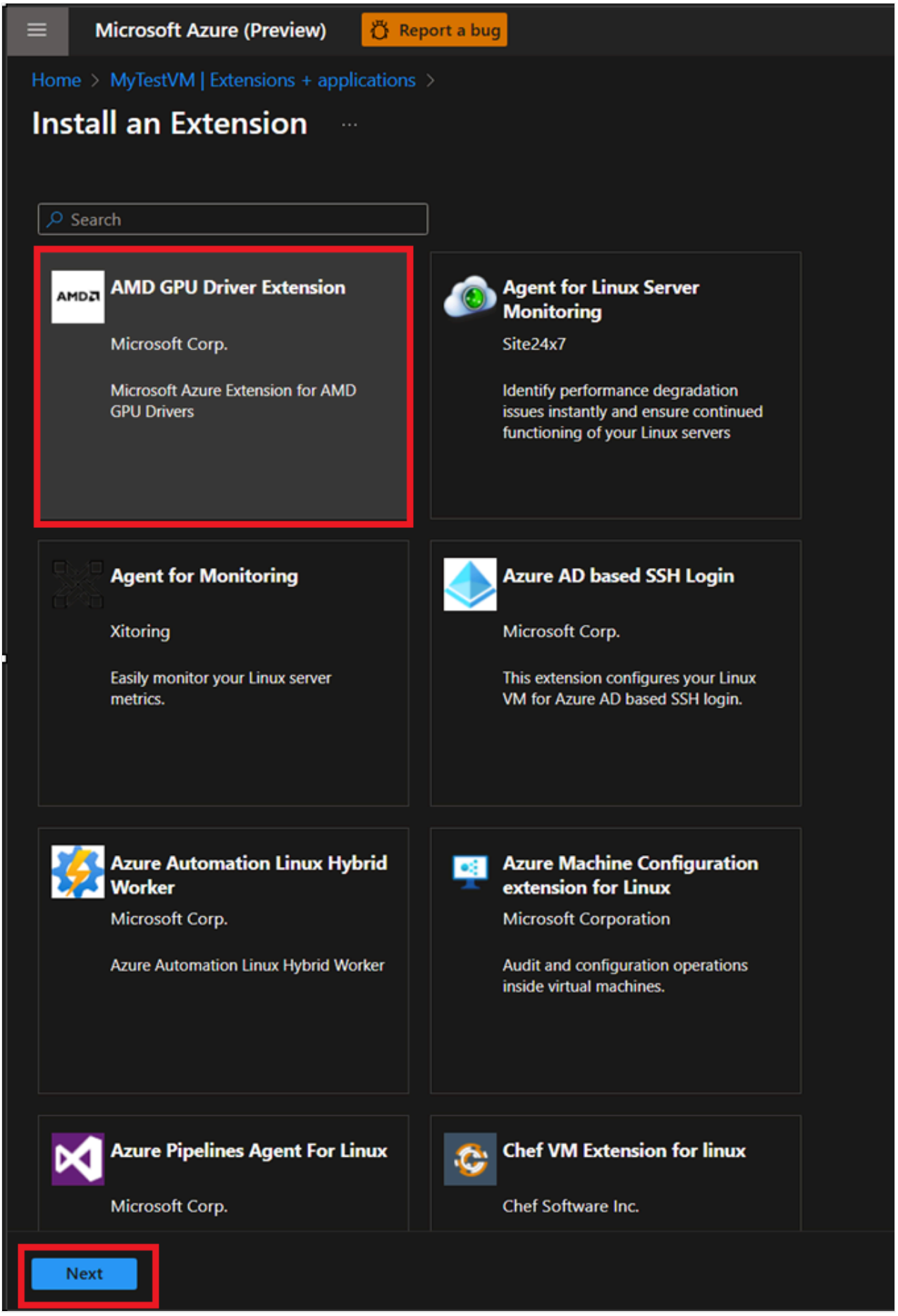
2. Under **Settings**, select **Extensions + Applications**.



3. Under **Extensions**, select **+ Add**.

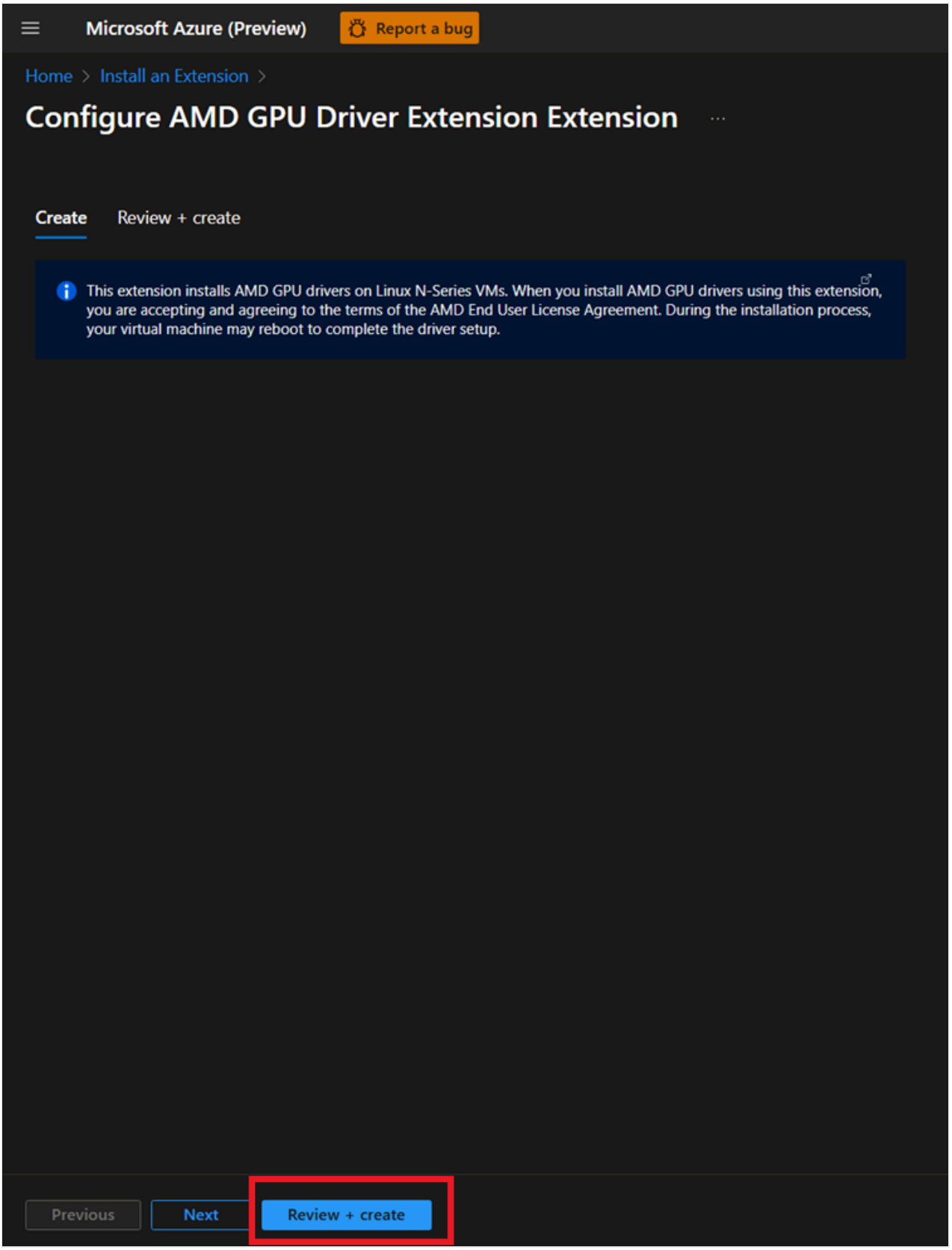


4. Scroll to find and select **AMD GPU Driver Extension**, and then select **Next**.

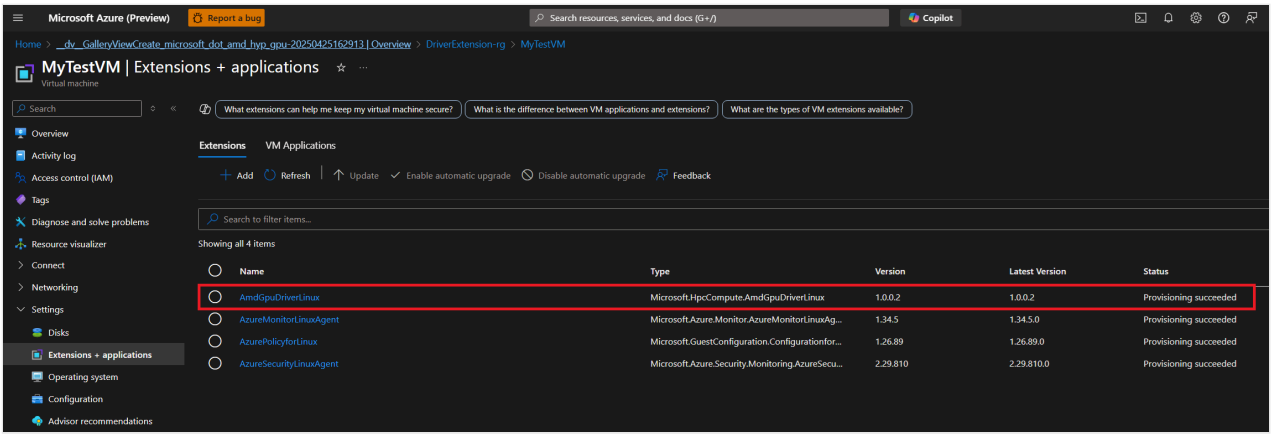


5. Select **Review + create**, and select **Create**.

   Wait a few minutes for the driver to deploy.



6. Confirm the extension is listed as an installed extension for the virtual machine.



Azure Resource Manager template

You can use Azure Resource Manager templates to deploy Azure VM extensions. Templates are ideal when you deploy one or more virtual machines that require post-deployment configuration.

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the extension is nested inside the virtual machine resource. When the extension resource is nested, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
  "name": "myExtensionName",
  "type": "extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2015-06-15",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', myVM)]"
  ],
  "properties": {
    "publisher": "Microsoft.HpcCompute",
    "type": "AmdGpuDriverLinux",
    "typeHandlerVersion": "1.0",
    "autoUpgradeMinorVersion": true,
    "settings": {
    }
  }
}
```

PowerShell

PowerShell

```
Set-AzVMExtension
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Location "southcentralus" `
    -Publisher "Microsoft.HpcCompute" `
    -ExtensionName "AmdGpuDriverLinux" `
    -ExtensionType "AmdGpuDriverLinux" `
    -TypeHandlerVersion 1.0 `
    -SettingString '{}'
```

Azure CLI

The following example mirrors the preceding Resource Manager and PowerShell examples:

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name AmdGpuDriverLinux \
  --publisher Microsoft.HpcCompute \
  --version 1.0
```

The following example also adds two optional custom settings as an example for nondefault driver installation.

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name AmdGpuDriverLinux \
  --publisher Microsoft.HpcCompute \
  --version 1.0 \
  --settings '{ \
    "driverVersion": "6.3.3" \
  }'
```

Troubleshooting

Check extension status

Check the status of your extension deployment in the Azure portal, or by using PowerShell or the Azure CLI.

To see the deployment state of extensions for a given VM, run the following commands:

PowerShell

```
Get-AzVMExtension -ResourceGroupName myResourceGroup -VMName myVM -Name
myExtensionName
```

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Review output logs

View output logs for the Azure AMD VM extension deployment. Refer to this file to track the status of any long-running installation and for troubleshooting any failures.

Bash

```
/var/log/azure/amd-vmext-status
```

Respond to exit codes

The following table lists common exit codes for deployment and potential follow-up actions.

Expand table

Exit code	Meaning	Possible action
0	Operation successful	No required action.
1	Incorrect usage of extension	Check the execution output log.
2	Python not found	Check the execution output log.
10	Linux Integration Services for Hyper-V and Azure not available or installed	Check the output of lspci.
11	AMD GPU not found on this VM size	Use a supported VM size and OS..
14	DPKG frontend (`/var/lib/dpkg/lock-frontend`) is locked by another process	Please try reinstalling after sometime.

Exit code	Meaning	Possible action
15	DPKG (`/var/lib/dpkg/lock`) is locked by another process	Please try reinstalling after sometime.
17	Failed to download the driver	Check the execution output log.
18	Failed to download the driver	Check the execution output log.
19	Failed to install the driver	Check the execution output log.
20	Insufficient disk space	Check the execution output log.
21	Incompatible kernel	Check kernel validity here AMD-GPU-Linux-Resources.
22	Compatibility check failed	Check kernel validity here AMD-GPU-Linux-Resources.
23	Required variable is not set	Check the execution output log.

Next steps

- For more information about extensions, see Virtual machine extensions and features for Linux
- For more information about N-series VMs, see GPU optimized virtual machine sizes.

Last updated on 11/10/2025

Install and manage the Azure Monitor Agent

This article describes the different methods you can use to install, uninstall, update, and configure the Azure Monitor Agent on Azure virtual machines, virtual machine scale sets, and Azure Arc-enabled servers.

Prerequisites

For prerequisites and other requirements for using the Azure Monitor Agent, see these articles:

- Azure Monitor Agent supported operating systems and environments
- Azure Monitor Agent requirements
- Azure Monitor Agent network configuration

> **Important**
>
> If the agent is going to connect to the Log Analytics workspace using Azure private link, see **Enable private link for monitoring virtual machines and Kubernetes clusters in Azure Monitor**.

> **Important**
>
> Installing, upgrading, or uninstalling the Azure Monitor Agent doesn't require a machine restart.

Installation options

The following table lists the options for installing the Azure Monitor Agent on Azure VMs and Azure Arc-enabled servers.

For any machine that isn't in Azure, the Azure Arc agent must be installed on the machine before the Azure Monitor Agent can be installed.

Expand table

Installation method	Description
Virtual machine (VM) extension	Use any of the methods described in this article to install the agent via the Azure extension framework. This method doesn't create a DCR, so you must create at least one DCR and associate it with the agent before data collection begins.
Create a DCR	When you create a DCR in the Azure portal, the Azure Monitor Agent is installed on any machine that's added as a resource for the DCR. The agent immediately begins to collect data as defined in the DCR.
VM insights	When you enable VM insights on a machine, the Azure Monitor Agent is installed and a DCR is created to collect a predefined set of data. You shouldn't modify this DCR, but you can create more DCRs to collect more data.
Container insights	When you enable collection of Prometheus metrics or container logs on a Kubernetes cluster, a containerized version of the Azure Monitor Agent is installed in the cluster and a DCR is created to immediately begin collecting data. You can modify the DCR by using the guidance in Configure data collection and cost optimization in Container insights by using DCRs.
Client installer	Install the agent by using a Windows MSI installer for Windows 11 and Windows 10 clients.
Azure Policy	Use Azure Policy to automatically install the agent on Azure virtual machines and Azure Arc-enabled servers, and to automatically associate them with required DCRs.

> **Note**
>
> - To send data across tenants, you must first enable Azure Lighthouse.
> - Cloning a machine that has Azure Monitor Agent installed isn't supported. The best practice for this scenario is to use Azure Policy or an infrastructure as a code (IaaC) tool to deploy the Azure Monitor Agent at scale.

Install the agent extension

Azure portal

To install the agent and create a DCR to collect data by using the Azure portal, use the guidance in Collect data by using the Azure Monitor Agent.

Uninstall

Azure portal

To uninstall the Azure Monitor Agent by using the Azure portal, go to your virtual machine, scale set, or Azure Arc-enabled server. Select the **Extensions** tab, and then select **AzureMonitorWindowsAgent** or **AzureMonitorLinuxAgent**. In the dialog that opens, select **Uninstall**.

Update

> **Note**
>
> We strongly recommend that you always update to the latest version of the agent or opt in to automatic extension upgrade.
>
> Automatic extension rollout follows standard Azure deployment practices to safely deploy the latest version of the agent. You should expect automatic updates to take weeks to roll out the latest version.
>
> Upgrades are issued in batches, so some of your virtual machines, scale sets, or Azure Arc-enabled servers might be upgraded before others.
>
> If you need to upgrade an extension immediately, you can use the manual instructions that are described in this article. Only agents released in the past year are supported.

Azure portal

To do a one-time update of the agent, go to your virtual machine or scale set. Select the **Extensions** tab, then check the agent and click **Update**.

We recommend that you enable automatic update of the agent by enabling automatic extension upgrade. Go to your virtual machine or scale set, select the **Extensions** tab, and then select **AzureMonitorWindowsAgent** or **AzureMonitorLinuxAgent**. In the dialog that opens, select **Enable automatic upgrade**.

Configure (preview)

Data collection rules (DCRs) serve as a management tool for the Azure Monitor Agent on your machine. The AgentSettings DCR can be used to configure certain Azure Monitor Agent

parameters to configure the agent to your specific monitoring needs.

> **Note**
>
> Important considerations when you work with the AgentSettings DCR:
>
> - Currently, the AgentSettings DCR can be configured only by using an Azure Resource Manager template.
> - AgentSettings must be a single DCR with no other settings.
> - The virtual machine and the AgentSettings DCR must be in the same region.

Supported parameters

The AgentSettings DCR currently supports setting the following parameters:

Expand table

Parameter	Description	Valid values
`MaxDiskQuotaInMB`	To provide resiliency, the agent collects data in a local cache when the agent can't send data. The agent sends the data in the cache after the connection is restored. This parameter is the amount of disk space used (in MB) by the Azure Monitor Agent log files and cache.	Linux: `4,000` to `1,000,000` `default: 10,000` Windows: `4,000` to `1,000,000`
`UseTimeReceivedForForwardedEvents`	Changes the **WEF** column in the Microsoft Sentinel Windows Event Forwarding (WEF) table to use `TimeReceived` instead of `TimeGenerated` data	`0` or `1`

Set up the AgentSettings DCR

Azure portal

Currently not supported.

Related content

Create a data collection rule to collect data from the agent and send it to Azure Monitor.

Last updated on 02/18/2026

Azure Diagnostics extension overview

Azure Diagnostics extension is an agent in Azure Monitor that collects monitoring data from the guest operating system of Azure compute resources including virtual machines. This article provides an overview of Azure Diagnostics extension, the specific functionality that it supports, and options for installation and configuration.

ⓘ Important

Migrate from Azure Diagnostic extension

Azure Diagnostics extension was deprecated on **March 31, 2026** and is no longer supported. Don't use new deployments of the extension.

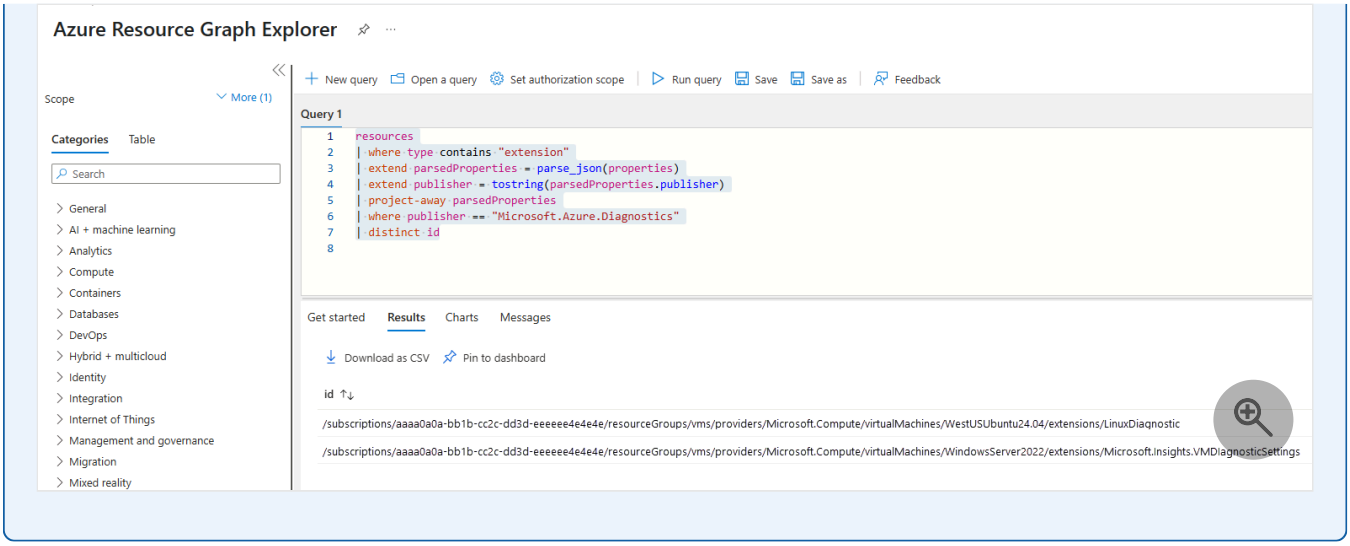
To ensure continued support and access to new features, migrate from Azure Diagnostics extensions for Linux (LAD) and Windows (WAD) to alternative solutions following the migration guidance. Remove LAD or WAD after you configure Azure Monitor Agent to avoid duplicate data.

To check which extensions are installed on a single VM, select **Extensions + applications** under **Settings** on your VM. To review the extensions installed on all virtual machines in subscriptions where you have access, use the following query in Azure Resource Graph:

kql

```
resources
| where type contains "extension"
| extend parsedProperties = parse_json(properties)
| extend publisher = tostring(parsedProperties.publisher)
| project-away parsedProperties
| where publisher == "Microsoft.Azure.Diagnostics"
| distinct id
```

It produces results similar to the following example:



Migration guidance

The Azure Diagnostics extension was deprecated on March 31, 2026. To ensure continued support and access to new features, migrate using the following options based on the data destination:

Expand table

Destination	Migration options
Azure Storage blobs	If you're using WAD or LAD agents to send data to storage for longer term storage and lower costs, migrate to Azure Monitor Agent. Then you can send data to custom tables with low-cost Auxiliary plan for cost-effective logging and added benefits of Log Analytics
Azure Event Hubs	If you're using WAD or LAD agents to send data to Event Hubs as a way to land it in your final destination or third party products, consider the following methods now available natively using Azure Monitor: • Configure Event Hubs using VM watch A native offering for virtual machines (VMs) and scale sets that send data Azure, including Event Hubs. **Note**: VM Watch doesn't collect application logs • **Event Hub -> Azure Data Explorer**: If your final data destination is ADX, migrate to Azure Monitor Agent to send data directly to ADX and Fabric eventhouses as a simpler, more reliable solution • **Event Hub -> Other destination(s)**: Contact Azure Monitor team for quick assistance regarding other destinations not listed here
Azure Monitor metrics	For VM Guest OS Perf Counter scenarios, migrate to using AMW as a destination for OpenTelemetry performance counters. For custom metric scenarios, migrate to using AMW as a destination for OpenTelemetry metrics.

Primary scenarios

Use the Azure Diagnostics extension if you need to:

- Send data to Azure Storage for archiving or to analyze it by using tools such as Azure Storage Explorer.
- Send data to Azure Monitor Metrics to analyze it by using metrics explorer and to take advantage of features such as near-real-time metric alerts and autoscale (Windows only).
- Send data to third-party tools by using Azure Event Hubs.
- Collect boot diagnostics to investigate VM boot problems. Boot diagnostics is a native VM feature and doesn't require the Azure Diagnostics extension. Use Microsoft-managed storage for boot diagnostics to avoid dependencies on customer-managed storage accounts and SAS tokens.

Limitations of Azure Diagnostics extension:

- It's deprecated on March 31, 2026.
- It works only with Azure resources.
- It has limited ability to send data to Azure Monitor Logs.

Costs

There's no cost for Azure Diagnostics extension, but you might incur charges for the data ingested. Check Azure Monitor pricing for the destination where you're collecting data.

Data collected

The following tables list the data that can get collected by the Windows and Linux diagnostics extension.

Windows diagnostics extension (WAD)

Expand table

Data source	Description
Windows event logs	Events from Windows event log.
Performance counters	Numerical values measuring performance of different aspects of operating system and workloads.
IIS logs	Usage information for IIS websites running on the guest operating system.
Application logs	Trace messages written by your application.

Data source	Description
.NET EventSource logs	Code writing events using the .NET EventSource class.
Manifest-based ETW logs	Event tracing for Windows events generated by any process.
Crash dumps (logs)	Information about the state of the process if an application crashes.
File-based logs	Logs created by your application or service.
Agent diagnostic logs	Information about Azure Diagnostics itself.

Linux diagnostics extension (LAD)

Expand table

Data source	Description
Syslog	Events sent to the Linux event logging system
Performance counters	Numerical values measuring performance of different aspects of operating system and workloads
Log files	Entries sent to a file-based log

Data destinations

The Azure Diagnostics extension for both Windows and Linux always collects data into an Azure Storage account. For a list of specific tables and blobs where this data is collected, see Install and configure Azure Diagnostics extension for Windows and Use Azure Diagnostics extension for Linux to monitor metrics and logs.

Configure one or more *data sinks* to send data to other destinations. The following sections list the sinks available for the Windows and Linux diagnostics extension.

Windows diagnostics extension (WAD)

Expand table

Destination	Description
Azure Monitor Metrics	Collect performance data to Azure Monitor Metrics. See Send Guest OS metrics to the Azure Monitor metric database.

Destination	Description
Event hubs	Use Azure Event Hubs to send data outside of Azure. See Streaming Azure Diagnostics data to Azure Event Hubs.
Azure Storage blobs	Write data to blobs in Azure Storage in addition to tables.
Application Insights	To integrate with other application monitoring, collect data from applications running in your VM to Application Insights. See Send diagnostic data to Application Insights.

You can also collect WAD data from storage into a Log Analytics workspace to analyze it with Azure Monitor Logs, although the Log Analytics agent is typically used for this functionality. It can send data directly to a Log Analytics workspace and supports solutions and insights that provide more functionality. See Collect Azure diagnostic logs from Azure Storage.

Linux diagnostics extension (LAD)

LAD writes data to tables in Azure Storage. It supports the sinks in the following table.

Expand table

Destination	Description
Event hubs	Use Azure Event Hubs to send data outside of Azure.
Azure Storage blobs	Write data to blobs in Azure Storage in addition to tables.
Azure Monitor Metrics	Install the Telegraf agent in addition to LAD. See Collect custom metrics for a Linux VM with the InfluxData Telegraf agent.

Installation and configuration

The diagnostics extension is implemented as a virtual machine extension in Azure, so it supports the same installation options using Azure Resource Manager templates, PowerShell, and the Azure CLI. For information on installing and maintaining virtual machine extensions, see Virtual machine extensions and features for Windows and Virtual machine extensions and features for Linux.

You can also install and configure both the Windows and Linux diagnostics extension in the Azure portal under **Diagnostic settings** in the **Monitoring** section of the virtual machine's menu.

See the following articles for information on installing and configuring the diagnostics extension for Windows and Linux:

- Install and configure Azure Diagnostics extension for Windows
- Use Linux diagnostics extension to monitor metrics and logs

Supported operating systems

The following tables list the operating systems supported by WAD and LAD. See the documentation for each agent for unique considerations and for the installation process. See Telegraf documentation for its supported operating systems. All operating systems are assumed to be x64. x86 isn't supported for any operating system.

Windows

Expand table

Operating system	Support
Windows Server 2022	❌
Windows Server 2022 Core	❌
Windows Server 2019	✅
Windows Server 2019 Core	❌
Windows Server 2016	✅
Windows Server 2016 Core	✅
Windows Server 2012 R2	✅
Windows Server 2012	✅
Windows 11 Client & Pro	❌
Windows 11 Enterprise (including multi-session)	❌
Windows 10 1803 (RS4) and higher	❌
Windows 10 Enterprise (including multi-session) and Pro (Server scenarios only)	✅

Linux

Expand table

Operating system	Support
CentOS Linux 9	❌
CentOS Linux 8	❌
CentOS Linux 7	✅
Debian 12	❌
Debian 11	❌
Debian 10	❌
Debian 9	✅
Debian 8	❌
Oracle Linux 9	❌
Oracle Linux 8	❌
Oracle Linux 7	✅
Oracle Linux 6.4+	✅
Red Hat Enterprise Linux Server 9	❌
Red Hat Enterprise Linux Server 8*	✅
Red Hat Enterprise Linux Server 7	✅
SUSE Linux Enterprise Server 15	❌
SUSE Linux Enterprise Server 12	✅
Ubuntu 22.04 LTS	❌
Ubuntu 20.04 LTS	✅
Ubuntu 18.04 LTS	✅
Ubuntu 16.04 LTS	✅
Ubuntu 14.04 LTS	✅

* Requires Python 2 to be installed on the machine and aliased to the python command.

Other documentation

For more information, see the following articles.

Azure Cloud Services (classic) web and worker roles

- Introduction to Azure Cloud Services monitoring
- Enabling Azure Diagnostics in Azure Cloud Services
- Application Insights for Azure Cloud Services
- Trace the flow of an Azure Cloud Services application with Azure Diagnostics

Azure Service Fabric

Monitor and diagnose services in a local machine development setup

Next steps

- Learn to use performance counters in Azure Diagnostics.
- If you have trouble with diagnostics starting or finding your data in Azure Storage tables, see Troubleshooting Azure Diagnostics.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/07/2026

Azure Monitor Dependency virtual machine extension for Linux

Article • 01/14/2025

The Azure Monitor for VMs Map feature gets its data from the Microsoft Dependency agent. The Azure VM Dependency agent virtual machine extension for Linux installs the Dependency agent on Azure virtual machines. This document details the supported platforms, configurations, and deployment options for the Azure VM Dependency agent virtual machine extension for Linux.

Prerequisites

Operating system

Because the Azure VM Dependency agent works at the kernel level, operating system support is also dependent on the kernel version. As of Dependency agent version 9.10.* the agent supports * kernels. The following table lists the major and minor Linux OS release and supported kernel versions for the Dependency agent.

> **Note**
>
> With Dependency agent 9.10.15 and above, installation is not blocked for unsupported kernel versions, but the agent will run in degraded mode. In this mode, connection and port data stored in VMConnection and VMBoundport tables is not collected. The VMProcess table may have some data, but it will be minimal.

Expand table

Distribution	OS version	Kernel version
Red Hat Linux 8	8.6	4.18.0-372.*el8.x86_64, 4.18.0-372.*el8_6.x86_64
	8.5	4.18.0-348.*el8_5.x86_644.18.0-348.*el8.x86_64
	8.4	4.18.0-305.*el8.x86_64, 4.18.0-305.*el8_4.x86_64
	8.3	4.18.0-240.*el8_3.x86_64
	8.2	4.18.0-193.*el8_2.x86_64
	8.1	4.18.0-147.*el8_1.x86_64

Distribution	OS version	Kernel version
	8.0	4.18.0-80.*el8.x86_64 4.18.0-80.*el8_0.x86_64
Red Hat Linux 7	7.9	3.10.0-1160
	7.8	3.10.0-1136
	7.7	3.10.0-1062
	7.6	3.10.0-957
	7.5	3.10.0-862
	7.4	3.10.0-693
Red Hat Linux 6	6.10	2.6.32-754
	6.9	2.6.32-696
CentOS Linux 8	8.6	4.18.0-372.*el8.x86_64, 4.18.0-372.*el8_6.x86_64
	8.5	4.18.0-348.*el8_5.x86_644.18.0-348.*el8.x86_64
	8.4	4.18.0-305.*el8.x86_64, 4.18.0-305.*el8_4.x86_64
	8.3	4.18.0-240.*el8_3.x86_64
	8.2	4.18.0-193.*el8_2.x86_64
	8.1	4.18.0-147.*el8_1.x86_64
	8.0	4.18.0-80.*el8.x86_64 4.18.0-80.*el8_0.x86_64
CentOS Linux 7	7.9	3.10.0-1160
	7.8	3.10.0-1136
	7.7	3.10.0-1062
CentOS Linux 6	6.10	2.6.32-754.3.5 2.6.32-696.30.1
	6.9	2.6.32-696.30.1 2.6.32-696.18.7
Ubuntu Server	20.04	5.8 5.4*
	18.04	5.3.0-1020 5.0 (includes Azure-tuned kernel)

Distribution	OS version	Kernel version
		4.18* 4.15*
	16.04.3	4.15.*
	16.04	4.13.* 4.11.* 4.10.* 4.8.* 4.4.*
	14.04	3.13.*-generic 4.4.*-generic
SUSE Linux 12 Enterprise Server	12 SP5	4.12.14-122.*-default, 4.12.14-16.*-azure
	12 SP4	4.12.* (includes Azure-tuned kernel)
	12 SP3	4.4.*
	12 SP2	4.4.*
SUSE Linux 15 Enterprise Server	15 SP1	4.12.14-197.*-default, 4.12.14-8.*-azure
	15	4.12.14-150.*-default
Debian	9	4.9

> **Note**
>
> Dependency agent is not supported for Azure Virtual Machines with Ampere Altra ARM–based processors.

Extension schema

The following JSON shows the schema for the Azure VM Dependency agent extension on an Azure Linux VM.

JSON

```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
      "vmName": {
```

```
        "type": "string",
        "metadata": {
            "description": "The name of existing Linux Azure VM."
      }
    }
  },
  "variables": {
      "vmExtensionsApiVersion": "2017-03-30"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "name": "[concat(parameters('vmName'),'/DAExtension')]",
      "apiVersion": "[variables('vmExtensionsApiVersion')]",
      "location": "[resourceGroup().location]",
      "dependsOn": [],
      "properties": {
          "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
          "type": "DependencyAgentLinux",
          "typeHandlerVersion": "9.5",
          "autoUpgradeMinorVersion": true,
          "settings": {
              "enableAMA": "true"
        }
      }
    }
  ],
    "outputs": {
    }
}
```

Property values

Expand table

Name	Value/Example
apiVersion	2015-01-01
publisher	Microsoft.Azure.Monitoring.DependencyAgent
type	DependencyAgentLinux
typeHandlerVersion	9.5
settings	"enableAMA": "true"

Important

Be sure to add `enableAMA` to your template if you're using Azure Monitor Agent; otherwise, Dependency agent attempts to send data to the legacy Log Analytics agent.

Template deployment

You can deploy Azure VM extensions with Azure Resource Manager templates. You can use the JSON schema detailed in the previous section in an Azure Resource Manager template to run the Azure VM Dependency agent extension during an Azure Resource Manager template deployment.

The JSON for a virtual machine extension can be nested inside the virtual machine resource. Or, you can place it at the root or top level of a Resource Manager JSON template. The placement of the JSON affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the Dependency agent extension is nested inside the virtual machine resource. When you nest the extension resource, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
    "type": "extensions",
    "name": "DAExtension",
    "apiVersion": "[variables('apiVersion')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
      "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
    ],
    "properties": {
        "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
        "type": "DependencyAgentLinux",
        "typeHandlerVersion": "9.5",
        "autoUpgradeMinorVersion": true,
        "settings": {
            "enableAMA": "true"
        }
    }
}
```

When you place the extension JSON at the root of the template, the resource name includes a reference to the parent virtual machine. The type reflects the nested configuration.

JSON

```
{
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "name": "<parentVmResource>/DAExtension",
    "apiVersion": "[variables('apiVersion')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
      "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
    ],
    "properties": {
        "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
        "type": "DependencyAgentLinux",
        "typeHandlerVersion": "9.5",
        "autoUpgradeMinorVersion": true,
        "settings": {
            "enableAMA": "true"
        }
    }
}
```

Azure CLI deployment

You can use the Azure CLI to deploy the Dependency agent VM extension to an existing virtual machine.

Azure CLI

```
az vm extension set \
    --resource-group myResourceGroup \
    --vm-name myVM \
    --name DependencyAgentLinux \
    --publisher Microsoft.Azure.Monitoring.DependencyAgent \
    --version 9.10 \
    --settings '{"enableAMA": "true"}'
```

Automatic extension upgrade

A new feature to automatically upgrade minor versions of Dependency extension is now available.

To enable automatic extension upgrade for an extension, you must ensure the property `enableAutomaticUpgrade` is set to `true` and added to the extension template. This property must be enabled on every VM or VM scale set individually. Use one of the methods described in the enablement section enable the feature for your VM or VM scale set.

When automatic extension upgrade is enabled on a VM or VM scale set, the extension is upgraded automatically whenever the extension publisher releases a new version for that extension. The upgrade is applied safely following availability-first principles as described here.

The `enableAutomaticUpgrade` attribute's functionality is different from that of the `autoUpgradeMinorVersion`. The `autoUpgradeMinorVersion` attribute doesn't automatically trigger a minor version update when the extension publisher releases a new version. The `autoUpgradeMinorVersion` attribute indicates whether the extension should use a newer minor version if one is available at deployment time. Once deployed, however, the extension won't upgrade minor versions unless redeployed, even with this property set to true.

To keep your extension version updated, we recommend using `enableAutomaticUpgrade` with your extension deployment.

> **Important**
>
> If you add the `enableAutomaticUpgrade` to your template, make sure that you use at API version 2019-12-01 or higher.

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal and by using the Azure CLI. To see the deployment state of extensions for a given VM, run the following command by using the Azure CLI:

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o
table
```

Extension execution output is logged to the following file:

```
/var/opt/microsoft/dependency-agent/log/install.log
```

Support

If you need more help at any point in this article, contact the Azure experts on the Microsoft Q & A and Stack Overflow forums. Or, you can file an Azure support incident. Go to the Azure support site and select **Get support**. For information about how to use Azure Support, read the Microsoft Azure support FAQ.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Monitor Dependency virtual machine extension for Windows

The Azure Monitor for VMs Map feature gets its data from the Microsoft Dependency agent. The Azure VM Dependency agent virtual machine extension for Windows installs the Dependency agent on Azure virtual machines. This document details the supported platforms, configurations, and deployment options for the Azure VM Dependency agent virtual machine extension for Windows.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Operating system

The Azure VM Dependency agent extension for Windows can be run against the supported operating systems listed in the following table. All operating systems in the following table are assumed to be x64. x86 isn't supported for any operating system.

Expand table

Operating system	Azure VM Dependency agent
Windows Server 2025	
Windows Server 2022	✓
Windows Server 2019	✓
Windows Server 2019 Core	
Windows Server 2016	✓
Windows Server 2016 Core	
Windows Server 2012 R2	✓
Windows Server 2012	✓
Windows 11 Client and Pro	✓
Windows 11 Enterprise	✓

Operating system	Azure VM Dependency agent
(including multi-session)	
Windows 10 1803 (RS4) and higher	✓
Windows 10 Enterprise (including multi-session) and Pro (Server scenarios only)	✓
Windows 8 Enterprise and Pro (Server scenarios only)	
Windows 7 SP1 (Server scenarios only)	
Azure Stack HCI	
Windows IoT Enterprise	

Extension schema

The following JSON shows the schema for the Azure VM Dependency agent extension on an Azure Windows VM.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "type": "string",
      "metadata": {
        "description": "The name of existing Azure VM. Supported Windows Server
versions:  2008 R2 and above (x64)."
      }
    }
  },
  "variables": {
    "vmExtensionsApiVersion": "2017-03-30"
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "name": "[concat(parameters('vmName'),'/DAExtension')]",
      "apiVersion": "[variables('vmExtensionsApiVersion')]",
      "location": "[resourceGroup().location]",
      "dependsOn": [],
      "properties": {
```

```
            "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
            "type": "DependencyAgentWindows",
            "typeHandlerVersion": "9.10",
            "autoUpgradeMinorVersion": true,
            "settings": {
                  "enableAMA": "true"
              }
        }
      }
    ],
      "outputs": {
      }
}
```

Property values

Expand table

Name	Value/Example
apiVersion	2015-01-01
publisher	Microsoft.Azure.Monitoring.DependencyAgent
type	DependencyAgentWindows
typeHandlerVersion	9.10
autoUpgradeMinorVersion	true
settings	"enableAMA": "true"

> **Important**
>
> Be sure to add `enableAMA` to your template if you're using Azure Monitor Agent; otherwise, Dependency agent attempts to send data to the legacy Log Analytics agent.

Template deployment

You can deploy the Azure VM extensions with Azure Resource Manager templates. You can use the JSON schema detailed in the previous section in an Azure Resource Manager template to run the Azure VM Dependency agent extension during an Azure Resource Manager template deployment.

The JSON for a virtual machine extension can be nested inside the virtual machine resource. Or, you can place it at the root or top level of a Resource Manager JSON template. The placement

of the JSON affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the Dependency agent extension is nested inside the virtual machine resource. When you nest the extension resource, the JSON is placed in the `"resources": []` object of the virtual machine.

JSON

```
{
    "type": "extensions",
    "name": "DAExtension",
    "apiVersion": "[variables('apiVersion')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
    ],
    "properties": {
      "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
      "type": "DependencyAgentWindows",
      "typeHandlerVersion": "9.10",
      "autoUpgradeMinorVersion": true,
      "settings": {
            "enableAMA": "true"
                }
    }
}
```

When you place the extension JSON at the root of the template, the resource name includes a reference to the parent virtual machine. The type reflects the nested configuration.

JSON

```
{
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "name": "<parentVmResource>/DAExtension",
    "apiVersion": "[variables('apiVersion')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
    ],
    "properties": {
      "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
      "type": "DependencyAgentWindows",
      "typeHandlerVersion": "9.10",
      "autoUpgradeMinorVersion": true,
      "settings": {
            "enableAMA": "true"
                }
    }
}
```

PowerShell deployment

You can use the `Set-AzVMExtension` command to deploy the Dependency agent virtual machine extension to an existing virtual machine. Before you run the command, the public and private configurations need to be stored in a PowerShell hash table.

PowerShell

```
Set-AzVMExtension -ExtensionName "Microsoft.Azure.Monitoring.DependencyAgent" `
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Publisher "Microsoft.Azure.Monitoring.DependencyAgent" `
    -ExtensionType "DependencyAgentWindows" `
    -TypeHandlerVersion 9.10 `
    -Settings @{"enableAMA" = "true"}
```

PowerShell deployment for Azure Monitor Agent

If you are using the Azure Monitor Agent, you must use the `enableAMA` setting. Otherwise, Dependency agent attempts to send data to the legacy Log Analytics agent. For example:

PowerShell

```
Set-AzVMExtension -ExtensionName "Microsoft.Azure.Monitoring.DependencyAgent" `
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Publisher "Microsoft.Azure.Monitoring.DependencyAgent" `
    -ExtensionType "DependencyAgentWindows" `
    -TypeHandlerVersion 9.10 `
    -Location WestUS `
    -Settings @{"enableAMA" = "true"}
```

Automatic extension upgrade

A new feature to automatically upgrade minor versions of Dependency extension is now available.

To enable automatic extension upgrade for an extension, you must ensure the property `enableAutomaticUpgrade` is set to `true` and added to the extension template. This property must be enabled on every VM or VM scale set individually. Use one of the methods described in the enablement section enable the feature for your VM or VM scale set.

When automatic extension upgrade is enabled on a VM or VM scale set, the extension is upgraded automatically whenever the extension publisher releases a new version for that

extension. The upgrade is applied safely following availability-first principles as described here.

The `enableAutomaticUpgrade` attribute's functionality is different from that of the `autoUpgradeMinorVersion`. The `autoUpgradeMinorVersion` attribute doesn't automatically trigger a minor version update when the extension publisher releases a new version. The `autoUpgradeMinorVersion` attribute indicates whether the extension should use a newer minor version if one is available at deployment time. Once deployed, however, the extension won't upgrade minor versions unless redeployed, even with this property set to true.

To keep your extension version updated, we recommend using `enableAutomaticUpgrade` with your extension deployment.

> **Important**
>
> If you add the `enableAutomaticUpgrade` to your template, make sure that you use at API version 2019-12-01 or higher.

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal and by using the Azure PowerShell module. To see the deployment state of extensions for a given VM, run the following command by using the Azure PowerShell module:

PowerShell

```
Get-AzVMExtension -ResourceGroupName myResourceGroup -VMName myVM -Name
myExtensionName
```

Extension execution output is logged to files found in the following directory:

Windows Command Prompt

```
C:\WindowsAzure\Logs\Plugins\Microsoft.Azure.Monitoring.DependencyAgent\
```

Support

If you need more help at any point in this article, you can contact the Azure experts on the Microsoft Q & A and Stack Overflow forums. Or, you can file an Azure support incident. Go

to the Azure support site and select **Get support**. For information about how to use Azure Support, read the Microsoft Azure support FAQ.

Last updated on 11/10/2025

Introduction to the Azure Desired State Configuration extension handler

> **Note**
>
> DSC extension will be retired on March 31, 2028. Please move to Azure Machine Configuration by that date. For more information, see the blog post announcement. The Azure Machine Configuration service combines certain features of DSC Extension, Azure Automation State Configuration, and commonly requested features from customer feedback. Azure Machine Configuration also includes hybrid machine support through Arc-enabled servers.

The Azure VM Extension for Azure virtual machines (VM) and the associated extensions are part of Microsoft Azure infrastructure services. Azure VM extensions are software components that extend VM functionality and simplify various VM management operations.

The DSC extension only pushes a configuration to the VM. No ongoing reporting is available, other than locally in the VM.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Available DSC versions

The DSC extension supports configurations from version 1.1 of the DSC platform. For more information, see PSDesiredStateConfiguration v1.1.

Prerequisites

- **Developer workstation**: To interact with the Azure DSC extension, you must use either the Azure portal or the Azure PowerShell/CLI SDK.

- **Guest agent**: The Azure VM that's prepared by the DSC configuration must use an operating system that supports Windows Management Framework (WMF) 4.0 or later. For the full list of supported operating system versions, see the Azure DSC extension version history.

Terms and concepts

This article assumes familiarity with the following concepts:

- **Configuration** refers to a DSC configuration document.
- **Node** identifies a target for a DSC configuration. In this article, *node* always refers to an Azure VM.
- **Configuration data** is stored in a PowerShell DSC format file (.psd1) that has environmental data for a configuration.

Architecture

The Azure DSC extension uses the Azure VM Extension framework to deliver, enact, and report on DSC configurations running on Azure VMs. The DSC extension accepts a configuration document and a set of parameters.

When the extension is deployed the first time, it installs a version of WMF by using the following logic:

- If the Azure VM operating system is Windows Server 2016, no action is taken. Windows Server 2016 already has the latest version of PowerShell installed.
- If the `wmfVersion` property is specified, the specified version of WMF is installed, unless the specified version is incompatible with the operating system on the VM.
- If no `wmfVersion` property is specified, the latest applicable version of WMF is installed.

The WMF installation process requires a restart. After you restart, the extension automates the download of the `.zip` file specified in the `modulesUrl` property, if provided. If this location is in Azure Blob Storage, you can specify an SAS token in the `sasToken` property to access the file. Next, the configuration function defined in `configurationFunction` runs to generate a Managed Object Format (MOF) file (.mof). The extension then runs the `Start-DscConfiguration -Force` command by using the generated .mof file. The extension captures output and writes it to the Azure status channel.

Node configuration name

For the `NodeConfigurationName` parameter, be sure to provide the name of the node configuration and not the Configuration.

The Configuration is defined in a script that's used to compile the node configuration (MOF file). The name of the node configuration is always the name of the Configuration followed by a period `.` and either `localhost` or a specific computer name.

ARM template deployment

The most common approach for deploying the DSC extension is to use Azure Resource Manager templates. For more information and for examples of how to include the DSC extension in ARM templates, see Desired State Configuration extension with ARM templates.

PowerShell cmdlet deployment

PowerShell cmdlets for managing the DSC extension are ideal for interactive troubleshooting and information-gathering scenarios. You can use the cmdlets to package, publish, and monitor DSC extension deployments.

Here are some of the PowerShell cmdlets that are available:

- The **Publish-AzVMDscConfiguration** cmdlet takes in a configuration file, scans it for dependent DSC resources, and then creates a `.zip` file. The `.zip` file contains the configuration and DSC resources that are needed to enact the configuration. The cmdlet can also create the package locally by using the `-OutputArchivePath` parameter. Otherwise, the cmdlet publishes the `.zip` file to Blob Storage, and then secures it with an SAS token.

  The PowerShell configuration script (`.ps1`) created by the cmdlet is in the `.zip` file at the root of the archive folder. The module folder is placed in the archive folder in resources.

- The **Set-AzVMDscExtension** cmdlet injects the settings that the PowerShell DSC extension requires into a VM configuration object.

- The **Get-AzVMDscExtension** cmdlet retrieves the DSC extension status of a specific VM.

- The **Get-AzVMDscExtensionStatus** cmdlet retrieves the status of the DSC configuration that's enacted by the DSC extension handler. This action can be performed on a single VM or a group of VMs.

- The **Remove-AzVMDscExtension** cmdlet removes the extension handler from a specific VM. Keep in mind that this cmdlet doesn't remove the configuration, uninstall WMF, or change the applied settings on the VM. The cmdlet only removes the extension handler.

Important considerations

There are several considerations to keep in mind when working with Azure Resource Manager cmdlets.

- Azure Resource Manager cmdlets are synchronous.
- Several parameters are required, including `ResourceGroupName`, `VMName`, `ArchiveStorageAccountName`, `Version`, and `Location`.
- `ArchiveResourceGroupName` is an optional parameter. Specify this parameter when your storage account belongs to a different resource group than the one where the VM is created.
- Use the `AutoUpdate` switch to automatically update the extension handler to the latest version when it's available. This parameter has the potential to cause restarts on the VM when a new version of WMF is released.

Configuration with PowerShell cmdlets

The Azure DSC extension can use DSC configuration documents to directly configure Azure VMs during deployment. This step doesn't register the node to Automation or Machine Configuration. Keep in mind that the node isn't centrally managed.

The following code shows a simple example configuration. To work with this example, save this configuration locally as the **iisInstall.ps1** script file.

PowerShell

```
configuration IISInstall
{
    node "localhost"
    {
        WindowsFeature IIS
        {
            Ensure = "Present"
            Name = "Web-Server"
        }
    }
}
```

The following PowerShell commands place the `iisInstall.ps1` script on the specified VM. The commands also execute the configuration, and then report back on status.

PowerShell

```
$resourceGroup = 'dscVmDemo'
$vmName = 'myVM'
$storageName = 'demostorage'
#Publish the configuration script to user storage
Publish-AzVMDscConfiguration -ConfigurationPath .\iisInstall.ps1 -ResourceGroupName
$resourceGroup -StorageAccountName $storageName -force
#Set the VM to run the DSC configuration
Set-AzVMDscExtension -Version '2.76' -ResourceGroupName $resourceGroup -VMName
$vmName -ArchiveStorageAccountName $storageName -ArchiveBlobName
'iisInstall.ps1.zip' -AutoUpdate -ConfigurationName 'IISInstall'
```

Azure CLI deployment

The Azure CLI can be used to deploy the DSC extension to an existing VM. The following examples show how to deploy a VM on Windows.

For a VM running Windows, use the following command:

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name DSC \
  --publisher Microsoft.Powershell \
  --version 2.77 --protected-settings '{}' \
  --settings '{}'
```

Azure portal deployment

To set up the DSC extension in the Azure portal, follow these steps:

1. Go to a VM.
2. Under **Settings**, select **Extensions + Applications**.
3. Under **Extensions**, select **+ Add**.
4. Select **PowerShell Desired State Configuration**, then select **Next**.
5. Configure the following parameters for the DSC extension.

- **Configuration Modules or Script**: (Required) Provide the Configuration modules or script file for your VM.

  Configuration modules and scripts require a `.ps1` file that has a configuration script or a `.zip` file with a `.ps1` configuration script at the root. If you use a `.zip` file, all dependent resources must be included in module folders in the `.zip` file. You can create the `.zip` file by using the **Publish-AzureVMDscConfiguration -OutputArchivePath** cmdlet that's included in the Azure PowerShell SDK. The `.zip` file is uploaded to your user Blob Storage and secured by an SAS token.

- **Module-qualified Name of Configuration**: (Required) Specify this setting to include multiple configuration functions in a single `.ps1` script file. For this setting, enter the name of the configuration `.ps1` script file followed by a slash `\` and then the name of the configuration function. For example, if the `.ps1` script file has the name **configuration.ps1** and the configuration name is **IisInstall**, enter the value `configuration.ps1\IisInstall` for the setting.

- **Configuration Arguments**: If the configuration function takes arguments, enter the values by using the format `argumentName1=value1,argumentName2=value2`. Notice that this format differs from the format that's used to specify configuration arguments in PowerShell cmdlets or ARM templates.

- **Configuration Data PSD1 File**: If your configuration requires a configuration data file in `.psd1` format, use this setting to select the data file and upload it to your user Blob Storage. The configuration data file is secured with an SAS token in Blob Storage.

- **WMF Version**: Specify the version of Windows Management Framework to install on your VM. If you choose **latest**, which is the default value, the system installs the most recent version of WMF. Other possible values include 4.0, 5.0, and 5.1. The possible values are subject to updates.

- **Data Collection**: Enable this setting if you want the DSC extension to collect telemetry about your VM. For more information, see Azure DSC extension data collection.

- **Version**: (Required) Specify the version of the DSC extension to install. For information about versions, see Azure DSC extension version history.

- **Auto Upgrade Minor Version**: This setting maps to the `AutoUpdate` switch in the cmdlets. Configure this setting to enable the DSC extension to automatically update to the latest version during installation. **Yes** instructs the DSC extension handler to

use the latest available version. **No** (default) forces installation of the version you specify in the **Version** setting.

6. After you configure the parameters, select **Review + Create**, and then select **Create**.

DSC extension logs

You can view logs for the Azure DSC extension on the VM under `C:\WindowsAzure\Logs\Plugins\Microsoft.Powershell.DSC\<version number>`.

Next steps

- For more information about PowerShell DSC, go to the PowerShell documentation center.
- Examine the ARM template for the Azure DSC extension.
- For more functionality that you can manage by using PowerShell DSC, and for more DSC resources, browse the PowerShell gallery.
- For details about passing sensitive parameters into configurations, see Manage credentials securely with the Azure DSC extension handler.

Last updated on 11/10/2025

PowerShell DSC Extension

> **Note**
>
> DSC extension will be retired on March 31, 2028. Move to Azure Machine Configuration by that date. For more information, see the blog post announcement. The Azure Machine Configuration service combines certain features of DSC Extension, Azure Automation State Configuration, and commonly requested features from customer feedback. Azure Machine Configuration also includes hybrid machine support through Arc-enabled servers.

Overview

The PowerShell DSC Extension for Windows uploads and applies a PowerShell DSC Configuration on an Azure virtual machine. The DSC Extension calls into PowerShell DSC to enact the received DSC configuration on the virtual machine. This document details the supported platforms, configurations, and deployment options for the DSC virtual machine extension for Windows.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Operating system

The DSC Extension supports the following operating systems

Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012R2, Windows Server 2012, Windows Server 2008 R2 SP1, Windows Client 7/8.1/10

Internet connectivity

The DSC extension for Windows requires that the target virtual machine is able to communicate with Azure and the location of the configuration package (.zip file) if stored outside of Azure.

Extension schema

The following JSON shows the schema for the settings portion of the DSC Extension in an Azure Resource Manager template.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachines/extensions",
  "name": "Microsoft.Powershell.DSC",
  "apiVersion": "2018-10-01",
  "location": "<location>",
  "properties": {
    "publisher": "Microsoft.Powershell",
    "type": "DSC",
    "typeHandlerVersion": "2.77",
    "autoUpgradeMinorVersion": true,
    "settings": {
        "wmfVersion": "latest",
        "configuration": {
            "url": "http://validURLToConfigLocation",
            "script": "ConfigurationScript.ps1",
            "function": "ConfigurationFunction"
        },
        "configurationArguments": {
            "argument1": "Value1",
            "argument2": "Value2"
        },
        "configurationData": {
            "url": "https://foo.psd1"
        },
        "privacy": {
            "dataCollection": "enable"
        },
        "advancedOptions": {
            "forcePullAndApply": false,
            "downloadMappings": {
             "specificDependencyKey": "https://myCustomDependencyLocation"
            }
        }
    },
    "protectedSettings": {
        "configurationArguments": {
            "parameterOfTypePSCredential1": {
                "userName": "UsernameValue1",
                "password": "PasswordValue1"
            },
```

```
            "parameterOfTypePSCredential2": {
                "userName": "UsernameValue2",
                "password": "PasswordValue2"
            }
        },
        "configurationUrlSasToken": "?g!bber1sht0k3n",
        "configurationDataUrlSasToken": "?dataAcC355T0k3N"
    }
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	2018-10-01	date
publisher	Microsoft.Powershell.DSC	string
type	DSC	string
typeHandlerVersion	2.77	int

Settings Property values

Expand table

Name	Data Type	Description
settings.wmfVersion	string	Specifies the version of the Windows Management Framework that should be installed on your VM. Setting this property to 'latest' installs the most updated version of WMF. The only current possible values for this property are ‘4.0’, ‘5.0’, and 'latest'. These possible values are subject to updates. The default value is 'latest'.
settings.configuration.url	string	Specifies the URL location from which to download your DSC configuration zip file. If the URL provided requires a SAS token for access, set the protectedSettings.configurationUrlSasToken property to the value of your SAS token. This property is required if

Name	Data Type	Description
		settings.configuration.script and/or settings.configuration.function are defined.
settings.configuration.script	string	Specifies the file name of the script that contains the definition of your DSC configuration. This script must be in the root folder of the zip file downloaded from the URL specified by the configuration.url property. This property is required if settings.configuration.url and/or settings.configuration.script are defined.
settings.configuration.function	string	Specifies the name of your DSC configuration. The configuration named must be contained in the script defined by configuration.script. This property is required if settings.configuration.url and/or settings.configuration.function are defined.
settings.configurationArguments	Collection	Defines any parameters you would like to pass to your DSC configuration. This property won't be encrypted.
settings.configurationData.url	string	Specifies the URL from which to download your configuration data (.pds1) file to use as input for your DSC configuration. If the URL provided requires a SAS token for access, set the protectedSettings.configurationDataUrlSasToken property to the value of your SAS token.
settings.privacy.dataEnabled	string	Enables or disables telemetry collection. The only possible values for this property are 'Enable', 'Disable', '', or $null. Leaving this property blank or null enables telemetry
settings.advancedOptions.forcePullAndApply	Bool	This setting is designed to enhance the experience of working with the extension to register nodes with Azure Automation DSC. If the value is `$true`, the extension waits for the first run of the configuration pulled from the service before returning success/failure. If the value is set to $false, the status returned by the extension refers to whether the node was registered with Azure Automation State Configuration successfully and the node configuration won't run during the registration.
settings.advancedOptions.downloadMappings	Collection	Defines alternate locations to download dependencies such as WMF and .NET

Protected Settings Property values

Expand table

Name	Data Type	Description
protectedSettings.configurationArguments	string	Defines any parameters you would like to pass to your DSC configuration.
protectedSettings.configurationUrlSasToken	string	Specifies the SAS token to access the URL defined by configuration.url.
protectedSettings.configurationDataUrlSasToken	string	Specifies the SAS token to access the URL defined by configurationData.url.

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. Templates are ideal when deploying one or more virtual machines that require post deployment configuration. A sample Resource Manager template that includes the DSC extension for Windows can be found on the Azure Quick Start Gallery.

Troubleshoot and support

Troubleshoot

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure CLI. To see the deployment state of extensions for a given VM, run the following command using the Azure CLI.

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Extension package is downloaded and deployed to this location on the Azure VM

```
C:\Packages\Plugins\{Extension_Name}\{Extension_Version}
```

Extension status file contains the sub status and status success/error codes along with the detailed error and description for each extension run.

```
C:\Packages\Plugins\{Extension_Name}\{Extension_Version}\Status\{0}.Status  -> {0}
being the sequence number
```

Extension output logs are logged to the following directory:

```
C:\WindowsAzure\Logs\Plugins\{Extension_Name}\{Extension_Version}
```

Error codes and their meanings

Expand table

Error Code	Meaning	Possible Action
1000	Generic error	The message for this error is provided by the specific exception in extension logs
52	Extension Install Error	The message for this error is provided by the specific exception
1002	Wmf Install Error	Error while installing WMF.
1004	Invalid Zip Package	Invalid zip ; Error unpacking the zip
1100	Argument Error	Indicates a problem in the input provided by the user. The message for the error is provided by the specific exception

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 11/10/2025

Pass credentials to the Azure DSCExtension handler

> **Note**
>
> DSC extension will be retired on March 31, 2028. Please transition to Azure Machine Configuration by that date. For more information, see the blog post announcement. The Azure Machine Configuration service combines certain features of DSC Extension, Azure Automation State Configuration, and commonly requested features from customer feedback. Azure Machine Configuration also includes hybrid machine support through Arc-enabled servers.

This article covers the Desired State Configuration (DSC) extension for Azure. For an overview of the DSC extension handler, see Introduction to the Azure Desired State Configuration extension handler.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Pass in credentials

As part of the configuration process, you might need to set up user accounts, access services, or install a program in a user context. To do these things, you need to provide credentials.

You can use DSC to set up parameterized configurations. In a parameterized configuration, credentials are passed into the configuration and securely stored in .mof files. The Azure extension handler simplifies credential management by providing automatic management of certificates.

The following DSC configuration script creates a local user account with the specified password:

PowerShell

```
configuration Main
{
    param(
        [Parameter(Mandatory=$true)]
```

```
        [ValidateNotNullorEmpty()]
        [PSCredential]
        $Credential
    )
    Node localhost {
        User LocalUserAccount
        {
            Username = $Credential.UserName
            Password = $Credential
            Disabled = $false
            Ensure = "Present"
            FullName = "Local User Account"
            Description = "Local User Account"
            PasswordNeverExpires = $true
        }
    }
}
```

It's important to include **node localhost** as part of the configuration. The extension handler specifically looks for the **node localhost** statement. If this statement is missing, the following steps don't work. It's also important to include the typecast **[PsCredential]**. This specific type triggers the extension to encrypt the credential.

To publish this script to Azure Blob storage:

```
Publish-AzVMDscConfiguration -ConfigurationPath .\user_configuration.ps1
```

To set the Azure DSC extension and provide the credential:

PowerShell

```
$configurationName = 'Main'
$configurationArguments = @{ Credential = Get-Credential }
$configurationArchive = 'user_configuration.ps1.zip'
$vm = Get-AzVM -Name 'example-1'

$vm = Set-AzVMDscExtension -VMName $vm -ConfigurationArchive $configurationArchive -
ConfigurationName $configurationName -ConfigurationArgument @configurationArguments

$vm | Update-AzVM
```

How a credential is secured

Running this code prompts for a credential. After the credential is provided, it's briefly stored in memory. When the credential is published by using the **Set-AzVMDscExtension** cmdlet, the credential is transmitted over HTTPS to the VM. In the VM, Azure stores the credential encrypted on disk by using the local VM certificate. The credential is briefly decrypted in memory, and then it's re-encrypted to pass it to DSC.

This process is different than using secure configurations without the extension handler. The Azure environment gives you a way to transmit configuration data securely via certificates. When you use the DSC extension handler, you don't need to provide **$CertificatePath** or a **$CertificateID**/ **$Thumbprint** entry in **ConfigurationData**.

Next steps

- Get an introduction to Azure DSC extension handler.
- Examine the Azure Resource Manager template for the DSC extension.
- For more information about PowerShell DSC, go to the PowerShell documentation center.
- For more functionality that you can manage by using PowerShell DSC, and for more DSC resources, browse the PowerShell gallery.

Last updated on 11/10/2025

Desired State Configuration extension with Azure Resource Manager templates

> **Note**
>
> DSC extension will be retired on March 31, 2028. Please move to Azure Machine Configuration by that date. For more information, see the blog post announcement. The Azure Machine Configuration service combines certain features of DSC Extension, Azure Automation State Configuration, and commonly requested features from customer feedback. Azure Machine Configuration also includes hybrid machine support through Arc-enabled servers.

This article describes the Azure Resource Manager template for the Desired State Configuration (DSC) extension handler.

> **Note**
>
> You might encounter slightly different schema examples. The change in schema occurred in the October 2016 release. For details, see Update from a previous format.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Bicep example for a virtual machine

The DSC extension inherits default extension properties. For more information, see VirtualMachineExtension class.

Bicep

```
@description('URI of the configuration package')
param configUri string

@description('first configuration parameter')
param arg1 string

@description('second configuration parameter')
```

```
@secure()
param arg2 string

var configName = split(split(configUri, '/')[4], '.')[0]

resource vm 'Microsoft.Compute/virtualMachines@2023-09-01' existing = {
  name: srvName
}

resource configuration_dscext 'Microsoft.Compute/virtualMachines/extensions@2024-07-
01' = {
  name: 'configurationname_dscext'
  parent: vm
  location: location
  properties: {
    publisher: 'Microsoft.Powershell'
    type: 'DSC'
    typeHandlerVersion: '2.77'
    autoUpgradeMinorVersion: true
    settings: {
      wmfVersion: 'latest'
      configuration: {
        url: configUri
        script: 'configurationname_dscext.ps1'
        function: 'configurationname_dscext'
      }
      configurationArguments: {
        arg1: arg1
      }
      advancedOptions: {
        rebootNodeIfNeeded: true
      }
    }
    protectedSettings: {
      configurationArguments: {
        arg2: arg2
      }
    }
  }
}
```

Bicep example for Windows virtual machine scale sets

A virtual machine scale set node has a section that includes an **extensionProfile** attribute. Under **extensions**, add the details for DSC Extension.

For the latest details about authoring templates that deploy extensions for Virtual Machine Scale Sets, see the document Microsoft.Compute virtualMachineScaleSets

The DSC extension inherits default extension properties. For more information, see VirtualMachineScaleSetExtension class.

Settings vs. protectedSettings

All settings are saved in a settings text file on the VM. Properties listed under **settings** are public properties. Public properties aren't encrypted in the settings text file. Properties listed under **protectedSettings** are encrypted with a certificate and aren't shown in plain text in the settings file on the VM.

If the configuration needs credentials, you can include the credentials in **protectedSettings**:

JSON

```
"protectedSettings": {
    "configurationArguments": {
        "parameterOfTypePSCredential1": {
               "userName": "UsernameValue1",
               "password": "PasswordValue1"
        }
    }
}
```

Example using the configuration script in Azure Storage

The following example is from the DSC extension handler overview. This example uses Resource Manager templates instead of cmdlets to deploy the extension. Save the IisInstall.ps1 configuration, place it in a .zip file (example: `iisinstall.zip`), and then upload the file in an accessible URL. This example uses Azure Blob storage, but you can download .zip files from any arbitrary location.

In the Resource Manager template, the following code instructs the VM to download the correct file, and then run the appropriate PowerShell function:

JSON

```
"settings": {
    "configuration": {
        "url": "https://demo.blob.core.windows.net/iisinstall.zip",
        "script": "IisInstall.ps1",
        "function": "IISInstall"
    }
},
"protectedSettings": {
```

```
    "configurationUrlSasToken": "odLPL/U1p9lvcnp..."
}
```

Update from a previous format

Any settings in a previous format of the extension (and which have the public properties **ModulesUrl**, **ModuleSource**, **ModuleVersion**, **ConfigurationFunction**, **SasToken**, or **Properties**) automatically adapt to the current format of the extension. They run just as they did before.

The following schema shows what the previous settings schema looked like:

JSON

```
"settings": {
    "WMFVersion": "latest",
    "ModulesUrl": "https://UrlToZipContainingConfigurationScript.ps1.zip",
    "SasToken": "SAS Token if ModulesUrl points to private Azure Blob Storage",
    "ConfigurationFunction": "ConfigurationScript.ps1\\ConfigurationFunction",
    "Properties": {
        "ParameterToConfigurationFunction1": "Value1",
        "ParameterToConfigurationFunction2": "Value2",
        "ParameterOfTypePSCredential1": {
            "UserName": "UsernameValue1",
            "Password": "PrivateSettingsRef:Key1"
        },
        "ParameterOfTypePSCredential2": {
            "UserName": "UsernameValue2",
            "Password": "PrivateSettingsRef:Key2"
        }
    }
},
"protectedSettings": {
    "Items": {
        "Key1": "PasswordValue1",
        "Key2": "PasswordValue2"
    },
    "DataBlobUri": "https://UrlToConfigurationDataWithOptionalSasToken.psd1"
}
```

Here's how the previous format adapts to the current format:

Expand table

Current Property name	Previous schema equivalent
settings.wmfVersion	settings.WMFVersion
settings.configuration.url	settings.ModulesUrl

Current Property name	Previous schema equivalent
settings.configuration.script	First part of settings.ConfigurationFunction (before \\)
settings.configuration.function	Second part of settings.ConfigurationFunction (after \\)
settings.configuration.module.name	settings.ModuleSource
settings.configuration.module.version	settings.ModuleVersion
settings.configurationArguments	settings.Properties
settings.configurationData.url	protectedSettings.DataBlobUri (without SAS token)
settings.privacy.dataCollection	settings.Privacy.dataCollection
settings.advancedOptions.downloadMappings	settings.AdvancedOptions.DownloadMappings
protectedSettings.configurationArguments	protectedSettings.Properties
protectedSettings.configurationUrlSasToken	settings.SasToken
protectedSettings.configurationDataUrlSasToken	SAS token from protectedSettings.DataBlobUri

Troubleshooting

Here are some of the errors you might run into and how you can fix them.

Invalid values

"Privacy.dataCollection is '{0}'. The only possible values are '', 'Enable', and 'Disable'".
"WmfVersion is '{0}'. Only possible values are ... and 'latest'".

Problem: A provided value isn't allowed.

Solution: Change the invalid value to a valid value.

Invalid URL

"ConfigurationData.url is '{0}'. This is not a valid URL" "DataBlobUri is '{0}'. This is not a valid URL" "Configuration.url is '{0}'. This is not a valid URL"

Problem: A provided URL isn't valid.

Solution: Check all your provided URLs. Ensure that all URLs resolve to valid locations that the extension can access on the remote machine.

Invalid ConfigurationArgument type

"Invalid configurationArguments type {0}"

Problem: The *ConfigurationArguments* property can't resolve to a **Hash table** object.

Solution: Make your *ConfigurationArguments* property a **Hash table**. Follow the format provided in the preceding examples. Watch for quotes, commas, and braces.

Duplicate ConfigurationArguments

"Found duplicate arguments '{0}' in both public and protected configurationArguments"

Problem: The *ConfigurationArguments* in public settings and the *ConfigurationArguments* in protected settings have properties with the same name.

Solution: Remove one of the duplicate properties.

Missing properties

"settings.Configuration.function requires that settings.configuration.url or settings.configuration.module is specified"

"settings.Configuration.url requires that settings.configuration.script is specified"

"settings.Configuration.script requires that settings.configuration.url is specified"

"settings.Configuration.url requires that settings.configuration.function is specified"

"protectedSettings.ConfigurationUrlSasToken requires that settings.configuration.url is specified"

"protectedSettings.ConfigurationDataUrlSasToken requires that settings.configurationData.url is specified"

Problem: A defined property needs another property, which is missing.

Solutions:

- Provide the missing property.
- Remove the property that needs the missing property.

Next steps

- Learn about using virtual machine scale sets with the Azure DSC extension.

- Find more details about DSC's secure credential management.
- Get an introduction to the Azure DSC extension handler.
- For more information about PowerShell DSC, go to the PowerShell documentation center.

Last updated on 11/10/2025

VMAccess Extension for Linux

08/27/2025

The VMAccess Extension is used to manage administrative users, configure SSH, and check or repair disks on Azure Linux virtual machines. The extension integrates with Azure Resource Manager templates. It can also be invoked using Azure CLI, Azure PowerShell, the Azure portal, and the Azure Virtual Machines REST API.

This article describes how to run the VMAccess Extension from the Azure CLI and through an Azure Resource Manager template. This article also provides troubleshooting steps for Linux systems.

> **Note**
>
> If you use the VMAccess extension to reset the password of your VM after you install the Microsoft Entra sign-in extension, rerun the Microsoft Entra sign-in extension to re-enable Microsoft Entra sign-in for your VM.

Prerequisites

Supported Linux distributions

Expand table

Publisher	Distribution	x64	ARM64
Alma Linux Community	Alma Linux	8.x+, 9.x+	8.x+, 9.x+
Credativ	Debian	10+	11.x+
Kinvolk	Flatcar Linux	3374.2.x+	3374.2.x+
Microsoft	Azure Linux	2.x	2.x
openSUSE Project	openSUSE	12.3+	*Not supported*
Oracle	Oracle Linux	6.4+, 7.x+, 8.x+	*Not supported*
Red Hat	Red Hat Enterprise Linux	6.7+, 7.x+, 8.x+, 9.x+, 10.x+	8.6+, 9.0+, 10.x+
CIQ	Rocky Linux	9.x+	9.x+

Publisher	Distribution	x64	ARM64
SUSE	SLES	12.x+, 15.x+	15.x SP4+
Canonical	Ubuntu (LTS releases)	18.04+, 20.04+, 22.04+, 24.04+	20.04+, 22.04+, 24.04+

Tips

- VMAccess was designed for regaining access to a VM given that access is lost. Based on this principle, it grants sudo permission to account specified in the username field. If you don't wish a user to gain sudo permissions, sign-in to the VM and use built-in tools (for example, usermod, change, etc.) to manage unprivileged users.
- You can only have one version of the extension applied to a VM. To run a second action, update the existing extension with a new configuration.
- During a user update, VMAccess alters the `sshd_config` file and takes a backup of it beforehand. It changes `ChallengeResponseAuthentication` to `no` and `PasswordAuthentication` to `yes`. To restore the original backed-up SSH configuration, run VMAccess with `restore_backup_ssh` set to `True`.

Extension schema

The VMAccess Extension configuration includes settings for username, passwords, SSH keys, etc. You can store this information in configuration files, specify it on the command line, or include it in an Azure Resource Manager (ARM) template. The following JSON schema contains all the properties available to use in public and protected settings.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachines/extensions",
  "name": "<name>",
  "apiVersion": "2023-09-01",
  "location": "<location>",
  "dependsOn": [
          "[concat('Microsoft.Compute/virtualMachines/', <vmName>)]"
  ],
  "properties": {
    "publisher": "Microsoft.OSTCExtensions",
    "type": "VMAccessForLinux",
    "typeHandlerVersion": "1.5",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "check_disk": true,
      "repair_disk": false,
```

```
      "disk_name": "<disk-name>",
    },
    "protectedSettings": {
      "username": "<username>",
      "password": "<password>",
      "ssh_key": "<ssh-key>",
      "reset_ssh": false,
      "remove_user": "<username>",
      "expiration": "<expiration>",
      "remove_prior_keys": false,
      "restore_backup_ssh": true
    }
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	2023-09-01	date
publisher	Microsoft.OSTCExtensions	string
type	VMAccessForLinux	string
typeHandlerVersion	1.5	int

Settings property values

Expand table

Name	Data Type	Description
check_disk	boolean	Whether or not to check disk (optional). Only one between `check_disk` and `repair_disk` can be set to true.
repair_disk	boolean	Whether or not to check disk (optional). Only one between `check_disk` and `repair_disk` can be set to true.
disk_name	string	Name of disk to repair (required when `repair_disk` is true).
username	string	The name of the user to manage (required for all actions on a user account).
password	string	The password to set for the user account.

Name	Data Type	Description
ssh_key	string	The SSH public key to add for the user account. The SSH key can be in `ssh-rsa`, `ssh-ed25519`, or `.pem` format.
reset_ssh	boolean	Whether or not to reset the SSH. If `true`, it replaces the sshd_config file with an internal resource file corresponding to the default SSH config for that distro.
remove_user	string	The name of the user to remove. Can't be used with `reset_ssh`, `restore_backup_ssh`, and `password`.
expiration	string	Expiration to set to for the account, in the form of `yyyy-mm-dd`. Defaults to never.
remove_prior_keys	boolean	Whether or not to remove old SSH keys when adding a new one. Must be used with `ssh_key`.
restore_backup_ssh	boolean	Whether or not to restore the original backed-up sshd_config.

Template deployment

Azure VM Extensions can be deployed with Azure Resource Manager (ARM) templates. The JSON schema detailed in the previous section can be used in an ARM template to run the VMAccess Extension during the template's deployment. You can find a sample template that includes the VMAccess extension on GitHub.

The JSON configuration for a virtual machine extension must be nested inside the virtual machine resource fragment of the template, specifically `"resources": []` object for the virtual machine template and for a virtual machine scale set under `"virtualMachineProfile":"extensionProfile":{"extensions" :[]` object.

Azure CLI deployment

Using Azure CLI VM user commands

The following CLI commands under az vm user use the VMAccess Extension. To use these commands, you need to install the latest Azure CLI and sign in to an Azure account by using az sign-in.

Update SSH key

The following example updates the SSH key for the user `azureUser` on the VM named `myVM`:

Azure CLI

```
az vm user update \
  --resource-group myResourceGroup \
  --name myVM \
  --username azureUser \
  --ssh-key-value ~/.ssh/id_rsa.pub
```

> **Note**
>
> The az vm user update command appends the new public key text to the `~/.ssh/authorized_keys` file for the admin user on the VM. This command doesn't replace or remove any existing SSH keys. This command doesn't remove prior keys set at deployment time or subsequent updates by using the VMAccess Extension.

Reset password

The following example resets the password for the user `azureUser` on the VM named `myVM`:

Azure CLI

```
az vm user update \
  --resource-group myResourceGroup \
  --name myVM \
  --username azureUser \
  --password myNewPassword
```

Restart SSH

The following example restarts the SSH daemon and resets the SSH configuration to default values on a VM named `myVM`:

Azure CLI

```
az vm user reset-ssh \
  --resource-group myResourceGroup \
  --name myVM
```

> **Note**

The az vm user reset-ssh command replaces the sshd_config file with a default config file from the internal resources directory. This command doesn't restore the original SSH configuration found on the virtual machine.

Create an administrative/sudo user

The following example creates a user named `myNewUser` with sudo permissions. The account uses an SSH key for authentication on the VM named `myVM`. This method helps you regain access to a VM when current credentials are lost or forgotten. As a best practice, accounts with sudo permissions should be limited.

Azure CLI

```
az vm user update \
  --resource-group myResourceGroup \
  --name myVM \
  --username myNewUser \
  --ssh-key-value ~/.ssh/id_rsa.pub
```

Delete a user

The following example deletes a user named `myNewUser` on the VM named `myVM`:

Azure CLI

```
az vm user delete \
  --resource-group myResourceGroup \
  --name myVM \
  --username myNewUser
```

Using Azure CLI VM/Virtual Machine Scale Sets extension commands

You can also use the az vm extension set and az vmss extension set commands to run the VMAccess Extension with the specified configuration.

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name VMAccessForLinux \
  --publisher Microsoft.OSTCExtensions \
```

```
  --version 1.5 \
  --settings '{"check_disk":true}'
  --protected-settings '{"username":"user1","password":"userPassword"}'
```

The `--settings` and `--protected-settings` parameters also accept JSON file paths. For example, to update the SSH public key of a user, create a JSON file named `update_ssh_key.json` and add settings in the following format. Replace the values within the file with your own information:

JSON

```
{
  "username":"azureuser",
  "ssh_key":"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCZ3S7gGp3rcbKmG2Y4vGZFMuMZCwoUzZNGxxxxxx2XV2x9FfAhy8
iGD+lF8UdjFX3t5ebMm6BnnMh8fHwkTRdOt3LDQq8o8ElTBrZaKPxZN2thMZnODs5Hlemb2UX0oRIGRcvW
qsd4oJmxsXa/Si98Wa6RHWbc9QZhw80KAcOVhmndZAZAGR+Wq6yslNo5TMOr1/ZyQAook5C4FtcSGn3Y+W
czaoGWIxG4ZaWk128g79VIeJcIQqOjPodHvQAhll7qDlItVvBfMOben3GyhYTm7k4YwlEdkONm4yV/UIW0
la1rmyztSBQIm9sZmSq44XXgjVmDHNF8UfCZ1ToE4r2SdwTmZv00T2i5faeYnHzxiLPA3Enub7xxxxxxwF
Arnqad7MO1SY1kLemhX9eFjLWN4mJe56Fu4NiWJkR9APSZQrYeKaqru4KUC68QpVasNJHbuxPSf/PcjF3c
jO1+X+4x6L1H5HTPuqUkyZGgDO4ynUHbko4dhlanALcriF7tIfQR9i2r2xOyv5gxJEW/zztGqWma/d4rBo
Pjnf6tO7rLFHXMt/DVTkAfn5wxxtLDwkn5FMyvThRmex3BDf0gujoI1y6cOWLe9Y5geNX0oj+MXg/W0cXA
tzSFocstV1PoVqy883hNoeQZ3mIGB3Q0rIUm5d9MA2bMMt31m1g3Sin6EQ== azureuser@myVM"
}
```

Run the VMAccess Extension through the following command:

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myVM \
  --name VMAccessForLinux \
  --publisher Microsoft.OSTCExtensions \
  --version 1.5 \
  --protected-settings update_ssh_key.json
```

Azure PowerShell deployment

Azure PowerShell can be used to deploy the VMAccess Extension to an existing virtual machine or virtual machine scale set. You can deploy the extension to a VM by running:

Azure PowerShell

```
$username = "<username>"
$sshKey = "<cert-contents>"
```

```
$settings = @{"check_disk" = $true};
$protectedSettings = @{"username" = $username; "ssh_key" = $sshKey};

Set-AzVMExtension -ResourceGroupName "<resource-group>" `
    -VMName "<vm-name>" `
    -Location "<location>" `
    -Publisher "Microsoft.OSTCExtensions" `
    -ExtensionType "VMAccessForLinux" `
    -Name "VMAccessForLinux" `
    -TypeHandlerVersion "1.5" `
    -Settings $settings `
    -ProtectedSettings $protectedSettings
```

You can also provide and modify extension settings by using strings:

Azure PowerShell

```
$username = "<username>"
$sshKey = "<cert-contents>"

$settingsString = '{"check_disk":true}';
$protectedSettingsString = '{"username":"' + $username + '","ssh_key":"' + $sshKey
+ '"}';

Set-AzVMExtension -ResourceGroupName "<resource-group>" `
    -VMName "<vm-name>" `
    -Location "<location>" `
    -Publisher "Microsoft.OSTCExtensions" `
    -ExtensionType "VMAccessForLinux" `
    -Name "VMAccessForLinux" `
    -TypeHandlerVersion "1.5" `
    -SettingString $settingsString `
    -ProtectedSettingString $protectedSettingsString
```

To deploy to a virtual machine scale set, run the following command:

Azure PowerShell

```
$resourceGroupName = "<resource-group>"
$vmssName = "<vmss-name>"

$protectedSettings = @{
  "username" = "azureUser"
  "password" = "userPassword"
}

$publicSettings = @{
  "repair_disk" = $true
  "disk_name" = "<disk_name>"
}

$vmss = Get-AzVmss `
```

```
            -ResourceGroupName $resourceGroupName `
            -VMScaleSetName $vmssName

Add-AzVmssExtension -VirtualMachineScaleSet $vmss `
    -Name "<extension-name>" `
    -Publisher "Microsoft.OSTCExtensions" `
    -Type "VMAccessForLinux" `
    -TypeHandlerVersion "1.5"" `
    -AutoUpgradeMinorVersion $true `
    -Setting $publicSettings `
    -ProtectedSetting $protectedSettings

Update-AzVmss `
    -ResourceGroupName $resourceGroupName `
    -Name $vmssName `
    -VirtualMachineScaleSet $vmss
```

Troubleshoot and support

The VMAccess extension logs exist locally on the VM and are most informative when it comes to troubleshooting.

Expand table

Location	Description
/var/log/waagent.log	Contains logs from the Linux Agent and shows when an update to the extension occurred. We can check it to ensure the extension ran.
/var/log/azure/Microsoft.OSTCExtensions.VMAccessForLinux/*	The VMAccess Extension produces logs, which can be found here. The directory contains `CommandExecution.log` where you can find each command executed along with its result, along with `extension.log`, which contains individual logs for each execution.
/var/lib/waagent/Microsoft.OSTCExtensions.VMAccessForLinux-<most recent version>/config/*	The configuration and binaries for VMAccess VM Extension.

You can also retrieve the execution state of the VMAccess Extension, along with other extensions on a given VM, by running the following command:

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myVM -o table
```

Error messages

Expand table

Error	Description
Provisioning of VM extension enablevmaccess has timed out. Extension provisioning has taken too long to complete. The extension didn't report a message. More information on troubleshooting is available at https://aka.ms/vmextensionlinuxtroubleshoot.	The error message indicates that the provisioning of the VM extension 'enablevmaccess' has timed out due to taking too long to complete. Additionally, the extension didn't provide any status message during the process. To resolve this issue, consider checking the VM's performance and network conditions, and retry the provisioning operation. For more information, see Troubleshoot VM extensions.
VM has reported a failure when processing extension 'enablevmaccess' (publisher 'Microsoft.OSTCExtensions' and type 'VMAccessForLinux'). Error message: 'Enable failed: No password or ssh_key is specified.'. More information on troubleshooting is available at https://aka.ms/vmextensionlinuxtroubleshoot.	The error message indicates that the VM failed to process the 'enablevmaccess' extension because no password or SSH key was specified. This failure is associated with the 'Microsoft.OSTCExtensions' publisher and the 'VMAccessForLinux' type. To resolve this issue, ensure that either a password or an SSH key is provided during the extension configuration.

For more help, you can contact the Azure experts at Azure Community Support. Alternatively, you can file an Azure support incident. Go to Azure support and select **Get support**. For more information about Azure Support, read the Azure support plans FAQ.

Next steps

To see the code, current versions, and more documentation, see VMAccess Linux - GitHub.

VMAccess Extension for Windows

The VMAccess Extension is used to manage administrative users, configure RDP, and check or repair disks on Azure Windows virtual machines. The extension integrates with Azure Resource Manager templates. It can also be invoked using Azure CLI, Azure PowerShell, the Azure portal, and the Azure Virtual Machines REST API.

This article describes how to run the VMAccess Extension from the Azure PowerShell and through an Azure Resource Manager template. This article also provides troubleshooting steps for Windows systems.

> **Note**
>
> If you use the VMAccess extension to reset the password of your VM after you install the Microsoft Entra Login extension, rerun the Microsoft Entra Login extension to re-enable Microsoft Entra Login for your VM.

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.
>
> If you're experiencing performance problems with virtual machines, before contacting support, run these tools.

Prerequisites

Supported Windows versions

Expand table

OS Version	x64	ARM64
Windows 10	Supported	Supported
Windows 11	Supported	Supported
Windows Server 2016	Supported	Supported
Windows Server 2016 Core	Supported	Supported

OS Version	x64	ARM64
Windows Server 2019	Supported	Supported
Windows Server 2019 Core	Supported	Supported
Windows Server 2022	Supported	Supported
Windows Server 2022 Core	Supported	Supported
Windows Server 2025	Supported	Supported
Windows Server 2025 Core	Supported	Supported

Tips

- VMAccess is designed for regaining access to a VM when access is lost. Based on this principle, it grants administrator privileges to the specified account in the username field. If you don't want a user to have admin permissions, log in to the VM and use built-in tools (such as `net user` and `Local Users and Groups`) to manage user privileges.
- You can only have one version of the extension applied to a VM. To run a second action, update the existing extension with a new configuration.
- When updating a user, VMAccess modifies the Remote Desktop settings to allow login.
- Not supported on Domain Controllers

Extension schema

The VMAccess Extension configuration includes settings for username, passwords, and resetting the administrator password. You can store this information in configuration files, specify it in PowerShell, or include it in an Azure Resource Manager (ARM) template. The following JSON schema contains all the properties available to use in public and protected settings.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachines/extensions",
  "name": "<name>",
  "apiVersion": "2023-09-01",
  "location": "<location>",
  "dependsOn": [
          "[concat('Microsoft.Compute/virtualMachines/', <vmName>)]"
  ],
  "properties": {
    "publisher": "Microsoft.Compute",
    "type": "VMAccessAgent",
```

```
    "typeHandlerVersion": "2.4",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "username": "<username>"
    },
    "protectedSettings": {
      "password": "<password>"
    }
  }
}
```

Property values

Expand table

Name	Value / Example	Data Type
apiVersion	2023-09-01	date
publisher	Microsoft.Compute	string
type	VMAccessAgent	string
typeHandlerVersion	2.4	int

Settings property values

Expand table

Name	Data Type	Description
username	string	The name of the user to manage (required for all actions on a user account).
password	string	The password to set for the user account.

Template deployment

Azure VM Extensions can be deployed with Azure Resource Manager (ARM) templates. The JSON schema detailed in the previous section can be used in an ARM template to run the VMAccess Extension during the template's deployment.

The JSON configuration for a virtual machine extension must be nested inside the virtual machine resource fragment of the template, specifically `"resources": []` object for the virtual

machine template and for a virtual machine scale set under `"virtualMachineProfile":"extensionProfile":{"extensions" :[]` object.

PowerShell

Set-AzVMAccessExtension - Reset Password

Set-AzVMAccessExtension

PowerShell

```
Set-AzVMAccessExtension `
    -ResourceGroupName "myRG" `
    -Location "myLocation" `
    -VMName "myVM" `
    -Credential (get-credential) `
    -typeHandlerVersion "2.0" `
    -Name VMAccessAgent
```

Set-AzVMAccessExtension - Reset RDP Configuration

PowerShell

```
Set-AzVMAccessExtension `
    -ResourceGroupName "myRG" `
    -VMName "myVM" `
    -Name "myVMAccess" `
    -Location "myLocation" `
    -typeHandlerVersion "2.0" `
    -ForceRerun $true
```

Set-AzVMExtension - Reset Password

Set-AzVMExtension

PowerShell

```
$Publicsettings = '{"UserName": "myuser"}'
$protectedSettings = '{"Password": "myPassWord"}'

Set-AzVMExtension `
  -ResourceGroupName "myRG" `
  -VMName "myVM" `
  -Location "myLocation" `
  -Name "VMAccessAgent" `
  -Publisher "Microsoft.Compute" `
```

```
  -ExtensionType "VMAccessAgent" `
  -TypeHandlerVersion "2.0" `
  -ProtectedSettingString $PrivateConf `
  -SettingString $settings
```

Set-AzVMExtension - Reset RDP Configuration

Set-AzVMExtension

PowerShell

```
Set-AzVMExtension `
  -ResourceGroupName "myRG" `
  -VMName "myVM" `
  -Location "myLocation" `
  -Name "VMAccessAgent" `
  -Publisher "Microsoft.Compute" `
  -ExtensionType "VMAccessAgent" `
  -TypeHandlerVersion "2.0" `
  -SettingString '{}' `
  -ForceRerun $true
```

CLI

Azure CLI - Reset Password

az vm extension set

```
az vm extension set --name VMAccessAgent --publisher Microsoft.Compute --version 2.0
--vm-name "myVM" --resource-group "myRG" --settings '{"username":"myuser"}' --
protected-settings '{"password":"myPassWord"}'
```

Azure CLI - Reset RDP Configuration

```
az vm extension set --name VMAccessAgent --publisher Microsoft.Compute --version 2.0
--vm-name "myVM" --resource-group "myRG" --settings '{}'
```

Virtual machine scale sets

PowerShell

PowerShell

```
$resourceGroupName = "<resource-group>"
$vmssName = "<vmss-name>"

$settings = @{"UserName"= "myuser"}
$protectedSettings = @{"Password"= "myPassWord"}

$vmss = Get-AzVmss `
    -ResourceGroupName $resourceGroupName `
    -VMScaleSetName $vmssName

Add-AzVmssExtension -VirtualMachineScaleSet $vmss `
    -Name "VMAccessAgent" `
    -Publisher "Microsoft.Compute" `
    -Type "VMAccessAgent" `
    -TypeHandlerVersion "2.0" `
    -AutoUpgradeMinorVersion $true `
    -ProtectedSetting $protectedSettings
    -Setting $settings

Update-AzVmss `
    -ResourceGroupName $resourceGroupName `
    -Name $vmssName `
    -VirtualMachineScaleSet $vmss
```

CLI

```
az vmss extension set --vmss-name my-vmss --name VMAccessAgent --resource-group my-group --version 2.0 --publisher Microsoft.Compute --settings '{"username":"myuser"}' --protected-settings '{"password":"myPassWord"}'
```

Troubleshoot and support

The VMAccess extension logs exist locally on the VM and are particularly useful for troubleshooting.

Expand table

Location	Description
`C:\WindowsAzure\Logs\Plugins\Microsoft.Compute.VMAccessAgent\`	Contains logs from the Windows Agent that show when an update to the extension occurred. Verify these

Location	Description
	logs to ensure the extension ran successfully.
`C:\WindowsAzure\Logs\Plugins\Microsoft.Compute.VMAccessAgent\<version>\`	The VMAccess Extension produces detailed logs in this directory. It includes `CommandExecution.log`, which records each command executed along with its result, and `extension.log`, which contains individual execution logs.
`C:\WindowsAzure\Logs\WaAppAgent\`	Contains configuration details and binary logs related to the VMAccess extension.

You can also retrieve the execution state of the VMAccess Extension by running the following command:

PowerShell

```
Get-AzVMExtension -ResourceGroupName "RG Name" -VMName "VM Name" -Name "Extension Name"
```

Error messages

Expand table

Error	Description
"internalErrorCode": "CannotModifyExtensionsWhenVMNotRunning", "code": "OperationNotAllowed","message": "Cannot modify extensions in the VM when the VM is not running."	The error indicates that the operation to modify extensions in the VM is not allowed because the VM is not running. Ensure that the VM is in a running state before attempting to modify extensions.
Error message: 'VMAccess Extension does not support Domain Controller.'	The error indicates that the VM extension 'enablevmAccess' failed because it does not support Domain Controller. Ensure that the VM is not configured as a Domain Controller when using this extension. For more information, see Reset Remote Desktop Services or its administrator password in a Windows VM.
VM 'vmname' has not reported status for VM agent or extensions. Verify that the OS is up and healthy, the VM	See Troubleshoot Azure Windows VM Agent issues.

Error	Description
has a running VM agent, and that it can establish outbound connections to Azure storage.	
Error message: 'Cannot update Remote Desktop Connection settings for Administrator account. The password does not meet the password policy requirements. Check the minimum password length, password complexity, and password history requirements.	The error indicates that the VM extension 'enablevmAccess' failed to update Remote Desktop Connection settings for the Administrator account due to a password policy violation. Ensure that the password meets Windows password policy requirements, including minimum length, complexity, and history.
The Admin User Account password cannot be null or empty if provided the username.	The error indicates that the VM extension 'enablevmAccess' failed because the Admin User Account password was not provided. Ensure that a non-null and non-empty password is specified for the Admin User Account to resolve this issue.
Provisioning of VM extension enablevmaccess has timed out. Extension provisioning has taken too long to complete. The extension did not report a message.	The error message indicates that the provisioning of the VM extension 'enablevmaccess' has timed out due to taking too long to complete. Additionally, the extension did not provide any status message during the process. To resolve this issue, consider checking the VM's performance and network conditions, and retry the provisioning operation. For more information, see Troubleshooting Azure Windows VM extension failures.
Error message: 'Cannot update Remote Desktop Connection settings for Administrator account. Error: User account username already exists but cannot be updated because it is not in the Administrators group.	The error indicates that the VM extension 'enablevmAccess' failed because the user account 'username' already exists but is not in the Administrators group. Ensure that the user account is added to the Administrators group to resolve this issue.
"internalErrorCode": "MultipleExtensionsPerHandlerNotAllowed", "code": "BadRequest","message": "Multiple VMExtensions per handler not supported for OS type 'Windows'. VMExtension 'enablevmaccess' with handler 'Microsoft.Compute.VMAccessAgent' already added or specified in input."	The error message indicates that multiple VM extensions per handler are not supported for the OS type 'Windows'. The 'enablevmaccess' extension with the handler 'Microsoft.Compute.VMAccessAgent' has already been added or specified in the input. To resolve this issue, ensure that only one extension per handler is configured for the VM. Manually remove extension and retry operation `Remove-AzVMExtension -ResourceGroupName`

Error	Description
	`"ResourceGroup11" -Name "ExtensionName" -VMName "VirtualMachineName"`

For more help, you can contact the Azure experts at Azure Community Support. Alternatively, you can file an Azure support incident. Go to Azure support and select **Get support**. For more information about Azure Support, read the Azure support plans FAQ.

Last updated on 11/10/2025

Chef VM Extension for Linux and Windows

Chef Software provides a DevOps automation platform for Linux and Windows that enables the management of both physical and virtual server configurations. The Chef VM Extension is an extension that enables Chef on virtual machines.

> ⓘ Note
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

Operating system

The Chef VM Extension is supported on all the Extension Supported OS's in Azure.

Internet connectivity

The Chef VM Extension requires that the target virtual machine is connected to the internet in order to retrieve the Chef Client payload from the content delivery network (CDN).

Extension schema

The following JSON shows the schema for the Chef VM Extension. The extension requires at a minimum the Chef Server URL, the Validation Client Name and the Validation Key for the Chef Server; these values can be found in the `knife.rb` file in the starter-kit.zip that is downloaded when you install Chef Automate or a standalone Chef Server. Because the validation key should be treated as sensitive data, it should be configured under the **protectedSettings** element, meaning that it will only be decrypted on the target virtual machine.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachines/extensions",
  "name": "[concat(variables('vmName'),'/', parameters('chef_vm_extension_type'))]",
  "apiVersion": "2017-12-01",
  "location": "[parameters('location')]",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
  ],
  "properties": {
    "publisher": "Chef.Bootstrap.WindowsAzure",
```

```
    "type": "[parameters('chef_vm_extension_type')]",
    "typeHandlerVersion": "1210.13",
    "settings": {
      "bootstrap_options": {
        "chef_server_url": "[parameters('chef_server_url')]",
        "validation_client_name": "[parameters('chef_validation_client_name')]"
      },
      "runlist": "[parameters('chef_runlist')]"
    },
    "protectedSettings": {
      "validation_key": "[parameters('chef_validation_key')]"
    }
  }
}
```

Core property values

Expand table

Name	Value / Example	Data Type
apiVersion	`2017-12-01`	string (date)
publisher	`Chef.Bootstrap.WindowsAzure`	string
type	`LinuxChefClient` (Linux), `ChefClient` (Windows)	string
typeHandlerVersion	`1210.13`	string (double)

Settings

Expand table

Name	Value / Example	Data Type	Required?
settings/bootstrap_options/chef_server_url	`https://api.chef.io/organizations/myorg`	string (url)	Y
settings/bootstrap_options/validation_client_name	`myorg-validator`	string	Y
settings/runlist	`recipe[mycookbook::default]`	string	Y

Protected settings

Expand table

Name	Example	Data Type	Required?
protectedSettings/validation_key	`-----BEGIN RSA PRIVATE KEY-----\nKEYDATA\n----- END RSA PRIVATE KEY-----`	string	Y

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. Templates can be used to deploy one or more virtual machines, install the Chef Client, connect to the Chef Server and perform the initial configuration on the server as defined by the Run-list

A sample Resource Manager template that includes the Chef VM Extension can be found in the Azure quickstart gallery .

The JSON configuration for a virtual machine extension can be nested inside the virtual machine resource, or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON configuration affects the value of the resource name and type. For more information, see Set name and type for child resources.

Azure CLI deployment

The Azure CLI can be used to deploy the Chef VM Extension to an existing VM. Replace the **validation_key** with the contents of your validation key (this file as a `.pem` extension). Replace **validation_client_name**, **chef_server_url** and **run_list** with those values from the `knife.rb` file in your Starter Kit.

Azure CLI

```
az vm extension set \
  --resource-group myResourceGroup \
  --vm-name myExistingVM \
  --name LinuxChefClient \
  --publisher Chef.Bootstrap.WindowsAzure \
  --version 1210.13 --protected-settings '{"validation_key": "<validation_key>"}' \
  --settings '{ "bootstrap_options": { "chef_server_url": "<chef_server_url>",
"validation_client_name": "<validation_client_name>" }, "runlist": "<run_list>" }'
```

Troubleshooting and support

Data about the state of extension deployments can be retrieved from the Azure portal, and by using the Azure CLI. To see the deployment state of extensions for a given VM, run the following command using the Azure CLI.

Azure CLI

```
az vm extension list --resource-group myResourceGroup --vm-name myExistingVM -o table
```

Extension execution output is logged to the following file:

Linux

Bash

```
/var/lib/waagent/Chef.Bootstrap.WindowsAzure.LinuxChefClient
```

Windows

PowerShell

```
C:\Packages\Plugins\Chef.Bootstrap.WindowsAzure.ChefClient\
```

Error codes and their meanings

Expand table

Error Code	Meaning	Possible Action
51	This extension is not supported on the VM's operating system	

Additional troubleshooting information can be found in the Chef VM Extension readme.

> **Note**
>
> For anything else directly related to Chef, contact Chef Support.

Next steps

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure Support, read the Microsoft Azure support FAQ.

Last updated on 11/10/2025

Qualys Agent on Linux or Windows VM

> **Important**
>
> Qualys has removed support for the Windows Cloud Agent extension version 3.1.3.34. This version will become inoperable starting January 31, 2026. We recommend that you uninstall it and install the latest Windows Cloud Agent extension version (1.6.x.x) on your Azure portal. To learn more about the uninstallation steps, refer to Uninstalling and Reinstalling Qualys Extension on Azure VMs.

The Qualys Cloud Agent is a lightweight, cloud-based security and monitoring tool installed on servers, virtual machines (VMs), and workstations. It continuously collects system data and sends it to Qualys Enterprise TruRisk Platform (ETP) for analysis and reporting.

Qualys supports Cloud Agent deployment on Windows and Linux Azure VMs via Microsoft's Azure portal. The Cloud Agent Platform Availability Matrix provides more information about the supported platforms.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Installation Methods

The following are the different deployment methods to deploy Cloud Agent extensions on Azure VMs.

Deploying Qualys Cloud Agent for Single Azure VM

The Microsoft Azure platform has the option to deploy a Cloud Agent extension for a single Azure VM. To learn more about the deployment steps, refer to Deploy Cloud Agent for Single Azure VM.

Embedding Qualys Cloud Agent via Golden Machine Image

The Qualys Cloud Agent supports configuration and deployment into cloned images in cloud environments like Microsoft Azure.

To learn more about embedding Qualys Cloud Agent as a part of Golden Machine Image, refer to the following links:

- **For Windows VMs:** Steps to Install Qualys Cloud Agent for Windows in Gold Image
- **For Linux VMs:** Steps to Install Qualys Cloud Agent for Linux in Gold Image

Deploy Qualys Cloud Agent via Azure Resource Manager Template

Qualys supports the Azure Resource Manager (ARM) templates to deploy and install Qualys Cloud Agent as a virtual machine extension on a list of Azure Windows and Linux VMs. VM extensions can be used whenever a virtual machine requires software installation, anti-virus protection, or running a script inside of it.

To learn more about deployment steps, refer to Cloud Agent Deployment via ARM Template.

Deploy Qualys Cloud Agent via Azure Deployment Policy

Qualys supports the Azure Deployment Policy to schedule auto-deployment of Cloud Agent Extensions on Windows and Linux VMs.

To learn more about deployment steps, refer to:

- **For Windows VM:** Deploy Cloud Agent for Windows VMs via Azure Deployment Policy
- **For Linux VM:** Deploy Cloud Agent for Linux VMs via Azure Deployment Policy

Deploy Cloud Agent via PowerShell

Qualys supports the PowerShell command-line tool to deploy Cloud Agent Extensions on Windows and Linux VMs. To learn more about deployment steps, refer to:

- **For Windows VM:** Deploy Cloud Agent For Windows via PowerShell
- **For Linux VM:** Deploy Cloud Agent For Linux via PowerShell

Deploy Qualys Cloud Agent via Other Tool Sets

Qualys Cloud Agent can be deployed via automation, orchestration, or configuration management tools in your environment, such as Ansible, Chef, and Puppet. Qualys provides a template for deploying Qualys Cloud Agent via Ansible. You can use this to deploy and configure Qualys Cloud Agent in their Azure environment.

To learn more, refer to Cloud Agent Deployment via Other Tool Sets.

Support

Contact Qualys Support if you encounter an issue when deploying Cloud Agent on Azure VMs.

Last updated on 11/10/2025

Install Salt Minion on Linux or Windows VMs using the VM Extension

Prerequisites

- A Microsoft Azure account with one (or more) Windows or Linux VMs
- A Salt Master (either on-premises or in a cloud) that can accept connections from Salt minions hosted on Azure
- The Salt Minion VM Extension requires that the target VM is connected to the internet in order to fetch Salt packages

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Supported platforms

Azure VM running any of the following supported OS:

- Ubuntu 20.04, 22.04 (x86_64)
- Debian 10, 11 (x86_64)
- Oracle Linux 7, 8, 9 (x86_64)
- RHEL 7, 8, 9 (x86_64)
- Microsoft Windows 10, 11 Pro (x86_64)
- Microsoft Windows Server 2012 R2, 2016, 2019, 2022 Datacenter (x86_64)

If you want another distro to be supported (assuming Salt supports it), an issue can be filed on GitLab.

Supported Salt Minion versions

- 3006 and up (onedir)

Extension details

- Publisher name: `turtletraction.oss`

- Linux extension name: `salt-minion.linux`
- Windows extension name: `salt-minion.windows`

Salt Minion settings

- `master_address` - Salt Master address to connect to (`localhost` by default)
- `minion_id` - Minion ID (hostname by default)
- `salt_version` - Salt Minion version to install, for example `3006.1` (`latest` by default)

Install Salt Minion using the Azure portal

1. Select one of your VMs.
2. In the left menu click **Extensions + applications**.
3. Click **+ Add**.
4. In the gallery, type **Salt Minion** in the search bar.
5. Select the **Salt Minion** tile and click **Next**.
6. Enter configuration parameters in the provided form (see Salt Minion settings).
7. Click **Review + create**.

Install Salt Minion using the Azure CLI

shell

```
az vm extension set --resource-group my-group --vm-name vm-ubuntu22 --name salt-minion.linux --publisher turtletraction.oss --settings '{"master_address": "10.x.x.x"}'
az vm extension set --resource-group my-group --vm-name vm-windows11 --name salt-minion.windows --publisher turtletraction.oss --settings '{"master_address": "10.x.x.x"}'
```

To uninstall it:

shell

```
az vm extension delete --resource-group my-group --vm-name vm-ubuntu22 --name salt-minion.linux
az vm extension delete --resource-group my-group --vm-name vm-windows11 --name salt-minion.windows
```

Install Salt Minion using the Azure ARM template

JSON

```
{
    "$schema": "http://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "vmName": {
            "type": "string"
        },
        "master_address": {
            "type": "string"
        },
        "salt_version": {
            "type": "string"
        },
        "minion_id": {
            "type": "string"
        }
    },
    "resources": [
        {
            "name": "[concat(parameters('vmName'),'/salt-minion.linux')]",
            "type": "Microsoft.Compute/virtualMachines/extensions",
            "location": "[resourceGroup().location]",
            "apiVersion": "2015-06-15",
            "properties": {
                "publisher": "turtletraction.oss",
                "type": "salt-minion.linux",
                "typeHandlerVersion": "1.0",
                "autoUpgradeMinorVersion": true,
                "settings": {
                    "master_address": "[parameters('master_address')]",
                    "salt_version": "[parameters('salt_version')]",
                    "minion_id": "[parameters('minion_id')]"
                }
            }
        }
    ]
}
```

Install Salt Minion using Terraform

Assuming that you have defined a VM resource in TerraForm named `vm_ubuntu`, then use something like this to install the extension on it:

HashiCorp Configuration Language

```
resource "azurerm_virtual_machine_extension" "vmext_ubuntu" {
  name                 = "vmext_ubuntu"
  virtual_machine_id   = azurerm_linux_virtual_machine.vm_ubuntu.id
  publisher            = "turtletraction.oss"
```

```
  type                 = "salt-minion.linux"
  type_handler_version = "1.0"

  settings = <<SETTINGS
{
  "salt_version": "3006.1",
  "master_address": "x.x.x.x",
  "minion_id": "ubuntu22"
}
SETTINGS
}
```

Support

- For commercial support or assistance with Salt, you can visit the extension creator, TurtleTraction
- The source code of this extension is available on GitLab
- For Azure related issues, you can file an Azure support incident. Go to the Azure support site and select Get support

---

Last updated on 11/10/2025

Stackify Retrace Linux Agent Extension

> ⊗ **Caution**
>
> This article references CentOS, a Linux distribution that is end of support status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Overview

Stackify provides products that track details about your application to help find and fix problems quickly. For developer teams, Retrace is a fully integrated, multi-environment, app performance super-power. It combines several tools every development team needs.

Retrace is the ONLY tool that delivers all of the following capabilities across all environments in a single platform.

- Application performance management (APM)
- Application and server logging
- Error tracking and monitoring
- Server, application, and custom metrics

> ⓘ **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

About Stackify Linux Agent Extension

This extension provides an install path for the Linux Agent for Retrace.

Prerequisites

Operating system

The Retrace agent can be run against these Linux distributions

Expand table

Distribution	Version
Ubuntu	16.04 LTS
Debian	9
Red Hat	6.10, 7.1+
CentOS	6.10, 7.0+

ⓘ Important

> Keep in consideration Red Hat Enterprise Linux 6.X is already deprecated. RHEL 6.10 has available ELS support, which will end on 06/2024.

Internet connectivity

The Stackify Agent extension for Linux requires that the target virtual machine is connected to the internet.

You may need to adjust your network configuration to allow connections to Stackify, see https://support.stackify.com/hc/en-us/articles/207891903-Adding-Exceptions-to-a-Firewall.

Extension schema

The following JSON shows the schema for the Stackify Retrace Agent extension. The extension requires the `environment` and `activationKey`.

JSON

```
{
  "type": "extensions",
  "name": "StackifyExtension",
  "apiVersion": "[variables('apiVersion')]",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[resourceId('Microsoft.Compute/virtualMachines',variables('vmName'))]"
  ],
  "properties": {
    "publisher": "Stackify.LinuxAgent.Extension",
    "type": "StackifyLinuxAgentExtension",
    "typeHandlerVersion": "1.0",
    "autoUpgradeMinorVersion": true,
    "settings": {
```

```
        "environment": "myEnvironment"
      },
      "protectedSettings": {
        "activationKey": "myActivationKey"
      }
    }
  }
```

Template deployment

Azure VM extensions can be deployed with Azure Resource Manager templates. The JSON schema detailed in the previous section can be used in an Azure Resource Manager template to run the Stackify Retrace Linux Agent extension during an Azure Resource Manager template deployment.

The JSON for a virtual machine extension can be nested inside the virtual machine resource, or placed at the root or top level of a Resource Manager JSON template. The placement of the JSON affects the value of the resource name and type. For more information, see Set name and type for child resources.

The following example assumes the Stackify Retrace Linux extension is nested inside the virtual machine resource. When nesting the extension resource, the JSON is placed in the "resources": [] object of the virtual machine.

The extension requires the `environment` and `activationKey`.

JSON

```
  {
    "type": "extensions",
    "name": "StackifyExtension",
    "apiVersion": "[variables('apiVersion')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
      "[resourceId('Microsoft.Compute/virtualMachines',variables('vmName'))]"
    ],
    "properties": {
      "publisher": "Stackify.LinuxAgent.Extension",
      "type": "StackifyLinuxAgentExtension",
      "typeHandlerVersion": "1.0",
      "autoUpgradeMinorVersion": true,
      "settings": {
        "environment": "myEnvironment"
      },
      "protectedSettings": {
        "activationKey": "myActivationKey"
      }
```

```
    }
}
```

When placing the extension JSON at the root of the template, the resource name includes a reference to the parent virtual machine, and the type reflects the nested configuration.

JSON

```
{
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "name": "<parentVmResource>/StackifyExtension",
    "apiVersion": "[variables('apiVersion')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
    ],
    "properties": {
        "publisher": "Stackify.LinuxAgent.Extension",
        "type": "StackifyLinuxAgentExtension",
        "typeHandlerVersion": "1.0",
        "autoUpgradeMinorVersion": true,
        "settings": {
          "environment": "myEnvironment"
        },
        "protectedSettings": {
          "activationKey": "myActivationKey"
        }
    }
}
```

PowerShell deployment

The `Set-AzVMExtension` command can be used to deploy the Stackify Retrace Linux Agent virtual machine extension to an existing virtual machine. Before running the command, the public and private configurations need to be stored in a PowerShell hash table.

The extension requires the `environment` and `activationKey`.

PowerShell

```
$PublicSettings = @{"environment" = "myEnvironment"}
$ProtectedSettings = @{"activationKey" = "myActivationKey"}

Set-AzVMExtension -ExtensionName "Stackify.LinuxAgent.Extension" `
    -ResourceGroupName "myResourceGroup" `
    -VMName "myVM" `
    -Publisher "Stackify.LinuxAgent.Extension" `
    -ExtensionType "StackifyLinuxAgentExtension" `
    -TypeHandlerVersion 1.0 `
```

```
    -Settings $PublicSettings `
    -ProtectedSettings $ProtectedSettings `
    -Location WestUS `
```

Azure CLI deployment

The Azure CLI tool can be used to deploy the Stackify Retrace Linux Agent virtual machine extension to an existing virtual machine.

The extension requires the `environment` and `activationKey`.

Azure CLI

```
az vm extension set --publisher 'Stackify.LinuxAgent.Extension' --version 1.0 --name 'StackifyLinuxAgentExtension' --protected-settings '{"activationKey":"myActivationKey"}' --settings '{"environment":"myEnvironment"}' --resource-group 'myResourceGroup' --vm-name 'myVmName'
```

Troubleshoot and support

Error codes

Expand table

Error code	Meaning	Possible action
10	Install Error	wget is required
20	Install Error	Python is required
30	Install Error	sudo is required
40	Install Error	activationKey is required
51	Install Error	OS distro not supported
60	Install Error	environment is required
70	Install Error	Unknown
80	Enable Error	Service setup failed
90	Enable Error	Service startup failed
100	Disable Error	Service Stop Failed
110	Disable Error	Service Removal Failed

Error code	Meaning	Possible action
120	Uninstall Error	Service Stop Failed

If you need more help you can contact Stackify support at https://support.stackify.com.

Last updated on 11/10/2025

Tenable One-Click Nessus Agent

Tenable now supports a One-Click deployment of Nessus Agents via Microsoft's Azure portal. This solution provides an easy way to install the latest version of Nessus Agent on Azure virtual machines (VM) (whether Linux or Windows) by either clicking on an icon within the Azure portal or by writing a few lines of PowerShell script.

> ⓘ **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Prerequisites

- A Tenable Vulnerability Management (Tenable.io), or Nessus Manager, account.
- A Microsoft Azure account with one (or more) Windows or Linux VMs.

Supported Platforms

Azure VM running any of the following:

- Debian 11 (x86_64)
- Oracle Linux 7 and 8 (x86_64)
- Red Hat ES 7, 8 and 9 (x86_64)
- Rocky Linux 9 (x86_64)
- Ubuntu 18.04, 20.04 and 22.04 (x86_64)
- Red Hat ES 8 and 9 (ARM64)
- Windows 10 and 11 (x86_64)
- Windows Server 2012 and 2012 R2 (x86_64)
- Windows Server 2016, 2019 and 2022 (x86_64)

Deploy with the Tenable User Interface (UI)

1. Select one of your VMs.
2. In the left column click **Extensions + applications**.
3. Click **+ Add**.
4. In the gallery, scroll down to **N** (for Nessus Agent) or type **nessus** in the search bar.
5. Select the **Nessus Agent** tile and click **Next**.
6. Enter configuration parameters in the Tenable user interface.
7. Click **Review + create**.

Deploy From Command-line

There is also a command-line interface available through PowerShell.

For example, you can type:

```
PS>
```

```
PS> $typeName="Linux" (or $typeName="Windows")
PS> $name = $publisherName + "." + $typeName
PS> $version="1.0"
PS> $Settings = @{"nessusManagerApp" = "IO"; "nessusAgentName" = "NA_name1";
"nessusAgentGroup" = "GROUP1"}
PS> $ProtectedSettings = @{"nessusLinkingKey" =
"abcd1234vxyz5678abcd1234vxyz5678abcd1234vxyz5678abcd1234vxyz5678"}
PS> Set-AzVMExtension -ResourceGroupName "EXAMPLE-resource-group" -Location "East US
2" -VMName "canary-example" -Name $name -Publisher $publisherName -ExtensionType
$typeName -TypeHandlerVersion $version -Settings $Settings -ProtectedSettings
$ProtectedSettings
```

Lines 1-4 identify the One-Click agent extension.

Lines 5-6 in the PowerShell example are equivalent to Step 6 in the UI procedure. This is where the user enters their configuration parameters for their Nessus Agent install.

Nessus Linking Key

The most important field is the Nessus Linking Key (**nessusLinkingKey**, required). It is always required. This document explains where to find it: Retrieve the Tenable Nessus Agent Linking Key (Tenable Nessus Agent 10.4). In the PowerShell interface, specify nessusLinkingKey under `-ProtectedSettings` so that it will be encrypted by Azure. All other fields are passed unencrypted through -Settings.

You can choose whether to link with Nessus Manager or Tenable.io. In the command-line interface, this is done by setting `nessusManagerApp` (**nessusManagerApp**, required) to `cloud`, or to `local`. Those are the only two choices.

If you choose Nessus Manager, you must provide the Nessus Manager host (**nessusManagerHost**) and port number (**nessusManagerPort**). The extension accepts an IP address or fully qualified domain name.

If you choose Tenable.io, then there is an optional field called **tenableIoNetwork**.

The Agent Name (**nessusAgentName**, optional) and Agent Group (**nessusAgentGroup**, optional) (actually "groups", a comma-delimited list of group names) are always optional.

Parameter names:

"nessusLinkingKey"

```
"nessusManagerApp"
"nessusManagerHost"
"nessusManagerPort"
"tenableIoNetwork"
"nessusAgentName"
"nessusAgentGroup"
```

Parameter descriptions:

"nessusLinkingKey" is called "--key" in this doc

"nessusManagerApp" is unique to our VM extension

"nessusManagerHost" equals "--host"

"nessusManagerPort" equals "--port"

"tenableIoNetwork" is "--network"

"nessusAgentName" is "--name"

"nessusAgentGroup" is "--groups"

For more definitions of these parameters, see Nessuscli Agent.

Support

If you need more help at any point in this article, you can contact the Azure experts on the MSDN Azure and Stack Overflow forums. Alternatively, you can file an Azure support incident. Go to the Azure support site and select Get support. For information about using Azure

Support, read the Microsoft Azure support FAQ. If you experience issues with the extension, contact Tenable support.

Last updated on 11/10/2025

Use Azure Policy to restrict extensions installation on Linux VMs

If you want to prevent the installation of certain extensions on your Linux VMs, you can create an Azure Policy definition using the Azure CLI to restrict extensions for VMs within a resource group. To learn the basics of Azure VM extensions for Linux, see Virtual machine extensions and features for Linux.

This tutorial uses the CLI within the Azure Cloud Shell, which is constantly updated to the latest version. If you want to run the Azure CLI locally, you need to install version 2.0.26 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Create a rules file

In order to restrict what extensions are available, you need to create a rule to identify the extension.

This example demonstrates how to deny the installation of disallowed VM extensions by defining a rules file in Azure Cloud Shell. However, if you're working in Azure CLI locally, you can create a local file and replace the path (~/clouddrive) with the path to the file on your local file system.

1. In a bash Cloud Shell create the file `~/clouddrive/azurepolicy.rules.json` using any text editor.
2. Copy and paste the following `.json` contents into the new file and save it.

JSON

```
{
    "if": {
        "allOf": [
            {
                "field": "type",
                "equals": "Microsoft.Compute/virtualMachines/extensions"
            },
            {
```

```
                "field": "Microsoft.Compute/virtualMachines/extensions/publisher",
                "equals": "Microsoft.OSTCExtensions"
            },
            {
                "field": "Microsoft.Compute/virtualMachines/extensions/type",
                "in": "[parameters('notAllowedExtensions')]"
            }
        ]
    },
    "then": {
        "effect": "deny"
    }
}
```

Create a parameters file

You also need a parameters file that creates a structure for you to use for passing in a list of the unauthorized extensions.

This example shows you how to create a parameter file for Linux VMs in Cloud Shell.

1. In the bash Cloud Shell opened before, create the file ~/clouddrive/azurepolicy.parameters.json using any text editor.

2. Copy and paste the following `.json` contents into the new file and save it.

JSON

```
{
    "notAllowedExtensions": {
        "type": "Array",
        "metadata": {
            "description": "The list of extensions that will be denied. Example: CustomScriptForLinux, VMAccessForLinux etc.",
            "displayName": "Denied extension"
        }
    }
}
```

Create the policy

A *policy definition* is an object used to store the configuration that you would like to use. The policy definition uses the rules and parameters files to define the policy. Create the policy definition using az policy definition create.

In this example, the rules and parameters are the files you created and stored as .json files in Cloud Shell or in your local file system.

Azure CLI

```
az policy definition create \
   --name 'not-allowed-vmextension-linux' \
   --display-name 'Block VM Extensions' \
   --description 'This policy governs which VM extensions that are blocked.' \
   --rules '~/clouddrive/azurepolicy.rules.json' \
   --params '~/clouddrive/azurepolicy.parameters.json' \
   --mode All
```

Assign the policy

This example assigns the policy to a resource group using az policy assignment create. Any VM created in the **myResourceGroup** resource group will be unable to install the Linux VM Access or the Custom Script Extensions for Linux.

> **Note**
>
> The resource group must exist before you can assign the policy.

Use az account list to find your subscription ID and replace the placeholder in the following example:

Azure CLI

```
az policy assignment create \
   --name 'not-allowed-vmextension-linux' \
   --scope /subscriptions/<subscription Id>/resourceGroups/myResourceGroup \
   --policy "not-allowed-vmextension-linux" \
   --params '{
        "notAllowedExtensions": {
            "value": [
                "VMAccessForLinux",
                "CustomScriptForLinux"
            ]
        }
    }'
```

Test the policy

Test the policy by creating a new VM and adding a new user.

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image myImage \
    --generate-ssh-keys
```

> **Note**
>
> Replace `myResourceGroup`, `myVM` and `myImage` values accordingly.

Try to create a new user named **myNewUser** using the VM Access extension.

Azure CLI

```
az vm user update \
  --resource-group myResourceGroup \
  --name myVM \
  --username myNewUser \
  --password 'mynewuserpwd123!'
```

Remove the assignment

Azure CLI

```
az policy assignment delete --name 'not-allowed-vmextension-linux' --resource-group
myResourceGroup
```

Remove the policy

Azure CLI

```
az policy definition delete --name 'not-allowed-vmextension-linux'
```

Next steps

For more information, see Azure Policy.

Last updated on 11/10/2025

Use Azure Policy to restrict extensions installation on Windows VMs

If you want to prevent the use or installation of certain extensions on your Windows VMs, you can create an Azure Policy definition using PowerShell to restrict extensions for VMs within a resource group.

This tutorial uses Azure PowerShell within the Cloud Shell, which is constantly updated to the latest version.

> **Note**
>
> **New! Try VM assist for Faster Diagnostics`** We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Create a rules file

In order to restrict what extensions can be installed, you need to have a rule to provide the logic to identify the extension.

This example shows you how to deny extensions published by 'Microsoft. Compute' by creating a rules file in Azure Cloud Shell, but if you're working in PowerShell locally, you can also create a local file and replace the path ($home/clouddrive) with the path to the local file on your machine.

1. In a Cloud Shell, create the file `$home/clouddrive/rules.json` using any text editor.

2. Copy and paste the following .json contents into the file and save it:

JSON

```
{
    "if": {
        "allOf": [
            {
                "field": "type",
                "equals": "Microsoft.Compute/virtualMachines/extensions"
            },
            {
                "field": "Microsoft.Compute/virtualMachines/extensions/publisher",
                "equals": "Microsoft.Compute"
            },
            {
```

```
                "field": "Microsoft.Compute/virtualMachines/extensions/type",
                "in": "[parameters('notAllowedExtensions')]"
            }
        ]
    },
    "then": {
        "effect": "deny"
    }
}
```

Create a parameters file

You also need a parameters file that creates a structure for you to use for passing in a list of the extensions to block.

This example shows you how to create a parameters file for VMs in Cloud Shell, but if you're working in PowerShell locally, you can also create a local file and replace the path ($home/clouddrive) with the path to the local file on your machine.

1. In Cloud Shell, create the file `$home/clouddrive/parameters.json` using any text editor.

2. Copy and paste the following .json contents into the file and save it:

JSON

```
{
    "notAllowedExtensions": {
        "type": "Array",
        "metadata": {
            "description": "The list of extensions that will be denied.",
            "displayName": "Denied extension"
        }
    }
}
```

Create the policy

A policy definition is an object used to store the configuration that you would like to use. The policy definition uses the rules and parameters files to define the policy. Create a policy definition using the New-AzPolicyDefinition cmdlet.

The policy rules and parameters are the files you created and stored as .json files in your cloud shell. Replace the example `-Policy` and `-Parameter` file paths as needed.

Azure PowerShell

```
$definition = New-AzPolicyDefinition `
   -Name "not-allowed-vmextension-windows" `
   -DisplayName "Not allowed VM Extensions" `
   -description "This policy governs which VM extensions that are explicitly
denied."   `
   -Policy 'C:\Users\ContainerAdministrator\clouddrive\rules.json' `
   -Parameter 'C:\Users\ContainerAdministrator\clouddrive\parameters.json'
```

Assign the policy

This example assigns the policy to a resource group using New-AzPolicyAssignment. Any VM created in the **myResourceGroup** resource group won't be able to install the VM Access Agent or Custom Script extensions.

Use the Get-AzSubscription | Format-Table cmdlet to get your subscription ID to use in place of the one in the example.

Azure PowerShell

```
$scope = "/subscriptions/<subscription id>/resourceGroups/myResourceGroup"
$assignment = New-AzPolicyAssignment `
   -Name "not-allowed-vmextension-windows" `
   -Scope $scope `
   -PolicyDefinition $definition `
   -PolicyParameter '{
    "notAllowedExtensions": {
        "value": [
            "VMAccessAgent",
            "CustomScriptExtension"
        ]
    }
}'
$assignment
```

Test the policy

To test the policy, try to use the VM Access extension. The following should fail with the message "Set-AzVMAccessExtension: Resource 'myVMAccess' was disallowed by policy."

Azure PowerShell

```
Set-AzVMAccessExtension `
   -ResourceGroupName "myResourceGroup" `
   -VMName "myVM" `
   -Name "myVMAccess" `
   -Location EastUS
```

In the portal, the password change should fail with the "The template deployment failed because of policy violation." message.

Remove the assignment

Azure PowerShell

```
Remove-AzPolicyAssignment -Name not-allowed-vmextension-windows -Scope $scope
```

Remove the policy

Azure PowerShell

```
Remove-AzPolicyDefinition -Name not-allowed-vmextension-windows
```

Next steps

For more information, see Azure Policy.

Last updated on 11/10/2025

How to update the Azure Linux Agent on a VM

> **Caution**
>
> This article references CentOS, a Linux distribution that is end of support (EOS) status. Consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

To update your Azure Linux Agent on a Linux VM in Azure, you must already have:

- A running Linux VM in Azure.
- A connection to that Linux VM using SSH.

You should always check for a package in the Linux distro repository first. It's possible the package available might not be the latest version, however, enabling autoupdate ensures the Linux Agent always gets the latest update. Should you have issues installing from the package managers, you should seek support from the distro vendor.

> **Note**
>
> For more information, see Endorsed Linux distributions on Azure

> **Note**
>
> Try VM assist for faster diagnostics. We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Verify the Minimum version support for virtual machine agents in Azure before proceeding.

Ubuntu

1. Check your current package version

```bash
sudo apt list --installed | grep walinuxagent
```

2. Update package cache

Bash

```
sudo apt-get -qq update
```

3. Install the latest package version

Bash

```
sudo apt-get install walinuxagent
```

4. Ensure auto update is enabled.

- First, check to see if auto update is enabled:

Bash

```
sudo cat /etc/waagent.conf | grep -i autoupdate
```

- Find 'AutoUpdate.Enabled" if you see this output when auto update is enabled:

config

```
AutoUpdate.Enabled=y
```

- To enable it, run:

Bash

```
sudo sed -i 's/# AutoUpdate.Enabled=y/AutoUpdate.Enabled=y/g' /etc/waagent.conf
```

5. Restart the waagent service

Bash

```
sudo systemctl restart walinuxagent
```

6. Validate waagent service is up and running

Bash

```
sudo systemctl status walinuxagent
```

Typically all you need, but if for some reason you need to install it from https://github.com directly, use the following steps.

Update the Linux Agent when no agent package exists for distribution

For more information regarding updating the Azure Linux Agent when no package exists, see Azure Linux Agent README.

Last updated on 11/10/2025

Export resource groups that contain VM extensions

08/18/2025

You can export Azure resource groups into a new Azure Resource Manager template that you can then redeploy. The export process interprets existing resources and creates a Resource Manager template that, when deployed, results in a similar resource group. When you're using the export option against a resource group that contains virtual machine (VM) extensions, you need to consider items such as extension compatibility and protected settings.

This article details how the resource group export process works for virtual machine extensions. It includes a list of supported extensions and details on how to handle secured data.

Supported VM extensions

Many VM extensions are available. You can't export all extensions into a Resource Manager template by using the automation script feature. If a virtual machine extension is not supported, you need to manually place it back into the exported template.

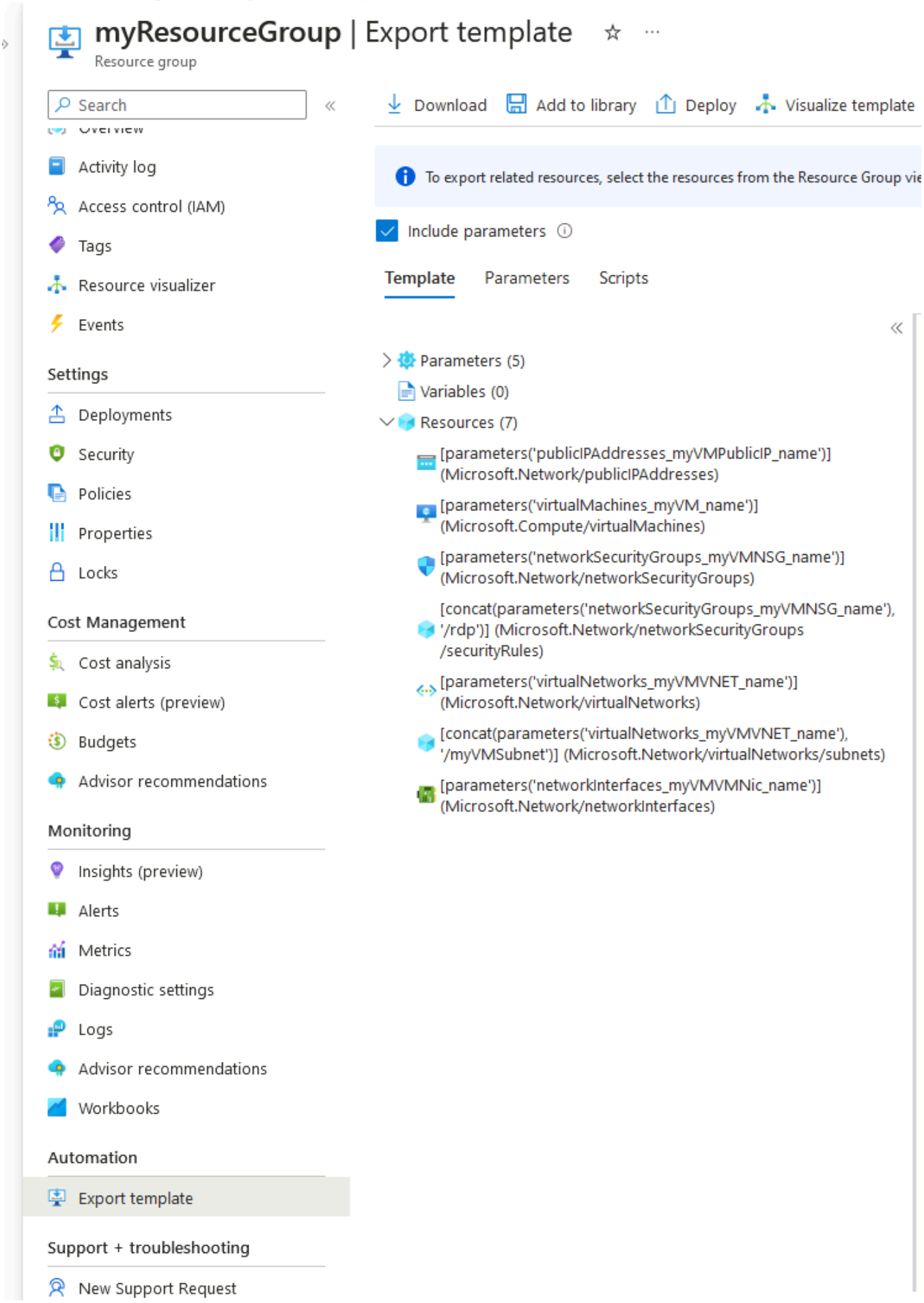
You can export the following extensions by using the automation script feature:

> Acronis Backup, Acronis Backup Linux, BGInfo, BMC Control-M Agent Linux, BMC Control-M Agent Windows, Chef Client, Custom Script, Custom Script Extension, Custom Script for Linux, Datadog Linux Agent, Datadog Windows Agent, Docker Extension, Dynatrace Linux, Dynatrace Windows, HPE Security Application Defender for Cloud, IaaS Antimalware, IaaS Diagnostics, Linux Chef Client, Linux Diagnostic, OS Patching for Linux, Puppet Agent, Site24x7 APM Insight, Site24x7 Linux Server, Site24x7 Windows Server, Trend Micro DSA, Trend Micro DSA Linux, VM Access For Linux, VM Access For Linux, VM Snapshot, VM Snapshot Linux

Export the resource group

To export a resource group into a reusable template, complete the following steps:

1. Sign in to the Azure portal.
2. On the **Hub** menu, select **Resource Groups**.
3. Select the target resource group from the list.
4. On the **Resource group** pane, select **Export template** under the **Automation** section.



The Azure Resource Manager automation script produces a Resource Manager template, a parameters file, and several sample deployment scripts, such as PowerShell and Azure CLI

scripts. At this point, you can download the exported template by using the download button, add the template to the template library, or redeploy the template by using the **Deploy** button.

Configure protected settings

Many Azure VM extensions include a protected settings configuration that encrypts sensitive data, such as credentials and configuration strings. Protected settings are not exported with the automation script. If necessary, reinsert protected settings into the exported template.

Step 1: Remove the template parameter

When you export a resource group, a single template parameter is created to provide a value to the exported protected settings. You can remove this parameter.

To remove the parameter, look through the list of parameters and delete the one that looks similar to this JSON example:

JSON

```
"extensions_extensionname_protectedSettings": {
    "defaultValue": null,
    "type": "SecureObject"
}
```

Step 2: Get properties for protected settings

Because each protected setting has a set of required properties, you need to gather a list of these properties. You can find each parameter of the protected settings configuration in the Azure Resource Manager schema on GitHub. This schema includes only the parameter sets for the extensions that are listed in the overview section of this article.

From within the schema repository, search for the desired extension. After you find the extension's `protectedSettings` object, take note of each parameter. In the following example of the `IaasDiagnostic` extension, the required parameters are `storageAccountName`, `storageAccountKey`, and `storageAccountEndPoint`:

JSON

```
"protectedSettings": {
    "type": "object",
    "properties": {
        "storageAccountName": {
            "type": "string"
```

```
        },
        "storageAccountKey": {
            "type": "string"
        },
        "storageAccountEndPoint": {
            "type": "string"
        }
    },
    "required": [
        "storageAccountName",
        "storageAccountKey",
        "storageAccountEndPoint"
    ]
}
```

Step 3: Re-create the protected configuration

On the exported template, search for `protectedSettings`. Replace the exported protected setting object with a new one that includes the required extension parameters and a value for each one.

In the example of the `IaasDiagnostic` extension, the new protected setting configuration would look like the following example:

JSON

```
"protectedSettings": {
    "storageAccountName": "[parameters('storageAccountName')]",
    "storageAccountKey": "[parameters('storageAccountKey')]",
    "storageAccountEndPoint": "https://core.windows.net"
}
```

The final extension resource looks similar to the following JSON example:

JSON

```
{
    "name": "Microsoft.Insights.VMDiagnosticsSettings",
    "type": "extensions",
    "location": "[resourceGroup().location]",
    "apiVersion": "[variables('apiVersion')]",
    "dependsOn": [
        "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
    ],
    "tags": {
        "displayName": "AzureDiagnostics"
    },
    "properties": {
        "publisher": "Microsoft.Azure.Diagnostics",
```

```
        "type": "IaaSDiagnostics",
        "typeHandlerVersion": "1.5",
        "autoUpgradeMinorVersion": true,
        "settings": {
            "xmlCfg": "[base64(concat(variables('wadcfgxstart'),
variables('wadmetricsresourceid'), variables('vmName'),
variables('wadcfgxend')))]",
            "storageAccount": "
[parameters('existingdiagnosticsStorageAccountName')]"
        },
        "protectedSettings": {
            "storageAccountName": "[parameters('storageAccountName')]",
            "storageAccountKey": "[parameters('storageAccountKey')]",
            "storageAccountEndPoint": "https://core.windows.net"
        }
    }
}
```

If you're using template parameters to provide property values, you need to create them. When you're creating template parameters for protected setting values, use the `SecureString` parameter type to help secure sensitive values. For more information on using parameters, see Authoring Azure Resource Manager templates.

In the example of the `IaasDiagnostic` extension, the following parameters would be created in the parameters section of the Resource Manager template:

JSON

```
"storageAccountName": {
    "defaultValue": null,
    "type": "SecureString"
},
"storageAccountKey": {
    "defaultValue": null,
    "type": "SecureString"
}
```

At this point, you can deploy the template by using any template deployment method.

Troubleshooting Azure Windows VM extension failures

08/27/2025

Overview of Azure Resource Manager templates

Azure Resource Manager templates allows you to declaratively specify the Azure IaaS infrastructure in JSON language by defining the dependencies between resources.

See Authoring extension templates to learn more about authoring templates for using extensions.

In this article we'll learn about troubleshooting some of the common VM extension failures.

Viewing extension status

Azure Resource Manager templates can be executed from Azure PowerShell. Once the template is executed, the extension status can be viewed from Azure Resource Explorer or the command-line tools.

Here's an example:

Azure PowerShell:

Azure PowerShell

```
Get-AzVM -ResourceGroupName $RGName -Name $vmName -Status
```

Here's the sample output:

Output

```
Extensions:  {
  "ExtensionType": "Microsoft.Compute.CustomScriptExtension",
  "Name": "myCustomScriptExtension",
  "SubStatuses": [
    {
      "Code": "ComponentStatus/StdOut/succeeded",
      "DisplayStatus": "Provisioning succeeded",
      "Level": "Info",
      "Message": "    Directory: C:\\temp\\n\\n\\nMode
LastWriteTime     Length Name
          \\n----                -------------     ------ ----
\\n-a---          9/1/2015   2:03 AM         11
```

```
          test.txt                          \\n\\n",
                  "Time": null
      },
    {
      "Code": "ComponentStatus/StdErr/succeeded",
      "DisplayStatus": "Provisioning succeeded",
      "Level": "Info",
      "Message": "",
      "Time": null
    }
  ]
}
```

Troubleshooting extension failures

Verify that the VM Agent is running and Ready

The VM Agent is required to manage, install and execute extensions. If the VM Agent isn't running or is failing to report a Ready status to the Azure platform, then the extension isn't working correctly.

The following pages to troubleshoot the VM Agent:

- Troubleshooting Windows Azure Guest Agent for a Windows VM
- Troubleshoot the Azure Linux Agent for a Linux VM

Check for your specific extension troubleshooting guide

Some extensions have a specific page describing how to troubleshoot them. You can find the list of these extensions and pages on Troubleshoot extensions .

View the extension's status

As explained previously, the extension's status can be found by running the PowerShell cmdlet:

Azure PowerShell

```
Get-AzVM -ResourceGroupName $RGName -Name $vmName -Status
```

Or the CLI command:x

Azure CLI

```
az vm extension show -g <RG Name> --vm-name <VM Name>  --name <Extension Name>
```

Or in the Azure portal, by browsing to the VM Blade / Settings / Extensions. You can then click on the extension and check its status and message.

Rerun the extension on the VM

If you're running scripts on the VM using Custom Script Extension, you could sometimes run into an error where VM was created successfully but the script fails. Under these conditions, the recommended way to recover from this error is to remove the extension and rerun the template again.

> **Note**
>
> In future, this functionality would be enhanced to remove the need for uninstalling the extension.

Remove the extension from Azure PowerShell

Azure PowerShell

```
Remove-AzVMExtension -ResourceGroupName $RGName -VMName $vmName -Name
"myCustomScriptExtension"
```

Once the extension is removed, the template can be re-executed to run the scripts on the VM.

Trigger a new GoalState to the VM

You might see that an extension isn't running because the 'Windows Azure CRP Certificate Generator' is missing. This certificate helps protect the extension's settings during transport. That certificate is automatically regenerated by restarting the Windows Guest Agent from inside the Virtual Machine:

- Open the Task Manager
- Go to the Details tab
- Locate the WindowsAzureGuestAgent.exe process
- Right-click, and select "End Task". The process is automatically restarted

You can also trigger a new GoalState to the VM, by executing a "VM Reapply". VM Reapply is an API introduced in 2020 to reapply a VM's state. We recommend doing this at a time when

you can tolerate a short VM downtime. Reapply usually doesn't reboot the VM, but in rare cases, it might trigger a pending update that does require a restart.

Azure portal:

In the portal, select the VM and in the left pane under the **Support + troubleshooting**, select **Redeploy + reapply**, then select **Reapply**.

Azure PowerShell *(replace the RG Name and VM Name with your values)*:

Azure PowerShell

```
Set-AzVM -ResourceGroupName <RG Name> -Name <VM Name> -Reapply
```

Azure CLI *(replace the RG Name and VM Name with your values)*:

Azure CLI

```
az vm reapply -g <RG Name> -n <VM Name>
```

If a "VM Reapply" didn't work, you can add a new empty Data Disk to the VM from the Azure Management Portal, and then remove it later once the certificate is added back.

Look at the extension logs inside the VM

If the previous steps didn't work and if your extension is still in a failed state, the next step is to look at its logs inside the Virtual Machine.

On a **Windows** VM, the extension logs by default reside in

```
C:\WindowsAzure\Logs\Plugins
```

And the Extension settings and status files by default are in

```
C:\Packages\Plugins
```

On a **Linux** VM, the extension logs be default reside in

```
/var/log/azure/
```

And the Extension settings and status files by details is in

```
/var/lib/waagent/
```

Each extension is different, but they usually follow similar principles:

Extension packages and binaries are downloaded on the VM (example: "*/var/lib/waagent/custom-script/download/1*" for Linux or "*C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.10.12\Downloads\0*" for Windows).

Their configuration and settings are passed from Azure Platform to the extension handler through the VM Agent (example: "*/var/lib/waagent/Microsoft.Azure.Extensions.CustomScript-2.1.3/config*" for Linux or "*C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.10.12\RuntimeSettings*" for Windows)

Extension handlers inside the VM are writing to a status file (example: "*/var/lib/waagent/Microsoft.Azure.Extensions.CustomScript-2.1.3/status/1.status*" for Linux or "*C:\Packages\Plugins\Microsoft.Compute.CustomScriptExtension\1.10.12\Status*" for Windows) which is reported to the Azure Platform. That status is the one reported through PowerShell, CLI or in the VM's extension blade in the Azure portal.

They also write detailed logs of their execution (example: "*/var/log/azure/custom-script/handler.log*" for Linux or "*C:\WindowsAzure\Logs\Plugins\Microsoft.Compute.CustomScriptExtension\1.10.12\CustomScriptHandler.log*" for Windows).

If the VM is recreated from an existing VM

It could happen that you're creating an Azure VM based on a specialized Disk coming from another Azure VM. In that case, it's possible that the old VM contained extensions, and binaries, logs, and status files left over. The new VM model isn't aware of the previous VM's extensions states, and it might report an incorrect status for these extensions. We strongly recommend you to remove the extensions from the old VM before creating the new one, and then reinstall these extensions once the new VM is created. The same can happen when you create a generalized image from an existing Azure VM. We invite you to remove extensions to avoid inconsistent state from the extensions.

Known issues

PowerShell isn't recognized as an internal or external command

You notice the following error entries in the RunCommand extension's output:

```
Log

RunCommandExtension failed with "'powershell' isn't recognized as an internal or
external command,"
```

Analysis

Extensions run under Local System account, so it's possible that powershell.exe works fine when you RDP into the VM, but fails when run with RunCommand.

Solution

- Check that PowerShell is properly listed in the PATH environment variable:
  - Open Control Panel
  - System and Security
  - System
  - Advanced tab -> Environmental Variables
- Under 'System variables' click edit and ensure that PowerShell is in the PATH environment variable (usually: "C:\Windows\System32\WindowsPowerShell\v1.0")
- Reboot the VM or restart the WindowsAzureGuestAgent service then try the Run Command again.

Command isn't recognized as an internal or external command

You see the following in the C:\WindowsAzure\Logs\Plugins<ExtensionName><Version>\CommandExecution.log file:

```
Log

Execution Error: '<command>' isn't recognized as an internal or external command,
operable program or batch file.
```

Analysis

Extensions run under Local System account, so it's possible that powershell.exe works fine when you RDP into the VM, but fails when run with RunCommand.

Solution

- Open a Command Prompt in the VM and execute a command to reproduce the error. The VM Agent uses the Administrator cmd.exe and you may have some preconfigured command to execute every time cmd is started.
- It's also likely that your PATH variable is misconfigured, but that depends on the command that is having the problem.

VMAccessAgent is failing with Cannot update Remote Desktop Connection settings for Administrator account. Error: System.Runtime.InteropServices.COMException (0x800706D9): There are no more endpoints available from the endpoint mapper.

You see the following in the extension's status:

Log

```
Type Microsoft.Compute.VMAccessAgent
Version 2.4.8
Status Provisioning failed
Status level Error
Status message Cannot update Remote Desktop Connection settings for Administrator
account. Error: System.Runtime.InteropServices.COMException (0x800706D9): There
are no more endpoints available from the endpoint mapper. (Exception from HRESULT:
0x800706D9) at NetFwTypeLib.INetFwRules.GetEnumerator() at
Microsoft.WindowsAzure.GuestAgent.Plugins.JsonExtensions.VMAccess.RemoteDesktopMan
ager.EnableRemoteDesktopFirewallRules()
at
Microsoft.WindowsAzure.GuestAgent.Plugins.JsonExtensions.VMAccess.RemoteDesktopMan
ager.EnableRemoteDesktop() at
```

Analysis

This error can happen when the Windows Firewall service isn't running.

Solution

Check if the Windows Firewall service is enabled and running. If it's not, enable and start it - then try again to run the VMAccessAgent.

The remote certificate is invalid according to the validation procedure.

You see the following in the WaAppAgent.log

```
Log

System.Net.WebException: The underlying connection was closed: Could not establish
trust relationship for the SSL/TLS secure channel. ---> System.Security.
Authentication.AuthenticationException: The remote certificate is invalid
according to the validation procedure.
```

Analysis

Your VM is probably missing the Baltimore CyberTrust Root certificate in "Trusted Root Certification Authorities".

Solution

Open the certificates console with certmgr.msc, and check if the certificate is there.

A third-party SSL inspection tool, like ZScaler, might break the certificate chain. You should configure such tools to bypass SSL inspection.

Issues using VM extensions in Python 3-enabled Linux Azure Virtual Machines systems

> ⓘ Note
>
> Microsoft encourages users to adopt **Python 3.x** in their systems unless your workload requires **Python 2.x** support. Examples of this requirement might include legacy administration scripts, or extensions such as **Azure Disk Encryption** and **Azure Monitor**.
>
> Before installing **Python 2.x** in production, consider the question of long-term support of Python 2.x, particularly their ability to receive security updates. As products, including some of the extension mentioned, update with **Python 3.8** support, you should discontinue use of Python 2.x.

Some Linux distributions have transitioned to Python 3.8 and removed the legacy `/usr/bin/python` entrypoint for Python altogether. This transition impacts the out-of-the-box, automated deployment of certain virtual machine (VM) extensions with these two conditions:

- Extensions that are still transitioning to Python 3.x support
- Extensions that use the legacy `/usr/bin/python` entrypoint

Linux distribution users who have transitioned to **Python 3.x** must ensure the legacy `/usr/bin/python` entrypoint exists before attempting to deploy those extensions to their VMs. Otherwise, the extension deployment might fail.

- Endorsed Linux distributions that are affected include **Ubuntu Server 20.04 LTS** and **Ubuntu Pro 20.04 LTS**.
- Affected VM Extensions include **Azure Disk Encryption**, **Log Analytics**, **VM Access** (used for Password Reset), and **Guest Diagnostics** (used for additional performance counters).

In-place upgrades, such as upgrading from **Ubuntu 18.04 LTS** to **Ubuntu 20.04 LTS**, should retain the `/usr/bin/python` symlink, and remain unaffected.

> ⓘ Note
>
> **New! Try VM assist for Faster Diagnostics**` We recommend that you run VM assist for Windows or VM assist for Linux. These script-based diagnostic tools help you to identify common issues that affect the Azure VM Guest Agent and overall VM health.

Resolution

Consider these general recommendations before deploying extensions in the known-affected scenarios described previously in the Summary:

1. Before deploying the extension, reinstate the `/usr/bin/python` symlink by using the Linux distribution vendor-provided method.

   - For example, for **Python 2.7**, use: `sudo apt update && sudo apt install python-is-python2`

2. This recommendation is for Azure customers and isn't supported in Azure Stack:

   - If you've already deployed an instance that exhibits this problem, use the Run command functionality in the VM blade to run the commands mentioned above. The Run command extension itself isn't affected by the transition to Python 3.8.

3. If you're deploying a new instance, and need to set an extension at provisioning time, use **cloud-init** user data to install the packages mentioned above.

   For example, for Python 2.7:

   Python

   ```
   # create cloud-init config
   cat > cloudinitConfig.json <<EOF
   #cloud-config
   package_update: true

   runcmd:
   - sudo apt update
   - sudo apt install python-is-python2
   EOF

   # create VM
   az vm create \
       --resource-group <resourceGroupName> \
       --name <vmName> \
       --image <Ubuntu 20.04 Image URN> \
       --admin-username azadmin \
       --ssh-key-value "<sshPubKey>" \
       --custom-data ./cloudinitConfig.json
   ```

4. If your organization's policy administrators determine that extensions shouldn't be deployed in VMs, you can disable extension support at provisioning time:

   - REST API

To disable and enable extensions when you can deploy a VM with this property:

Python

```
"osProfile": {
  "allowExtensionOperations": false
},
```

Next steps

Please refer to Other base system changes since 18.04 LTS - Python 3 by default for additional information.

Last updated on 11/10/2025

What is Nested Virtualization?

07/25/2025

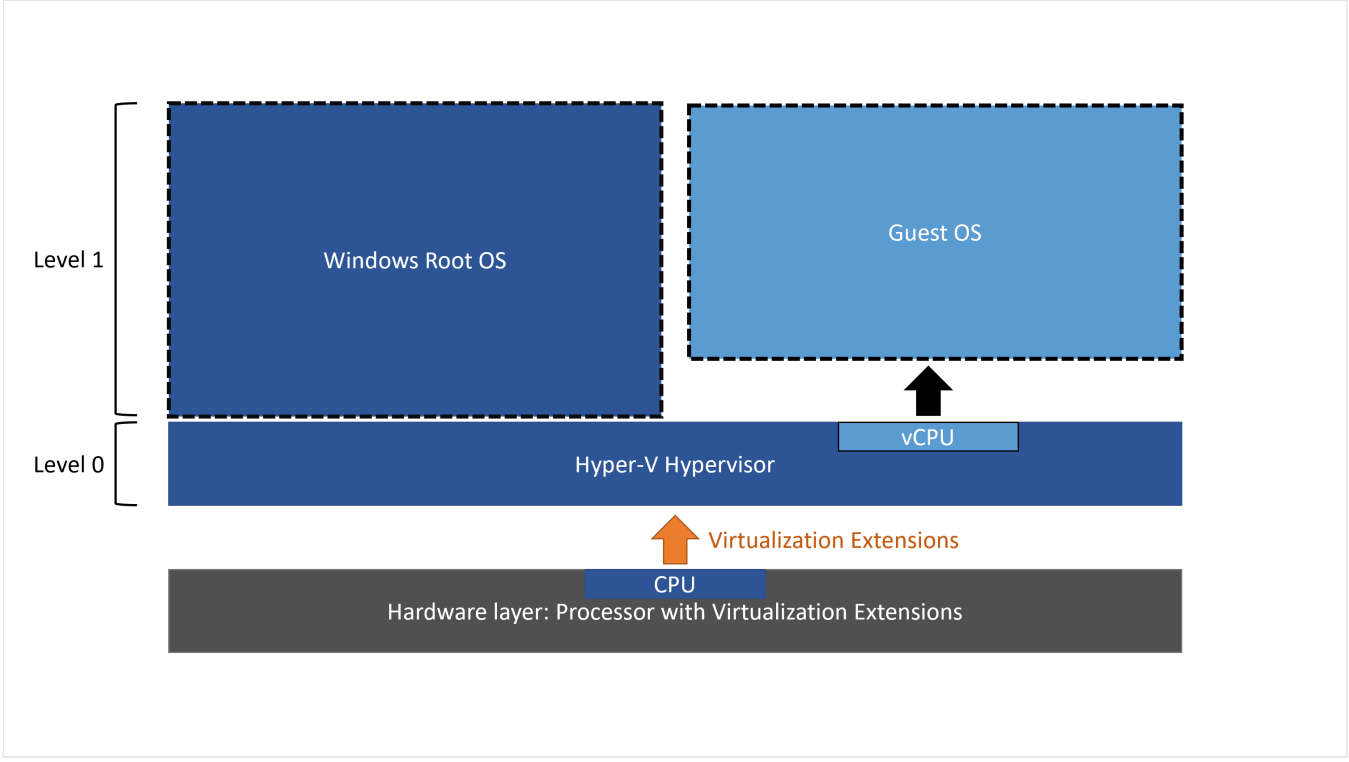
Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10, ✅ Azure Local 2311.2 and later

Nested virtualization is a feature that lets you run Hyper-V inside a Hyper-V virtual machine (VM). This article explains how nested virtualization works, its benefits, and when to use it in your environment. For example, nested virtualization can be useful for:

- Running applications or emulators in a nested VM
- Testing software releases on VMs
- Reducing deployment times for training environments
- Using Hyper-V isolation for containers

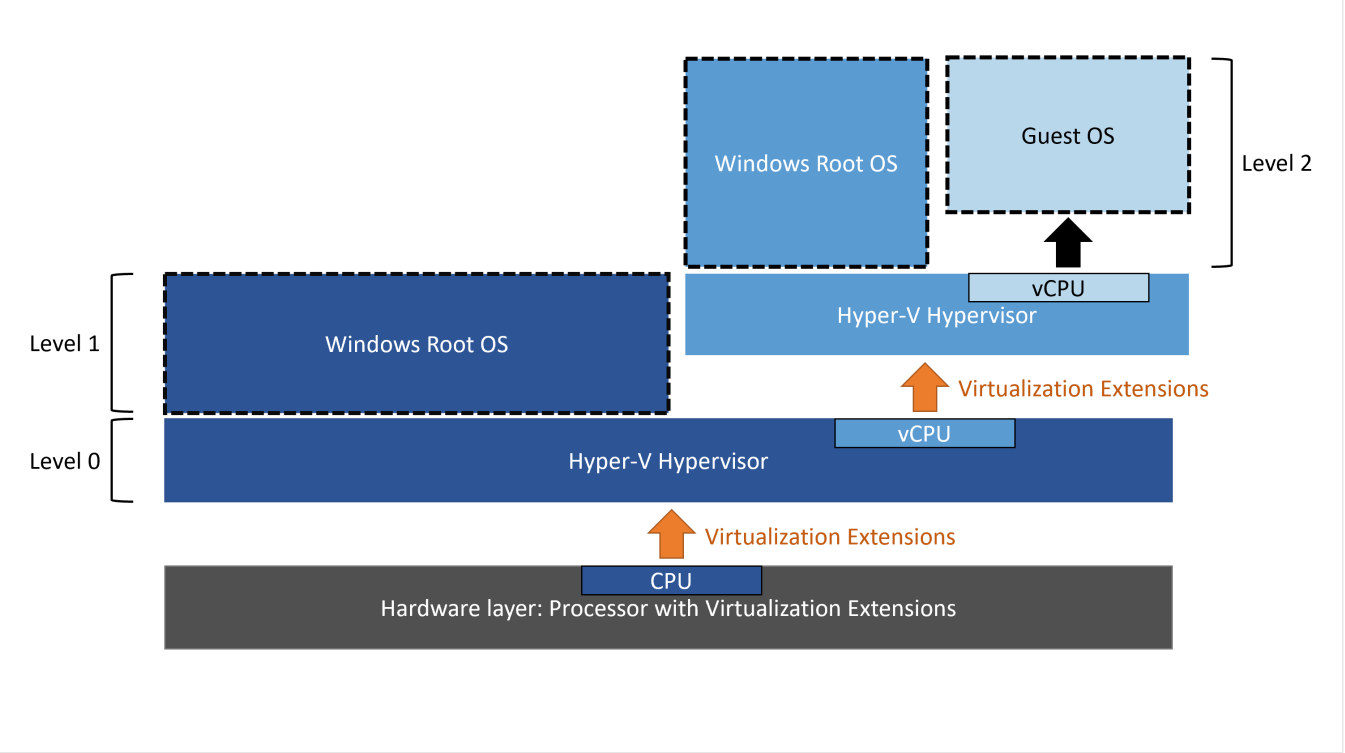
Modern processors include hardware features that make virtualization faster and more secure. Hyper-V relies on these processor extensions to run virtual machines, for example, Intel VT-x and AMD-V. Nested virtualization makes this hardware support available to guest virtual machines.

The following diagram shows Hyper-V without nesting. The Hyper-V hypervisor takes full control of the hardware virtualization capabilities (orange arrow), and doesn't expose them to the guest operating system.



In contrast, the following diagram shows Hyper-V with Nested Virtualization enabled. In this case, Hyper-V exposes the hardware virtualization extensions to its virtual machines. With

nesting enabled, a guest virtual machine can install its own hypervisor and run its own guest VMs.



Dynamic Memory and Runtime Memory Resize

When Hyper-V is running inside a virtual machine, the virtual machine must be turned off to adjust its memory. Meaning that even if dynamic memory is enabled, the amount of memory doesn't fluctuate. Simply enabling nested virtualization has no effect on dynamic memory or runtime memory resize.

For virtual machines without dynamic memory enabled, attempting to adjust the amount of memory while running fails. The incompatibility only occurs while Hyper-V is running in the VM.

Non-Microsoft virtualization apps

Virtualization applications other than Hyper-V aren't supported in Hyper-V virtual machines, and are likely to fail. Virtualization applications include any software that requires hardware virtualization extensions.

Supported scenarios

Using a nested Hyper-V VM in production is supported for both Azure and on-premises in the following scenarios. We also recommend you make sure that your services and applications are

also supported.

Nested Virtualization isn't suitable for Windows Server Failover Clustering, and performance sensitive applications. We recommended you fully evaluate the services and applications.

Hyper-V VMs on Hyper-V VMs

Running Hyper-V VMs nested on Hyper-V VMs is great for test labs and evaluation environments. Especially where configurations can be easily modified and saved states can be used to revert to specific configurations. Test labs don't typically require the same service level agreement (SLA) as production environments.

Production environments running Hyper-V VMs running on Hyper-V VMs are supported. Ensure that your services and applications are also supported. If you use a nested Hyper-V VM in production, make sure you fully evaluate whether your services or applications provide the expected behavior.

To learn more about setting up Nested Virtualization on Azure, see our Tech Community blog How to Setup Nested Virtualization for Azure VM/VHD.

Non-Microsoft virtualization on Hyper-V virtualization

While it might be possible for non-Microsoft virtualization to run on Hyper-V, Microsoft doesn't test this scenario. Non-Microsoft virtualization on Hyper-V virtualization isn't supported. Ensure your hypervisor vendor supports this scenario.

Hyper-V virtualization on non-Microsoft virtualization

While it might be possible for Hyper-V virtualization to run on non-Microsoft virtualization, Microsoft doesn't test this scenario. Hyper-V virtualization on non-Microsoft virtualization isn't supported. Ensure your hypervisor vendor supports this scenario.

Azure Local nested on Hyper-V VMs

Azure Local is designed and tested to run on validated physical hardware. Azure Local can run nested in a virtual machine for evaluation, but production environments in a nested configuration aren't supported.

To learn more about Azure Local nested on Hyper-V VMs, see Nested virtualization in Azure Local.

Hyper-V isolated containers running nested on Hyper-V

Microsoft offers Hyper-V isolation for containers. This isolation mode offers enhanced security and broader compatibility between host and container versions. With Hyper-V isolation, multiple container instances run concurrently on a host. Each container runs inside of a highly optimized virtual machine and effectively gets its own kernel. Since a Hyper-V isolated container offers isolation through a hypervisor layer between itself and the container host, when the container host is a Hyper-V based virtual machine, there's performance overhead. The associated performance overhead occurs in terms of container start-up time, storage, network, and CPU operations.

When a Hyper-V isolated container is run in a Hyper-V VM, it's running nested. Using a Hyper-V VM opens many useful scenarios but also increases latency, as there are two levels of hypervisors running above the physical host.

Running Hyper-V isolated containers nested on Hyper-V is supported. One level of nested virtualization is supported in production, which allows for isolated container deployments.

To learn more about Nested Hyper-V Containers, see Performance tuning Windows Server Containers.

Run WSL2 in a Hyper-V VM running nested on Hyper-V

Windows Subsystem for Linux (WSL) is a feature of the Windows operating system that enables you to run a Linux file system, along with Linux command-line tools and GUI apps, directly on Windows.

Running WSL2 in a Hyper-V VM running nested on Hyper-V is supported.

To learn more about how to enabled WSL 2 to run in a VM, see Frequently Asked Questions about Windows Subsystem for Linux.

Related content

- Run Hyper-V in a Virtual Machine with Nested Virtualization

Estimate the cost of creating a Virtual Machine in the Azure portal (Preview)

Article • 08/22/2024

> **Important**
>
> The estimated cost card feature for virtual machines in the Azure portal is currently in PREVIEW. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

Azure virtual machines (VM) can be created through the Azure portal. This method provides a browser-based user interface to create VMs and their associated resources. As you create VMs in the Azure portal and configure them through the different tabs during the creation process, the cost card reflects the pricing estimations based on your selections.

The cost displayed in this card doesn't reflect usage and other costs incurred on the VMs beyond the creation of the resources. The final price of your VMs may factor in other respective networking, storage, and extra costs that pertain to the usage and application of the VMs.

> **Note**
>
> Costs reflected in the card are estimates only. Pricing may vary depending on your Microsoft agreement, date of purchase, subscription type, usage costs, licensing, and currency exchange rates. Total costs may include other resource costs, licensing, and subscription implications. The costs indicated on the card may vary across regions, subscription types, sizes, and images.

Prerequisites

Currently, the estimated cost card feature for VMs in the Azure portal is limited to Pay-as-you-go customers. This feature isn't available or visible for other subscription types and other discount plans, such as the Savings Plan and Reserved Instances.

If your subscription has policy restrictions for specific users, those users aren't able to view the cost card unless those restrictions are lifted. If your subscription has policy

restrictions to not show costs, cost card will not be shown unless those restrictions are lifted.

Cost card

The following sections reflect the various tabs during the VM creation process in the Azure portal. The listed resources and features are included in the estimated cost card. For further pricing information, refer to their respective pricing pages linked in the following tables.

Costs associated with selecting existing resources during the VM creation process are only reflected in the cost card if those resources may incur extra charges.

Basics

Expand table

Resource or feature	Notes
Image and Size (Windows)	Estimates of VM size and Windows based image are displayed.
Image and Size (Linux)	Estimates of VM size and Linux based image are displayed.
Spot	If selected, the discounted Spot price is shown.
Azure Hybrid Benefit (AHB) licensing	If selected, the AHB discounts are applied and shown.

Disks

Expand table

Resource or feature	Notes
OS Disk	Estimate for using the selected disk to host the selected operating system is displayed.
Ultra disk Reservation charge	A reservation charge is only imposed if Ultra Disk compatibility is enabled on the VM without attaching an Ultra Disk.
Data Disk	Estimate for using the selected disk, to store and access data, is displayed. A disk share estimate is displayed if enabled and other charges apply. Transaction and bursting costs aren't included in the estimate.

Networking

Expand table

Resource or feature	Notes
Public IP	Estimated cost of running a public IP Address. A Standard Public IP is selected when a new IP is created as part of the default selection.
VM Data transfer	Estimated cost is based on inter-region data transfer routed via Microsoft Premium Global Network.
Azure Load Balancer	A new Standard Azure Load Balancer estimate includes the cost of one rule and estimated data processed. For an existing Azure load balancer, the estimate includes only the estimated data transfer and Azure assumes rules already exist.
Azure Application Gateway	A new Application Gateway v2 estimate includes Fixed, Capacity units and estimate outbound data transfer costs. For this estimate, one capacity unit consumption is assumed. For an existing Azure Application gateway, only the outbound data transfer estimate is displayed. Azure assumes that other fixed costs are already paid for.

Management

Expand table

Resource or feature	Notes
Azure Backup	Monthly average backup cost is based on the operating system disk size of the VM that needs backup, the backup policy selected, and estimated moderate daily data churn. Storage consumed by Azure Backup grows incrementally over a period; the monthly cost of Azure Back also increases as consumption grows. Azure models the growth of consumed storage over a year, aggregates it and uses the average monthly consumption to propose the estimate.
Azure Site Recovery	Estimate for recovering a VM to Azure.

Monitoring

Expand table

Resource or feature	Notes
Alerts	Estimate for the selected alerting rules.
Boot Diagnostics	Estimate for enabling boot diagnostics can vary based the storage account and number of transactions performed. If enabled, a charged is incurred. However, due to the variability in consumption, the charge isn't included in the estimate.

Advanced

Expand table

Resource or feature	Notes
Azure Dedicated Host	Adding an existing Azure Dedicated Host deducts the cost of the VM size as the host had been prepaid for.

Next steps

The Azure pricing calculator is another tool that can be referenced for price estimations for different parts of your workload.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Availability options for Azure Virtual Machines

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article provides an overview of the availability options for Azure virtual machines (VMs).

Availability zones

Availability zones expands the level of control you have to maintain the availability of the applications and data on your VMs. An Availability Zone is a physically separate zone, within an Azure region. There are three Availability Zones per supported Azure region.

Each Availability Zone has a distinct power source, network, and cooling. By designing your solutions to use replicated VMs in zones, you can protect your apps and data from the loss of a data center. If one zone is compromised, then replicated apps and data are instantly available in another zone.

> ⓘ **Note**
>
> Regional resources may or may not exist in an Availability zone, and there is no insight into what physical or logical zone a regional resource is in. A failure in any of the availability zones in a region has the potential to bring down a regional VM.

Virtual Machines Scale Sets

Azure virtual machine scale sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update many VMs. There is no cost for the scale set itself, you only pay for each VM instance that you create.

Virtual machines in a scale set can also be deployed into multiple availability zones, a single availability zone, or regionally. Availability zone deployment options may differ based on the orchestration mode.

Availability sets

An availability set is a logical grouping of VMs that allows Azure to understand how your application is built to provide for redundancy and availability. We recommended that two or more VMs are created within an availability set to provide for a highly available application and to meet the 99.95% Azure SLA. There is no cost for the Availability Set itself, you only pay for each VM instance that you create.

Load balancer

Combine the Azure Load Balancer with availability zones and scale sets to get the most application resiliency. The Azure Load Balancer distributes traffic between multiple virtual machines. For our Standard tier virtual machines, the Azure Load Balancer is included. Not all virtual machine tiers include the Azure Load Balancer. For more information about load balancing your virtual machines, see the Load Balancer quickstarts using the CLI or PowerShell.

Azure Storage redundancy

Azure Storage always stores multiple copies of your data so that it is protected from planned and unplanned events, including transient hardware failures, network or power outages, and massive natural disasters. Redundancy ensures that your storage account meets its availability and durability targets even in the face of failures.

When deciding which redundancy option is best for your scenario, consider the tradeoffs between lower costs and higher availability. The factors that help determine which redundancy option you should choose include:

- How your data is replicated in the primary region
- Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters
- Whether your application requires read access to the replicated data in the secondary region if the primary region becomes unavailable for any reason

For more information, see Azure Storage redundancy

Azure Site Recovery

As an organization you need to adopt a business continuity and disaster recovery (BCDR) strategy that keeps your data safe, and your apps and workloads online, when planned and unplanned outages occur.

Azure Site Recovery helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery replicates workloads running on physical and virtual

machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to secondary location, and access apps from there. After the primary location is running again, you can fail back to it.

Site Recovery can manage replication for:

- Azure VMs replicating between Azure regions.
- On-premises VMs, Azure Stack VMs, and physical servers.

Next steps

- Create a virtual machine in an availability zone
- Create a virtual machine in an availability set
- Create a virtual machine scale set

Last updated on 12/05/2025

What are availability zones?

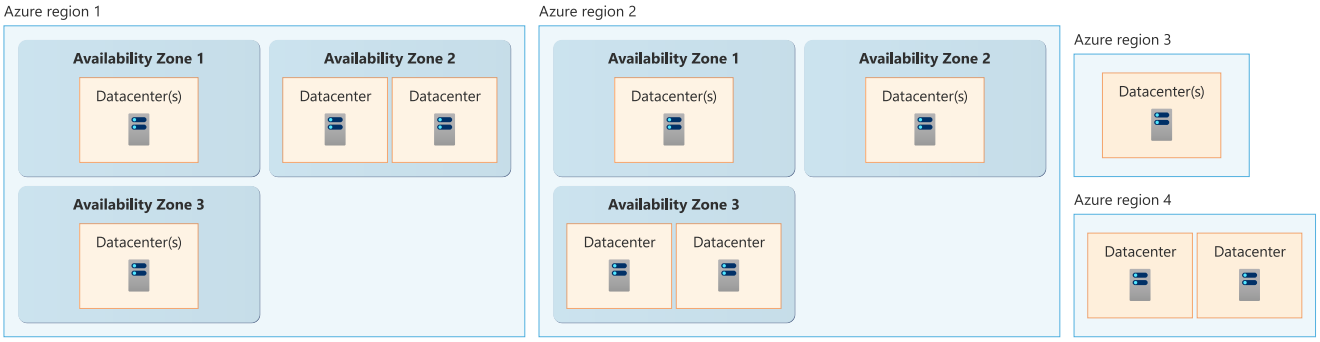
Summarize this article for me

https://learn-video.azurefd.net/vod/player?id=d36b5b2d-8bd2-43df-a796-b0c77b2f82fc&locale=en-us&embedUrl=%2Fazure%2Freliability%2Favailability-zones-overview

Many Azure regions provide *availability zones*, which are separated groups of datacenters within a region. Each availability zone has independent power, cooling, and networking infrastructure, so that if one zone experiences an outage, then regional services, capacity, and high availability are supported by the remaining zones. Some Azure services automatically use multiple zones, while others require you to configure multiple-zone deployment.

Availability zones don't only protect against large-scale outages that affect an entire zone. They also provide resilience against smaller-scoped failures, such as a server rack or cluster failure within a zone. In these scenarios, workloads can continue running in other zones, even if parts of the affected zone remain operational.

The following diagram shows several example Azure regions. Regions 1 and 2 support availability zones, and regions 3 and 4 don't have availability zones. Some zones have one datacenter and others have multiple.



> **Tip**
>
> To see which regions support availability zones, see List of Azure regions.

Datacenters and availability zones

An availability zone is a *logical grouping* of one or more physically separate datacenters within a region. Each availability zone is built in a way that if something goes wrong in one (like a power outage or network issue), the others keep working. A single datacenter doesn't offer this level of protection on its own.

Availability zones are typically separated by several kilometers, and usually are within 100 kilometers of each other. This provides low-latency connectivity while reducing the likelihood local outages or weather events impact multiple zones simultaneously.

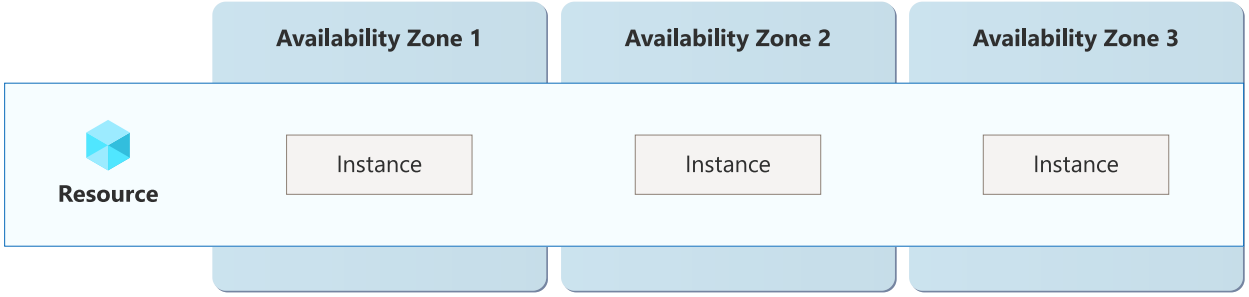
Datacenter locations are selected by using rigorous vulnerability risk assessment criteria. This process identifies all significant datacenter-specific risks and considers shared risks between availability zones.

Types of availability zone support

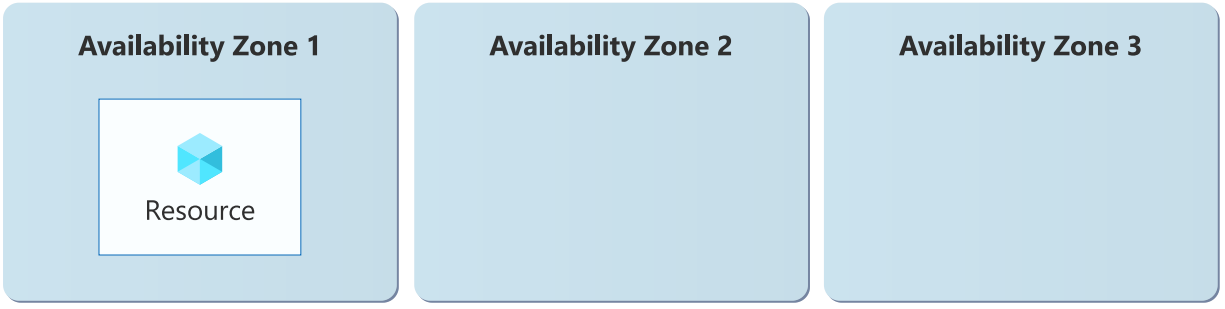
Azure services can provide different types of availability zone support for their resources, and most services support deploying either *zone-redundant* or *zonal* resources:

- **Zone-redundant resources**: Zone-redundant resources are replicated or distributed across multiple availability zones by the service. For example, zone-redundant data services replicate the data across multiple zones so that a failure in one zone doesn't affect the availability of the data. Some services are automatically zone-redundant in supported regions, while other services require that you configure your resource to be zone-redundant. For most services, Microsoft selects the zones your resources use, but sometimes you can select the set of zones.

  With zone-redundant deployments, Microsoft manages spreading requests across zones and the replication of data across zones. If an outage occurs in an availability zone, Microsoft manages failover to another zone automatically.



- **Zonal resources**: A zonal resource is deployed to a single availability zone that you select yourself.



Zonal deployments don't automatically provide resiliency to availability zone outages. However, zonal resources are isolated from faults in other zones. They can also help you achieve unusually stringent latency or performance requirements. For example, for a chatty workload built using virtual machines, you might choose to deploy multiple virtual machines to the same zone to reduce the latency of the connections between them.

To make zonal resources resilient to availability zone outages, you need to design an architecture with separate resources in multiple availability zones within the region. Microsoft doesn't manage the process for you. If an outage occurs in an availability zone, you're responsible for failover to another zone.



When you configure a resource to be zone redundant, or if you use multiple instances of a zonal resource in different availability zones, then your resource is considered to be *zone-resilient*: that is, it's resilient to the outage of a single availability zone. To learn more about how to use zonal deployments and maintain zone resiliency, see Zonal resources and zone resiliency.

When designing your reliability strategy, make sure that you understand how each service in your workload supports availability zones. For example, some services may have extra requirements to meet for availability zone support, such as certain tiers or SKUs. Reliability guides contain details of any such requirements.

If a resource isn't configured to use availability zones, either due to the region you use not supporting zones or because of your configuration choices, it's called a *nonzonal* or *regional* deployment. Azure might place nonzonal resources across any zones in the region. You don't choose which resources go into which zones. If any availability zone in the region experiences an outage, nonzonal resources might be in the affected zone and could experience downtime.

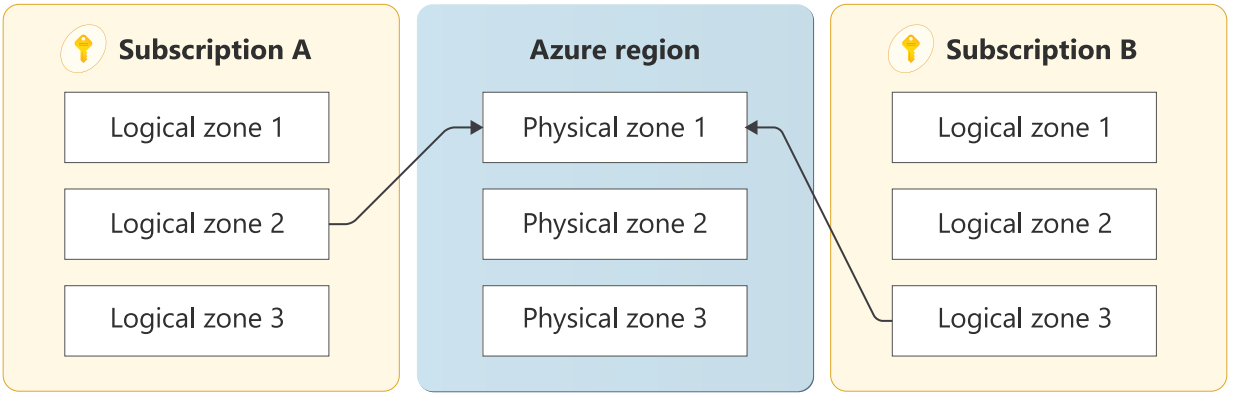
Configuring resources for availability zone support

Each service has its own method for configuring availability zone support, and some services don't require any configuration. To learn about how each service supports availability zones and how to configure that support, see Azure reliability guides by service.

Physical and logical availability zones

Each datacenter is assigned to a physical zone. Physical zones are mapped to logical zones in your Azure subscription, and different subscriptions might have a different mapping order. Azure subscriptions are automatically assigned their mapping at the time the subscription is created. Because of this, the zone mapping for one subscription could be different for other subscriptions.

For example, subscription A may have physical zone 1 mapped to logical zone 2, while subscription B has physical zone 1 mapped to logical zone 3:



To understand the mapping between logical and physical zones for your subscription, use the Azure CLI or Azure PowerShell, or the Azure Resource Manager APIs.

Azure CLI

Use the az account list-locations command:

Azure CLI

```
az account list-locations \
    --query "[?availabilityZoneMappings].{availabilityZoneMappings:
availabilityZoneMappings, displayName: displayName, name: name}"
```

Availability zones and Azure updates

For each region, Microsoft aims to deploy updates to Azure services within a single availability zone at a time. This approach reduces the impact that updates might have on an active workload, allowing the workload to continue to run in other zones while the update is in process. To take advantage of sequenced zone updates, your workload must be already configured to run across multiple zones. For more information about how Azure deploys updates, see Advancing safe deployment practices.

Inter-zone networking

Within each region, availability zones are connected through a high-performance network. Microsoft strives to achieve an inter-zone communication with round-trip latency of less than approximately 2 milliseconds. Low latency allows for high-performance communication within a region, and for synchronous replication of data across multiple availability zones. The target latency refers to the latency of the network links. Depending on the communication protocol you use and the network hops required for any specific network flow, the latency you observe might be different.

In most workloads, you can distribute components of your solution across availability zones without a noticeable effect on your performance. If you have a workload with a high degree of sensitivity to inter-zone latency, it's important to test the latency between your selected availability zones with your actual protocols and configuration. To reduce inter-zone traffic, it's possible to use zonal deployments, but optimally, you should use multiple availability zones in your reliability strategy plan. To learn more about how to use zonal deployments and maintain zone resiliency, see Zonal resources and zone resiliency.

Azure doesn't charge for data transfer between availability zones in the same region, whether you use private or public IP addressing.

Availability zone architectural guidance

To achieve reliable workloads:

- Production workloads should be configured to use multiple availability zones if the region they are in supports availability zones.
- For mission-critical workloads, you should consider a solution that is *both* multi-region and multi-zone.

For more detailed information on how to use regions and availability zones in a solution architecture, see Recommendations for using availability zones and regions.

Related content

- What are Azure regions?
- List of Azure regions
- Azure services with availability zones
- Recommendations for using availability zones and regions

Last updated on 02/11/2026

Create a virtual machine in an availability zone using Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article steps through using the Azure CLI to create a Linux VM in an Azure availability zone. An availability zone is a physically separate zone in an Azure region. Use availability zones to protect your apps and data from an unlikely failure or loss of an entire datacenter.

To use an availability zone, create your virtual machine in a supported Azure region.

Make sure that you have installed the latest Azure CLI and logged in to an Azure account with az login.

Check VM SKU availability

The availability of VM sizes, or SKUs, may vary by region and zone. To help you plan for the use of Availability Zones, you can list the available VM SKUs by Azure region and zone. This ability makes sure that you choose an appropriate VM size, and obtain the desired resiliency across zones. For more information on the different VM types and sizes, see VM Sizes overview.

You can view the available VM SKUs with the az vm list-skus command. The following example lists available VM SKUs in the *eastus2* region:

Azure CLI

```
az vm list-skus --location eastus2 --output table
```

The output is similar to the following condensed example, which shows the Availability Zones in which each VM size is available:

Output

```
ResourceType      Locations  Name              [...]    Tier       Size
Zones
----------------  ---------  ----------------          ---------  -------
-------
virtualMachines   eastus2    Standard_DS1_v2           Standard   DS1_v2
1,2,3
virtualMachines   eastus2    Standard_DS2_v2           Standard   DS2_v2
```

```
1,2,3
[...]
virtualMachines   eastus2    Standard_F1s               Standard   F1s
1,2,3
virtualMachines   eastus2    Standard_F2s               Standard   F2s
1,2,3
[...]
virtualMachines   eastus2    Standard_D2s_v3            Standard   D2_v3
1,2,3
virtualMachines   eastus2    Standard_D4s_v3            Standard   D4_v3
1,2,3
[...]
virtualMachines   eastus2    Standard_E2_v3             Standard   E2_v3
1,2,3
virtualMachines   eastus2    Standard_E4_v3             Standard   E4_v3
1,2,3
```

Create resource group

Create a resource group with the az group create command.

An Azure resource group is a logical container into which Azure resources are deployed and managed. A resource group must be created before a virtual machine. In this example, a resource group named *myResourceGroupVM* is created in the *eastus2* region. East US 2 is one of the Azure regions that supports availability zones.

Azure CLI

```
az group create --name myResourceGroupVM --location eastus2
```

The resource group is specified when creating or modifying a VM, which can be seen throughout this article.

Create virtual machine

Create a virtual machine with the az vm create command.

When creating a virtual machine, several options are available such as operating system image, disk sizing, and administrative credentials. In this example, a virtual machine is created with a name of *myVM* running Ubuntu Server. The VM is created in availability zone *1*. By default, the VM is created in the *Standard_DS1_v2* size.

Azure CLI

```
az vm create --resource-group myResourceGroupVM --name myVM --location
eastus2 --image Ubuntu2204 --generate-ssh-keys --zone 1
```

It may take a few minutes to create the VM. Once the VM has been created, the Azure CLI outputs information about the VM. Take note of the `zones` value, which indicates the availability zone in which the VM is running.

Output


```
{
  "fqdns": "",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroupVM/providers/Microsoft.Compute/vi
rtualMachines/myVM",
  "location": "eastus2",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.174.34.95",
  "resourceGroup": "myResourceGroupVM",
  "zones": "1"
}
```


Confirm zone for managed disk and IP address

When the VM is deployed in an availability zone, a managed disk for the VM is created in the same availability zone. By default, a public IP address is also created in that zone. The following examples get information about these resources.

To verify that the VM's managed disk is in the availability zone, use the az vm show command to return the disk ID. In this example, the disk ID is stored in a variable that is used in a later step.

Azure CLI

```
osdiskname=$(az vm show -g myResourceGroupVM -n myVM --query
"storageProfile.osDisk.name" -o tsv)
```

Now you can get information about the managed disk:

Azure CLI

```
az disk show --resource-group myResourceGroupVM --name $osdiskname
```

The output shows that the managed disk is in the same availability zone as the VM:

Output


```
{
  "creationData": {
    "createOption": "FromImage",
    "imageReference": {
      "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/Providers/Microsoft.Compute/Locations/westeurope/Publishers/Can
onical/ArtifactTypes/VMImage/Offers/UbuntuServer/Skus/16.04-
LTS/Versions/latest",
      "lun": null
    },
    "sourceResourceId": null,
    "sourceUri": null,
    "storageAccountId": null
  },
  "diskSizeGb": 30,
  "encryptionSettings": null,
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroupVM/providers/Microsoft.Compute/di
sks/osdisk_761c570dab",
  "location": "eastus2",
  "managedBy": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroupVM/providers/Microsoft.Compute/vi
rtualMachines/myVM",
  "name": "myVM_osdisk_761c570dab",
  "osType": "Linux",
  "provisioningState": "Succeeded",
  "resourceGroup": "myResourceGroupVM",
  "sku": {
    "name": "Premium_LRS",
    "tier": "Premium"
  },
  "tags": {},
  "timeCreated": "2018-03-05T22:16:06.892752+00:00",
  "type": "Microsoft.Compute/disks",
  "zones": [
    "1"
  ]
}
```


Use the az vm list-ip-addresses command to return the name of public IP address resource in *myVM*. In this example, the name is stored in a variable that is used in a later step.

Azure CLI

```
ipaddressname=$(az vm list-ip-addresses -g myResourceGroupVM -n myVM --query
"[].virtualMachine.network.publicIpAddresses[].name" -o tsv)
```

Now you can get information about the IP address:

Azure CLI

```
az network public-ip show --resource-group myResourceGroupVM --name
$ipaddressname
```

The output shows that the IP address is in the same availability zone as the VM:

Output


```
{
  "dnsSettings": null,
  "etag": "W/\"b7ad25eb-3191-4c8f-9cec-c5e4a3a37d35\"",
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroupVM/providers/Microsoft.Network/pu
blicIPAddresses/myVMPublicIP",
  "idleTimeoutInMinutes": 4,
  "ipAddress": "52.174.34.95",
  "ipConfiguration": {
    "etag": null,
    "id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroupVM/providers/Microsoft.Network/ne
tworkInterfaces/myVMVMNic/ipConfigurations/ipconfigmyVM",
    "name": null,
    "privateIpAddress": null,
    "privateIpAllocationMethod": null,
    "provisioningState": null,
    "publicIpAddress": null,
    "resourceGroup": "myResourceGroupVM",
    "subnet": null
  },
  "location": "eastUS2",
  "name": "myVMPublicIP",
  "provisioningState": "Succeeded",
  "publicIpAddressVersion": "IPv4",
  "publicIpAllocationMethod": "Dynamic",
  "resourceGroup": "myResourceGroupVM",
  "resourceGuid": "8c70a073-09be-4504-0000-000000000000",
  "tags": {},
  "type": "Microsoft.Network/publicIPAddresses",
  "zones": [
    "1"
  ]
}
```


Next steps

In this article, you learned how to create a VM in an availability zone. Learn more about availability for Azure VMs.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a virtual machine in an availability zone using Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Windows VMs

This article details using Azure PowerShell to create an Azure virtual machine running Windows Server 2016 in an Azure availability zone. An availability zone is a physically separate zone in an Azure region. Use availability zones to protect your apps and data from an unlikely failure or loss of an entire datacenter.

To use an availability zone, create your virtual machine in a supported Azure region.

Sign in to Azure

Sign in to your Azure subscription with the `Connect-AzAccount` command and follow the on-screen directions.

PowerShell

```
Connect-AzAccount
```

Check VM SKU availability

The availability of VM sizes, or SKUs, may vary by region and zone. To help you plan for the use of Availability Zones, you can list the available VM SKUs by Azure region and zone. This ability makes sure that you choose an appropriate VM size, and obtain the desired resiliency across zones. For more information on the different VM types and sizes, see VM Sizes overview.

You can view the available VM SKUs with the Get-AzComputeResourceSku command. The following example lists available VM SKUs in the *eastus2* region:

PowerShell

```
Get-AzComputeResourceSku | where {$_.Locations.Contains("eastus2")};
```

The output is similar to the following condensed example, which shows the Availability Zones in which each VM size is available:

PowerShell

```
ResourceType                Name  Location      Zones   [...]
------------                ----  --------      -----
virtualMachines  Standard_DS1_v2   eastus2  {1, 2, 3}
virtualMachines  Standard_DS2_v2   eastus2  {1, 2, 3}
[...]
virtualMachines     Standard_F1s   eastus2  {1, 2, 3}
virtualMachines     Standard_F2s   eastus2  {1, 2, 3}
[...]
virtualMachines  Standard_D2s_v3   eastus2  {1, 2, 3}
virtualMachines  Standard_D4s_v3   eastus2  {1, 2, 3}
[...]
virtualMachines   Standard_E2_v3   eastus2  {1, 2, 3}
virtualMachines   Standard_E4_v3   eastus2  {1, 2, 3}
```

Create resource group

Create an Azure resource group with New-AzResourceGroup. A resource group is a logical container into which Azure resources are deployed and managed. In this example, a resource group named *myResourceGroup* is created in the *eastus2* region.

PowerShell

```
New-AzResourceGroup -Name myResourceGroup -Location EastUS2
```

Create networking resources

Create a virtual network, subnet, and a public IP address

These resources are used to provide network connectivity to the virtual machine and connect it to the internet. Create the IP address in an availability zone, *2* in this example. In a later step, you create the VM in the same zone used to create the IP address.

PowerShell

```
# Create a subnet configuration
$subnetConfig = New-AzVirtualNetworkSubnetConfig -Name mySubnet -
AddressPrefix 192.168.1.0/24

# Create a virtual network
$vnet = New-AzVirtualNetwork -ResourceGroupName myResourceGroup -Location
eastus2 `
    -Name myVNet -AddressPrefix 192.168.0.0/16 -Subnet $subnetConfig
```

```
# Create a public IP address in an availability zone and specify a DNS name
$pip = New-AzPublicIpAddress -ResourceGroupName myResourceGroup -Location
eastus2 -Zone 2 `
    -AllocationMethod Static -IdleTimeoutInMinutes 4 -Name
"mypublicdns$(Get-Random)" -Sku Standard
```

Create a network security group and a network security group rule

The network security group secures the virtual machine using inbound and outbound rules. In this case, an inbound rule is created for port 3389, which allows incoming remote desktop connections. We also want to create an inbound rule for port 80, which allows incoming web traffic.

PowerShell

```
# Create an inbound network security group rule for port 3389 - change -
Access to "Allow" if you want to allow RDP access
$nsgRuleRDP = New-AzNetworkSecurityRuleConfig -Name
myNetworkSecurityGroupRuleRDP  -Protocol Tcp `
    -Direction Inbound -Priority 1000 -SourceAddressPrefix * -
SourcePortRange * -DestinationAddressPrefix * `
    -DestinationPortRange 3389 -Access Deny

# Create an inbound network security group rule for port 80 - - change -
Access to "Allow" if you want to allow TCP traffic over port 80
$nsgRuleWeb = New-AzNetworkSecurityRuleConfig -Name
myNetworkSecurityGroupRuleWWW  -Protocol Tcp `
    -Direction Inbound -Priority 1001 -SourceAddressPrefix * -
SourcePortRange * -DestinationAddressPrefix * `
    -DestinationPortRange 80 -Access Deny

# Create a network security group
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName myResourceGroup -
Location eastus2 `
    -Name myNetworkSecurityGroup -SecurityRules $nsgRuleRDP,$nsgRuleWeb
```

Create a network card for the virtual machine

Create a network card with New-AzNetworkInterface for the virtual machine. The network card connects the virtual machine to a subnet, network security group, and public IP address.

PowerShell

```
# Create a virtual network card and associate with public IP address and NSG
$nic = New-AzNetworkInterface -Name myNic -ResourceGroupName myResourceGroup
```

```
-Location eastus2 `
    -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -
NetworkSecurityGroupId $nsg.Id
```

Create virtual machine

Create a virtual machine configuration. This configuration includes the settings that are used when deploying the virtual machine such as a virtual machine image, size, and authentication configuration. The *Standard_DS1_v2* size in this example is supported in availability zones. This configuration also specifies the availability zone you set when creating the IP address. When running this step, you are prompted for credentials. The values that you enter are configured as the user name and password for the virtual machine.

PowerShell

```
# Define a credential object
$cred = Get-Credential

# Create a virtual machine configuration
$vmConfig = New-AzVMConfig -VMName myVM -VMSize Standard_DS1_v2 -Zone 2 | `
    Set-AzVMOperatingSystem -Windows -ComputerName myVM -Credential $cred |
`
    Set-AzVMSourceImage -PublisherName MicrosoftWindowsServer -Offer
WindowsServer `
    -Skus 2016-Datacenter -Version latest | Add-AzVMNetworkInterface -Id
$nic.Id
```

Create the virtual machine with New-AzVM.

PowerShell

```
New-AzVM -ResourceGroupName myResourceGroup -Location eastus2 -VM $vmConfig
```

Confirm zone for managed disk

You created the VM's IP address resource in the same availability zone as the VM. The managed disk resource for the VM is created in the same availability zone. You can verify this with Get-AzDisk:

PowerShell

```
Get-AzDisk -ResourceGroupName myResourceGroup
```

The output shows that the managed disk is in the same availability zone as the VM:

```PowerShell
ResourceGroupName  : myResourceGroup
AccountType        : PremiumLRS
OwnerId            : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroup/providers/Microsoft.
                     Compute/virtualMachines/myVM
ManagedBy          : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx//resourceGroups/myResourceGroup/providers/Microsoft.
                     Compute/virtualMachines/myVM
Sku                : Microsoft.Azure.Management.Compute.Models.DiskSku
Zones              : {2}
TimeCreated        : 9/7/2017 6:57:26 PM
OsType             : Windows
CreationData       : Microsoft.Azure.Management.Compute.Models.CreationData
DiskSizeGB         : 127
EncryptionSettings :
ProvisioningState  : Succeeded
Id                 : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/myResourceGroup/providers/Microsoft.

Compute/disks/myVM_OsDisk_1_bd921920bb0a4650becfc2d830000000
Name               : myVM_OsDisk_1_bd921920bb0a4650becfc2d830000000
Type               : Microsoft.Compute/disks
Location           : eastus2
Tags               : {}
```

Next steps

In this article, you learned how to create a VM in an availability zone. Learn more about availability for Azure VMs.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machines in an availability zone using the Azure portal

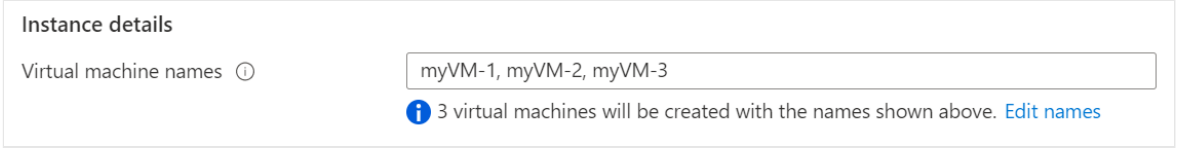
Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

This article steps through using the Azure portal to create highly resilient virtual machines in availability zones. Azure availability zones are physically separate locations within each Azure region that are tolerant to local failures. Use availability zones to protect your applications and data against unlikely datacenter failures.

To use availability zones, create your virtual machines in a supported Azure region.

1. Sign in to the Azure portal.
2. Click **Create a resource** > **Compute** > **Virtual machine**.
3. In the **Virtual machines** page, select **Create** and then **Virtual machine**. The **Create a virtual machine** page opens.
4. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose a resource group or create a new one.
5. Under **Instance details**, type a name for the **Virtual machine name**.
6. For **Availability options**, leave the default of **Availability zone**.
7. For **Availability zone**, the drop-down defaults to *Zone 1*. If you choose multiple zones, a new VM is created in each zone. For example, if you select all three zones, then three VMs are created. The VM names are the original name you entered, with **-1**, **-2**, and **-3** appended to the name based on number of zones selected. If you want, you can edit each of the default VM names.



8. Complete the rest of the page as usual. If you want to create a load balancer, go to the **Networking** tab > **Load Balancing** > **Load balancing options**. You can choose either an Azure load balancer or an Application gateway.

   For an **Azure load balancer**:
   a. You can select an existing load balancer or select **Create a load balancer**.

b. To create a new load balancer, for **Load balancer name** type a load balancer name.
c. Select the **Type** of load balancer, either *Public* or *Internal*.
d. Select the **Protocol**, either **TCP** or **UDP**.
e. You can leave the default **Port** and **Backend port**, or change them if needed. The backend port you select will be opened up on the Network Security Group (NSG) of the VM.
f. When you're done, select **Create**.

For an **Application Gateway**:

a. Select either an existing application gateway or **Create an application gateway**.
b. To create a new gateway, type the name for the application gateway. The Application Gateway can load balance multiple applications. Consider naming the Application Gateway according to the workloads you wish to load balance, rather than specific to the virtual machine name.
c. In **Routing rule**, type a rule name. The rule name should describe the workload you are load balancing.
d. For HTTP load balancing, you can leave the defaults and then select **Create**. For HTTPS load balancing, you have two options:

- Upload a certificate and add the password (application gateway manages certificate storage). For certificate name, type a friendly name for the certificate.
- Use a key vault (application gateway will pull a defined certificate from a defined key vault). Select your **Managed identity**, **Key Vault**, and **Certificate**.

> **ⓘ Important**
>
> After the VMs and application gateway are deployed, log in to the VMs to ensure that either the application gateway certificate is uploaded onto the VMs or the domain name of the VM certificate matches with the domain name of the application gateway.

> **ⓘ Note**
>
> A separate subnet will be defined for Application Gateway upon creation. For more information, see Application Gateway infrastructure configuration.

9. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.

10. On the **Create a virtual machine** page, you can see the details about the VM you are about to create. When you're ready, select **Create**.

11. If you are creating a Linux VM and the **Generate new key pair** window opens, select **Download private key and create resource**. Your key file will download as **myKey.pem**.

12. When the deployment is finished, select **Go to resource**.

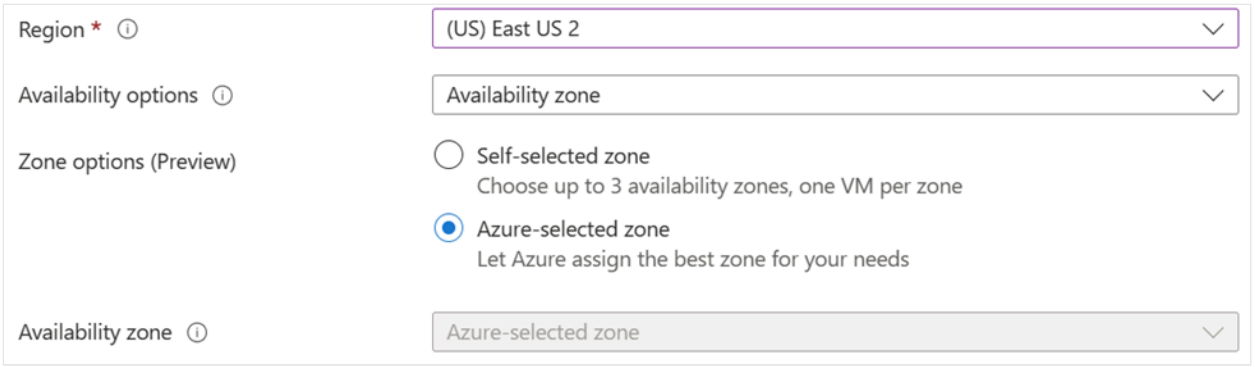
Create a Zonal Virtual Machine in an Azure-selected Zone

> **Important**
>
> These features are currently in PREVIEW. See the Supplemental Terms of Use for Microsoft Azure Previews for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

If you would like the resiliency benefits of a Zonal Virtual Machine and want help with selecting a zone, you can have Azure select the best-fit zone for your deployment.

> **Note**
>
> Please review the Restrictions section before deploying your VM using an Azure-Selected Zone.

1. Sign in to the Azure portal using this link enabling the feature.
2. In the Basics tab, under Project details, make sure the correct subscription is selected and then choose a resource group or create a new one.
3. Under Instance details, type a name for the Virtual machine name.
4. For Availability options, Ensure Availability zone is selected.
5. For Zone options (Preview), select Azure-selected zone. The Availability zone selection should now be grayed out.
6. Configure the rest of your Virtual Machine as normal for deployment.



Restrictions

Regions

Deploying a VM into an Azure-selected zone is available in all Zonal Regions except the following regions.

- BrazilSouth
- CentralIndia
- EastUS
- JapanEast
- KoreaCentral

Disks

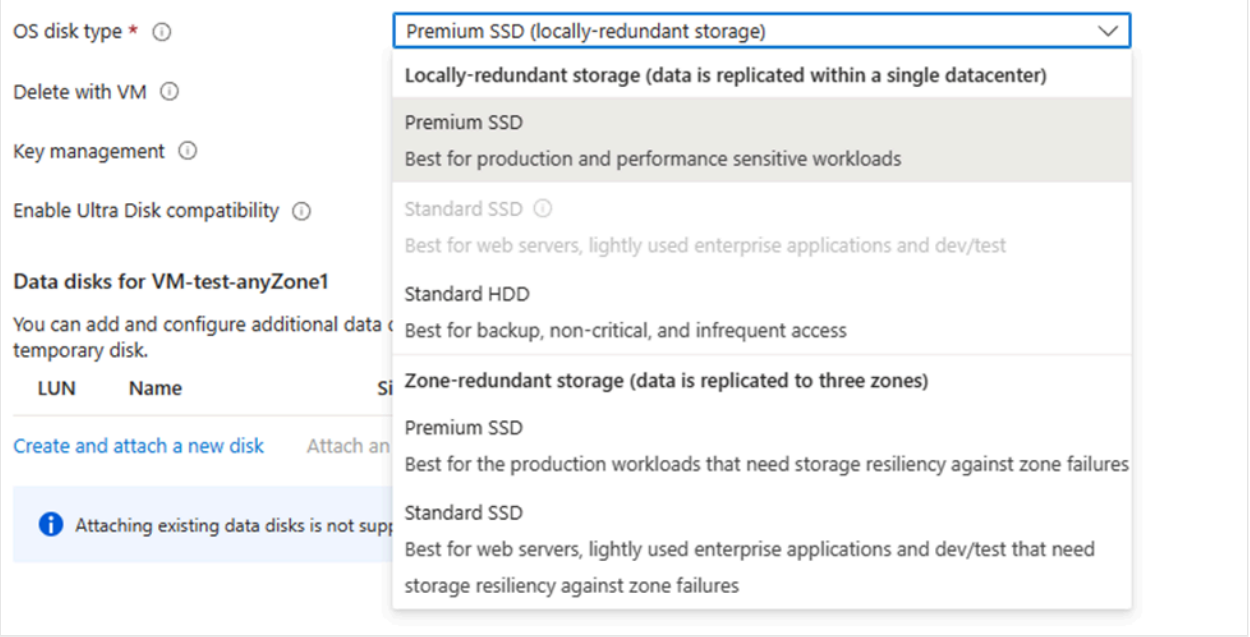
Existing OS and Data Disks are currently not supported. New Data Disk Source Type must be "none".

Supported Disk types

- Standard HDDs (hard disk drives)
- Premium SSDs
- Premium SSD V2
- Ultra Disks
- Standard SSD ZRS

Unsupported Disk types

- Standard SSD LRS



Other resources

- Existing Public IPs aren't supported for an Azure-selected zone. They can be added after deployment.
- New Public IPs must be Zone Redundant or use an Azure-Selected Zone.
- Load Balancers and Application Gateways must be Zonally Resilient to attach during the VM deployment.
- Proximity Placement Groups, Capacity Reservation Groups, and Azure Dedicated Host aren't supported.
- Site Recovery configuration isn't available during VM creation but can be configured after deployment.

Next steps

In this article, you learned how to create a VM in an availability zone. Learn more about availability for Azure VMs.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Reliability in Azure Virtual Machines

Azure Virtual Machines provides on-demand, scalable compute resources. As a foundational infrastructure service, it's designed to deliver enterprise-grade reliability and availability for mission-critical workloads.

When you use Azure, reliability is a shared responsibility. Microsoft provides a range of capabilities to support resiliency and recovery. You're responsible for understanding how those capabilities work within all of the services you use, and selecting the capabilities you need to meet your business objectives and uptime goals.

This article describes how to make Virtual Machines resilient to a variety of potential outages and problems, including transient faults, availability zone outages, region outages, and service maintenance. It also describes how you can use backups to recover from other types of problems, and highlights some key information about the Virtual Machines service level agreement (SLA).

> **Important**
>
> When you consider the reliability of a virtual machine (VM), you also need to consider the reliability of your disks, network infrastructure, and applications that run on your VMs. Improving the resiliency of the VM alone might have limited impact if the other components aren't equally resilient. Depending on your resiliency requirements, you might need to make configuration changes across multiple areas.

Production deployment recommendations

For more information about how to deploy VMs to support your solution's reliability requirements and how reliability affects other aspects of your architecture, see Architecture best practices for Virtual Machines and scale sets in the Azure Well-Architected Framework.

Reliability architecture overview

VMs are the fundamental compute unit in Azure, whether you provision the VMs yourself or use other Azure compute services that transparently provision and manage them for you.

An individual VM is also known as a *single instance VM*. It runs on a specific host, which is a physical server. Most VMs share their host with other VMs.

When you create your VMs, you can influence where they run within the underlying infrastructure. Typically, you make placement decisions based on your requirements for reliability, latency, and isolation. Azure provides several configuration options that affect how your VMs are placed.

- **Region:** You can select which Azure region that your VM should run in. A region is a geographic area that might contain multiple datacenters, each with a large number of hosts.

- **Availability zone:** Availability zones are physically separate groups of datacenters within each Azure region. In regions that support availability zones, you can select which zone the VM runs in. For more information, see Resilience to availability zone failures.

- **Availability sets:** An availability set is a logical grouping of VMs that allows Azure to understand how your application is built to provide for redundancy and availability.

  When you use availability sets, Azure distributes a group of VMs across different fault domains. This distribution minimizes the risk of localized hardware failures by grouping VMs that share a common power source and network switch.

  Availability sets can also place different VMs in different update domains, which controls how the Azure platform rolls out platform updates. By using update domains, you can ensure that only a subset of your VMs is restarted for updates at one time.

- **Proximity placement groups:** For workloads that need to achieve the lowest possible latency between VMs, you can use a proximity placement group to ensure that Azure places the VMs physically close to each other. However, proximity placement means that an outage of the datacenter can affect all of the VMs in the group. To achieve high reliability, you might need to provision multiple proximity placement groups in different availability zones.

- **Dedicated hosts:** You can use Azure Dedicated Host to provision your own physical server that runs one or more VMs, such as for strict compliance requirements. However, when you provision a dedicated host, an outage in its datacenter can affect all of the VMs on that host. To achieve high reliability, you might need to provision multiple dedicated hosts in different availability zones.

If you create a set of VMs that perform similar functions, consider using Azure Virtual Machine Scale Sets to create and manage the VMs as a group. Scale sets also provide more reliability options, such as spreading the VMs across multiple availability zones.

For more information about availability for VMs, see Availability options for Virtual Machines.

Resilience to transient faults

Transient faults are short, intermittent failures in components. They occur frequently in a distributed environment like the cloud, and they're a normal part of operations. Transient faults correct themselves after a short period of time. It's important that your applications can handle transient faults, usually by retrying affected requests.

All cloud-hosted applications should follow the Azure transient fault handling guidance when they communicate with any cloud-hosted APIs, databases, and other components. For more information, see Recommendations for handling transient faults.

Applications that run on your VMs should implement appropriate fault-handling strategies to ensure that any temporary interruptions in service don't affect your workload.

Resilience to availability zone failures

Availability zones are physically separate groups of datacenters within an Azure region. When one zone fails, services can fail over to one of the remaining zones.

An individual VM can be deployed in a *zonal* configuration, which means that it's pinned to a single availability zone that you select. By itself, a zonal VM isn't resilient to zone outages. However, you can create multiple VMs and place them in different availability zones, then spread your applications and data across the VM instances. Alternatively, you can use virtual machine scale sets to deploy a set of VMs across multiple availability zones.

If you don't configure a VM to be zonal, it's considered *nonzonal* or *regional*. Nonzonal VMs might be placed in any availability zone within the region. If any availability zone in the region experiences an outage, nonzonal VMs might be in the affected zone and can experience downtime.

Requirements

Region support: Zonal VMs can be deployed into any region that supports availability zones.

However, some VM types and sizes are only available in specific regions, or specific zones within a region. To check which regions and zones support the VM types that you need, use the following resources:

- To check the VM types available in each region, see Products available by region.
- To check the supported VM types and sizes within each zone of a specific region, see Check VM SKU availability.

Cost

There's no cost difference between a zonal and nonzonal VM.

Configure availability zone support

This section explains how to configure availability zone support for your VM instance.

> **Note**
>
> When you select which availability zones to use, you're actually selecting the *logical availability zone*. If you deploy other workload components in a different Azure subscription, they might use a different logical availability zone number to access the same physical availability zone. For more information, see Physical and logical availability zones.

- **Create a zonal VM.** You can create a zonal VM by using the following guides:
  - Create VMs in an availability zone by using the Azure portal
  - Create a VM in an availability zone by using the Azure CLI
  - Create a VM in an availability zone by using Azure PowerShell
- **Convert existing VMs to a zonal configuration.** You can move from a nonzonal VM to a zonal VM. This process creates a new VM in the target availability zone and requires that the VM be stopped during the move process. For more information, see Move Azure single instance VMs from regional to zonal target availability zones.
- **Change the availability zone of an existing zonal VM.** Zonal VMs can't be moved to a different availability zone. You need to deploy a new zonal VM in the desired availability zone instead.

- **Convert a zonal VM to a nonzonal configuration.** Zonal VMs can't be converted to a nonzonal configuration. You need to deploy a new nonzonal VM instead.

Behavior when all zones are healthy

This section describes what to expect when VM instances are configured with availability zone support and all availability zones are operational.

- **Traffic routing between zones:** You're responsible for routing traffic between VMs, including VMs that are in different availability zones. Common approaches include Azure Load Balancer and Azure Application Gateway. For more information, see Load balancing options.

- **Data replication between zones:** You're responsible for any data replication that needs to happen between VMs, including across VMs in different availability zones. Databases and other similar stateful applications that run on VMs often provide capabilities to replicate data.

Behavior during a zone failure

This section describes what to expect when VM instances are configured with availability zone support and there's an outage in their availability zones.

- **Detection and response:** You're responsible for detecting and responding to zone failures that affect your VMs.

- **Notification**: Microsoft doesn't automatically notify you when a zone is down. However, you can use Azure Resource Health to monitor for the health of an individual resource, and you can set up Resource Health alerts to notify you of problems. You can also use Azure Service Health to understand the overall health of the service, including any zone failures, and you can set up Service Health alerts to notify you of problems.

- **Active requests:** Any active requests or other work that occurs on the VM during the zone failure are likely to be terminated.

- **Expected data loss:** Zonal LRS disks might be unavailable during a zone failure.

  If you use zone-redundant storage (ZRS) disks and an outage affects your VM, you can force detach your ZRS disks from the failed VM. This approach allows you to attach the ZRS disks to another VM.

For more information about how to make disks resilient to availability zone failures and other problems, see Reliability in Azure Disk Storage.

- **Expected downtime:** VMs remain down until the availability zone recovers.
- **Traffic rerouting:** You're responsible for rerouting traffic to other VMs in healthy zones.

  If you configure a zone-resilient load balancer and it performs health checks, the load balancer typically detects failed VMs and can route traffic to other VM instances in healthy zones.

Zone recovery

When the zone is healthy, VMs in the zone restart. You're responsible for any zone recovery procedures and data synchronization that your workloads require.

Test for zone failures

You can use Azure Chaos Studio to simulate the loss of a VM as part of an experiment. Chaos Studio provides built-in faults for VMs, including the ability to shut down a VM. You can use these capabilities to simulate zone-level failures and test your failover processes.

Custom multi-zone solutions for resiliency

When you deploy multiple VMs into different zones, you're responsible for configuring and managing replication, load balancing, failover, and failback processes.

Some applications provide built-in capabilities that can help when you deploy across multiple VMs. For example, SQL Server on Azure VMs provides a set of capabilities to simplify your configuration and management processes across availability zones.

You can consider using Azure Site Recovery zone-to-zone disaster recovery (DR) when your application runs in a single zone at a time and you don't require near-instant failover between zones. Zone-to-zone DR has some important limitations, so review your requirements thoroughly.

Resilience to region-wide failures

VMs are single-region resources. If the region becomes unavailable, your VM is also unavailable.

Custom multi-region solutions for resiliency

You can deploy multiple VMs into different regions, but you need to implement replication, load balancing, and failover processes.

Site Recovery is a service that enables DR by replicating VMs and their data to a secondary region. You can select almost any Azure region as your secondary region, including nonpaired region combinations. For more information, see Azure to Azure DR architecture.

Some applications create clusters or other constructs to replicate data and distribute work across multiple VMs, including in different regions. These applications can simplify the configuration of a multi-region solution.

For an example architecture that illustrates using VMs across multiple regions, see Multi-region load balancing with Azure Traffic Manager, Azure Firewall, and Application Gateway.

Resilience to service maintenance

Azure performs regular periodic maintenance on VMs to ensure reliability. There are multiple ways that you can ensure your workloads remain operational during maintenance activities:

- When you use availability sets or virtual machine scale sets, you can configure update domains. Update domains help distribute maintenance activities across different VMs at different times, so your VMs don't all restart simultaneously.
- You can customize the timing that maintenance is applied to your VMs by using *maintenance control*. You can use maintenance configurations to schedule it at a time that suits your workload.
- You can receive notifications of upcoming maintenance activities.

For more information, see Guest updates and host maintenance overview.

Backup and restore

Azure Backup provides native backup support for VMs. Azure Backup creates and manages backups, and provides application-consistent protection for the entire VM, including all attached disks. A VM backup solution with Azure Backup is ideal when you need coordinated backup of multiple disks or application-aware backups. However, for database workloads, consider application-specific backup solutions that provide transaction-consistent protection and faster recovery options.

With Azure Backup for VMs, you can customize the backup frequency, retention duration, and storage configuration to suit your needs. For more information, see Azure Backup for VMs.

Backup also supports disks that are attached to VMs. For more information, see Overview of Azure Disk Backup.

For most solutions, you shouldn't rely exclusively on backups. Instead, use the other capabilities described in this guide to support your resiliency requirements. However, backups protect against some risks that other approaches don't. For more information, see What are redundancy, replication, and backup?.

Service-level agreement

The service-level agreement (SLA) for Azure services describes the expected availability of each service and the conditions that your solution must meet to achieve that availability expectation. For more information, see SLAs for online services.

For Virtual Machines, the SLA provides a base level of availability. The uptime percentage defined in the SLA increases when you have two or more VMs and you take the following actions:

- Configure those VMs to be deployed across two or more availability zones.
- Configure those VMs to be deployed into an availability set.

Next steps

- VMs in Azure
- Virtual Machine Scale Sets

Related resource

- Reliability in Azure Disk Storage
- Reliability in Azure

Last updated on 01/23/2026

Move Azure single-instance virtual machines from regional to zonal availability

This article explains how to move a single-instance Azure virtual machine (VM) from a regional deployment to a zonal deployment in the same Azure region.

Prerequisites

Before you begin, verify the following:

- **Availability Zone support**: Confirm that your target region supports Availability Zones. Learn more.
- **VM SKU availability**: VM size (SKU) availability can vary by region and zone. Check SKU availability before you start. Learn more.
- **Subscription permissions**: Verify that you have *Owner* access to the subscription that contains the VMs you want to move. The first time you move a VM to a zonal deployment, Azure requires a trusted system-assigned managed identity. The account you use must have Owner permissions so Azure can create the identity and assign the required role (Contributor or User Access Administrator) in the source subscription. Learn more.
- **VM support**: Confirm that the VM configuration is supported for this move workflow. Learn more.
- **Subscription quota**: Ensure the subscription has enough quota to create the target zonal VM and related networking resources. If needed, request additional limits.
- **Regional or zonal capacity**: Quota approval doesn't guarantee that capacity is available in the target zone. If the move validation reports capacity constraints, select a different target zone or VM size.
- **VM health**: Ensure the VM is healthy and that required reboots or mandatory updates are complete.

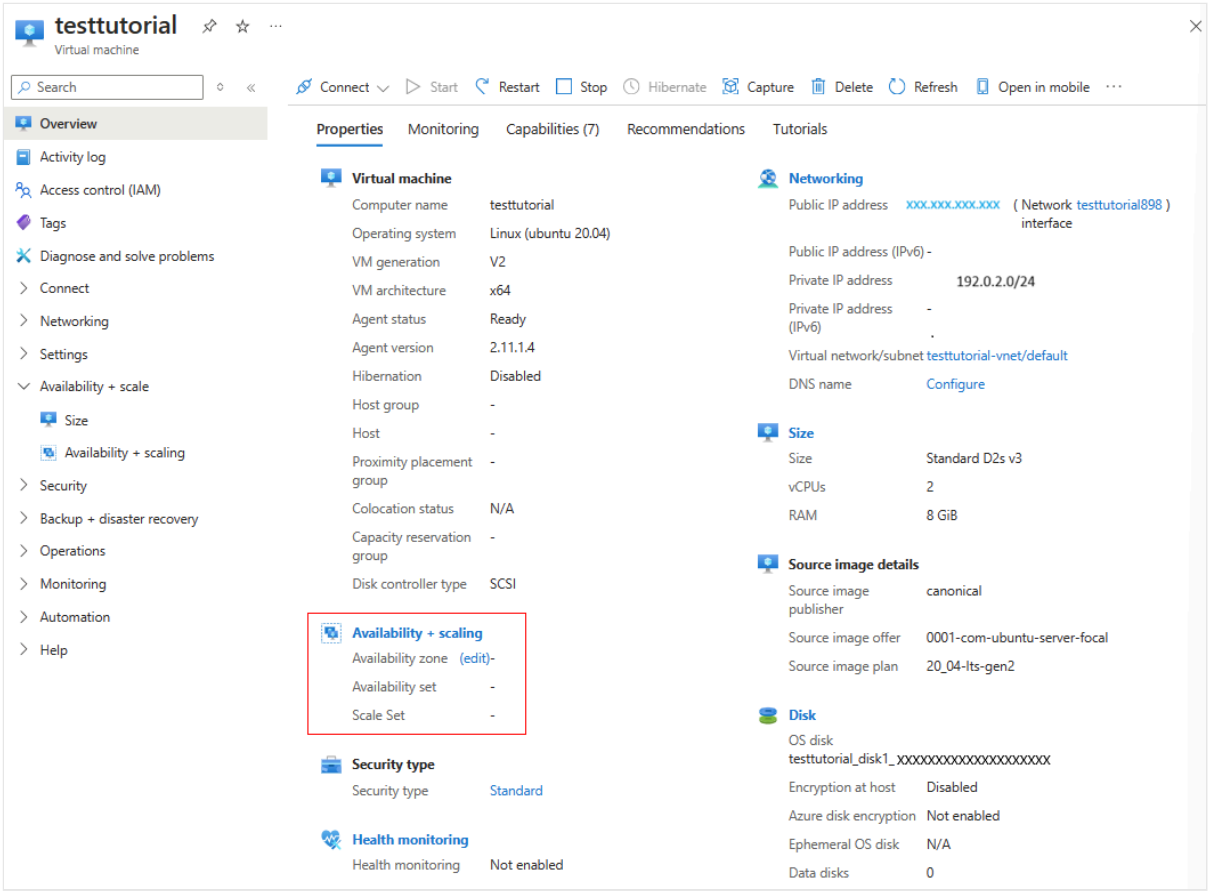
Select and move VMs

Choose a tool to move a VM from regional to zonal deployment in the same region.

Portal

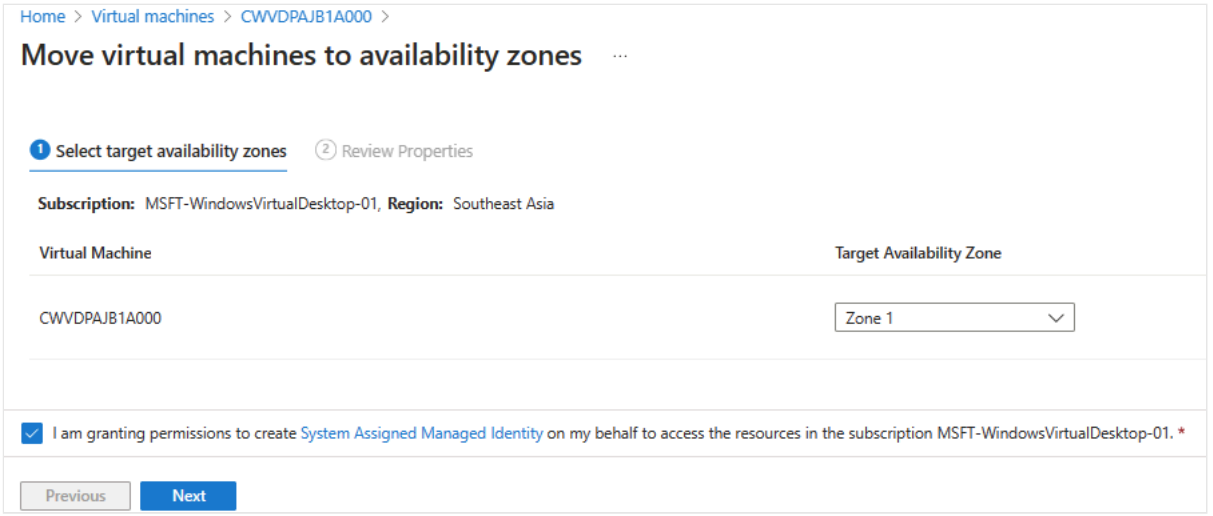
Portal: Select the VM

1. In the Azure portal, go to your VM.
2. In the VM overview, select **Availability + scaling** > **Edit**.



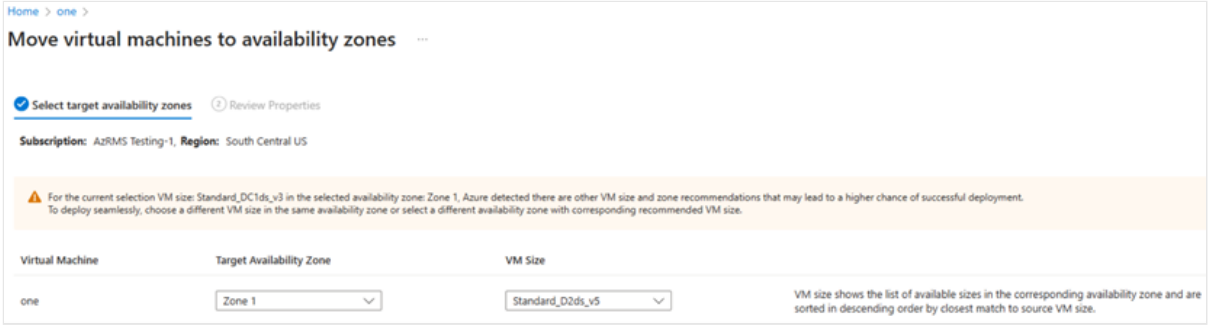
Portal: Select the target availability zone

1. Under **Target availability zone**, select the zone you want to use (for example, Zone 1).



If the selected VM configuration isn't supported for move, validation fails and you must restart with a supported VM configuration. See the support matrix.

2. If Azure recommends a different VM size to improve deployment success in the selected zone, choose the recommended size, or choose a different zone and keep the current size.



3. Select the consent statement for the **System Assigned Managed Identity** process, then select **Next**.

The managed identity setup can take a few minutes. Progress updates are shown in the portal.

Portal: Review VM properties before move

On **Review properties**, review the target VM configuration.

The following source VM properties are retained by default in the target zonal VM:

Expand table

Property	Description
VM name	The source VM name is retained in the target zonal VM.

Property	Description
Virtual network/subnet	The source virtual network (VNET) and subnet are retained, and the target zonal VM is created in the same VNET and subnet by default. You can also select or create a different VNET and subnet.
Network Security Group (NSG)	The source NSG is retained by default. You can select or create a different NSG.
Load balancer (Standard SKU)	Standard SKU load balancers support zonal deployment and are retained.
Public IP (Standard SKU)	Standard SKU public IP addresses support zonal deployment and are retained.

The following properties are newly created by default in the target zonal VM as needed:

Expand table

Property	Description
VM	A copy of the source VM is created in the target zonal configuration. The source VM remains intact and is stopped after the move. Source VM ARM ID isn't retained.
Resource group	A new resource group is created by default. The source resource group can't be reused when the target VM uses the same name in the same region. You can select an existing target resource group.
NIC	A new NIC is created for the target zonal VM. The source NIC remains intact and is stopped after move. Source NIC ARM ID isn't retained.
Disks	New disks are created in the target zonal configuration and attached to the new zonal VM.
Load balancer (Basic SKU)	Basic SKU load balancers don't support zonal deployment and aren't retained. A new Standard SKU load balancer is created by default. You can edit this or select an existing target load balancer.
Public IP (Basic SKU)	Basic SKU public IP addresses don't support zonal deployment and aren't retained. A new Standard SKU public IP is created by default. You can edit this or select an existing target public IP.

After reviewing the target configuration:

1. Resolve any validation errors.
2. Select the consent statement at the bottom of the page to show you understand that the source VM(s) will be stopped during the process.

3. Select **Move** to complete the move.

During the move:

- The source VM is **stopped**, so brief downtime occurs.
- A target zonal VM is created and started.

Configure settings after move

- Review source VM settings and reconfigure as needed (for example, extensions, RBAC, Public IP configuration, backup, and disaster recovery settings).
- After move completion, the source VM remains stopped. You can delete it or keep it for another purpose.
- After the move, you can manually delete the move collection. To remove the move collection:

    1. Enable hidden resources (the move collection is hidden by default).
    2. Go to the move collection resource group by searching for *ZonalMove-MC-RG-SourceRegion*.
    3. Delete the move collection (for example, *ZonalMove-MC-RG-UKSouth*).

Delete source regional VMs

After you commit the move and verify that the resources work as expected in the target region, you can delete each source resource using:

- Azure portal
- PowerShell
- Azure CLI

Related content

- For deployment guidance and limits, see Move to Availability Zones (Azure portal).
- See Regional to zonal move FAQ.

Last updated on 04/10/2026

Frequently asked questions - Move Azure single instance virtual machines from regional to zonal target availability zones

Article • 08/22/2024

This article answers common questions about Azure single instance virtual machines - regional to zonal move.

Regional to zonal move

Can I move virtual machine(s) in all Azure regions?

Currently, you can move virtual machine(s) across all public regions that are supported by Availability Zones. Learn more about the availability zone service and regional support.

> **Note**
>
> Azure China (China North 3) and Azure Govt (US Gov Virginia) are also supported.

Where is the metadata stored?

The service doesn't retain any customer data, and all data remains within the source virtual machine region. The following table shows the mapping between the virtual machine region and metadata region:

Expand table

Region group	Region	Metadata region
Americas	eastus2	eastus2
	eastus	eastus2
	westus2	eastus2
	southcentralus	eastus2

Region group	Region	Metadata region
	brazilsouth	brazilsouth
	canadacentral	canadacentral
	westus3	eastus2
Europe	northeurope	northeurope
	westeurope	northeurope
	uksouth	uksouth
	francecentral	francecentral
	switzerlandnorth	switzerlandnorth
	germanywestcentral	germanywestcentral
	norwayeast	norwayeast
	swedencentral	swedencentral
	polandcentral	polandcentral
	spaincentral	northeurope
	italynorth	northeurope
Middle East	uaenorth	uaenorth
	qatarcentral	qatarcentral
Asia Pacific	japaneast	japaneast
	eastasia	southeastasia
	southeastasia	southeastasia
	australiaeast	australiaeast
	centralindia	centralindia
	koreacentral	koreacentral
Africa	southafricanorth	southeastasia

Is the collected metadata encrypted?

Yes, the collected metadata is encrypted both during transit and at rest. While in transit, the metadata is securely sent to the Resource Mover service over the internet using HTTPS. The metadata is also encrypted while in storage.

What resources are supported for this Zonal Move?

Currently, managed disks are supported for virtual machines that only have a single instance.

What source resources can be used in the target zonal configuration, if preferred?

The following resources can be used in the target zonal configuration:

- Networking resources such as VNET, Subnet, and NSG can be reused.
- Public IP address (Standard SKU)
- Load Balancers (Standard SKU)

What resources are created new by default in the target zonal configuration?

The following resources are created in the target zonal configuration:

- **Resource group**: By default, a new resource group is automatically created. The source resource group can't be used, as we're using the same source virtual machine name in the target zone and two identical virtual machines can't coexist in the same resource group. However, you can still modify the properties of the new resource group or choose a different target resource group.
- **Virtual machine**: A copy of the source virtual machine is created in the target zonal configuration. The source virtual machine remains unchanged and is stopped after the transfer.
- **Disks**: The disks attached to the source virtual machine are recreated in the target zonal configuration.
- **NIC**: A new network interface card (NIC) is produced and linked to the newly created virtual machine in the designated zone.

What permissions do I need to use managed identity?

To use the managed identity service, you must have the following permissions:

- Permission to write or create resources in your subscription (which is available with the *Contributor* role).
- Permission to create role assignments (which is available with the *Owner* or *User Access Administrator* roles, or, custom roles that have the Microsoft.Authorization or role assignments or write permission assigned). This permission isn't required if the data share resource's managed identity has already been granted access to the Azure data store.

When adding resources in the portal, permissions to use managed identity are handled automatically as long as you have the appropriate role assignments.

> **Important**
>
> We recommend that you don't modify or remove identity role assignments.

What if I don't have permissions to assign role identity?

There are a couple of reasons you might not have the permissions. Consider the following scenarios:

Expand table

Scenario	Resolution
You don't have Contributor and User Access Administrator (or Owner) permissions when you add a resource for the first time.	Use an account with Contributor and User Access Administrator (or Owner) permissions for the subscription.
The Resource Mover managed identity doesn't have the necessary role.	Add the Contributor and User Access Administrator roles.

How is managed identity used?

Managed identity previously known as Managed Service Identity (MSI), is a feature that provides Azure services with an automatically managed identity in Microsoft Entra ID. This identity is used to access Azure subscriptions and perform various tasks, such as moving resources to Availability Zones.

- Managed identity is used so that you can access Azure subscriptions to move resources to availability zones.
- To move resources using a move collection, you need a system-assigned identity that has access to the subscription containing the resources you want to move.

- If you're using the Azure portal to move the virtual machines, this process is automated once the user consent is provided. The process typically takes a few minutes to complete.

Can I move my resources from Regional to Zonal and across subscriptions?

You can use virtual machine Regional to Zonal Move capability to move virtual machines from a regional to a zonal deployment within the same subscription and then use Azure Resource Manager to move them across subscriptions.

Are Azure Backup/DR, RBAC, Tags, Policies, and extensions on virtual machines supported?

Only tags and user assigned managed identities are replicated to the target zones. RBAC, policies and extensions must be reconfigured after the move. See the support matrix for further details.

Is customer data stored during the move?

Customer data isn't stored during the move. The system only stores metadata information that helps track and monitor the progress of the resources being moved.

What happens to the source virtual machine(s)?

When you select **Move**, the following steps are performed on the source virtual machines:

1. The source virtual machines are stopped and left intact in their original configuration.

   > ⓘ **Note**
   >
   > Stopping the VMs could lead to a brief downtime.

2. Virtual machine restore points of the source virtual machine are taken. These restore points contain a disk restore point for each of the attached disks and a disk restore point consists of a snapshot of an individual managed disk.
3. Using these restore points, a new virtual machine with its associated disks (a copy of the source VM) is created in the zonal configuration.
4. After the move is complete, you can choose to delete the source virtual machines.

Is there any cost associated as part of this move?

The Zonal Move feature of virtual machines is offered free of cost, but you may incur cost of goods for the creation of disk snapshots or restore points.

> **Note**
>
> The snapshot of virtual machine or disks is automatically deleted after the move is complete.

Can I retain my Public IP of the source virtual machine?

Review the following scenarios where you can or can't retain Public IP addresses associated with the source virtual machine.

Expand table

Source Property	Description
Public IP addresses (Basic SKU) attached to source virtual machine NIC	The source public IP address isn't retained. The source public IP SKU doesn't support target zonal configuration. By default, a copy of the source virtual machine and a new network interface card (NIC) is created. The source virtual machine and NIC are left intact after the move and the source virtual machine will be in a shutdown state.
Public IP addresses (Standard SKU) attached to source virtual machine NIC	The source Public IP address isn't retained. A new NIC and a copy of the source virtual machine (VM) is created, and both the source virtual machine and NIC will remain intact after the move. However, the virtual machine will be in a shutdown state. **Note**: After the move, if you wish, you can separate the source public IP from the source NIC and connect it to a new target zonal virtual machine NIC.
Public IP address (Basic SKU) attached to Load Balancer (Basic SKU)	The source public IP address isn't retained. Source Public IP SKU doesn't support target zonal configuration.
Public IP address (Standard SKU) with Non-Zonal configuration	Source Public IP address is retained.

Source Property	Description
attached to Load Balancer (Standard SKU)	
Public IP address (Standard SKU) with Zone pined configuration attached to Load Balancer (Standard SKU)	Source Public IP address will be retained. **Note:** The target virtual machine zone# might not be the same as the zone pinned Public IP.
Public IP address (Standard SKU) with Zone redundant configuration attached to Load Balancer (Standard SKU)	Source Public IP address is retained.

Next steps

- Learn more about moving single instance Azure VMs from regional to zonal configuration.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What are Virtual Machine Scale Sets?

Azure Virtual Machine Scale Sets let you create and manage a group of load balanced virtual machines (VM) instances. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide the following key benefits:

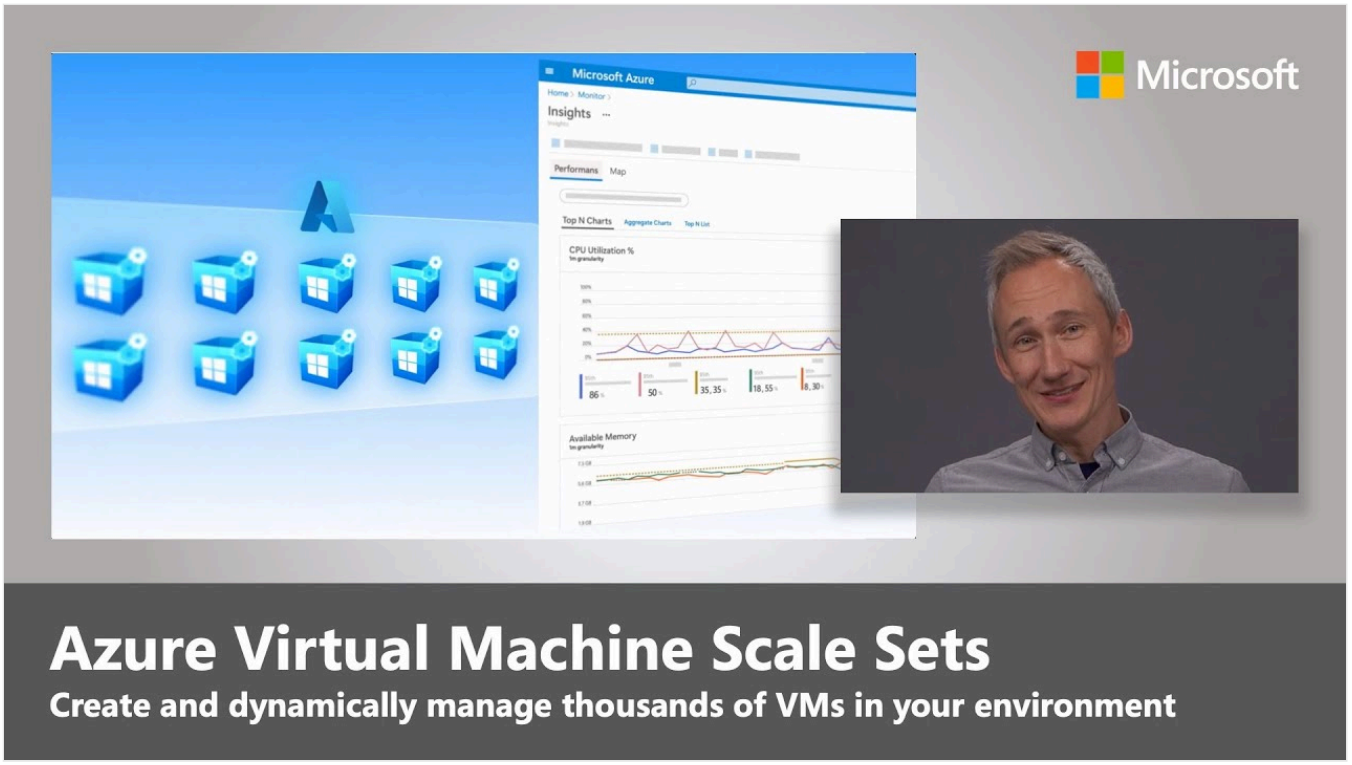
- Easy to create and manage multiple VMs
- Provides high availability and application resiliency by distributing VMs across availability zones or fault domains
- Allows your application to automatically scale as resource demand changes
- Works at large-scale

With Flexible orchestration, Azure provides a unified experience across the Azure VM ecosystem. Flexible orchestration offers high availability guarantees (up to 1,000 VMs) by spreading VMs across fault domains in a region or within an Availability Zone. This enables you to scale out your application while maintaining fault domain isolation that is essential to run workloads, including:

- Quorum-based workloads
- Open-source databases
- Stateful applications
- Services that require high availability and large scale
- Services that want to mix virtual machine types or use Spot and on-demand VMs together
- Existing Availability Set applications

Learn more about the differences between Uniform scale sets and Flexible scale sets in Orchestration Modes.

> ⓘ **Important**
>
> The orchestration mode is defined when you create the scale set and cannot be changed or updated later.



Why use Virtual Machine Scale Sets?

To provide redundancy and improved performance, applications are typically distributed across multiple instances. Customers may access your application through a load balancer that distributes requests to one of the application instances. If you need to perform maintenance or update an application instance, your customers must be distributed to another available application instance. To keep up with extra customer demand, you may need to increase the number of application instances that run your application.

Azure Virtual Machine Scale Sets provide the management capabilities for applications that run across many VMs, automatic scaling of resources, and load balancing of traffic. Scale sets provide the following key benefits:

- **Easy to create and manage multiple VMs**
  - When you have many VMs that run your application, it's important to maintain a consistent configuration across your environment. For reliable performance of your application, the VM size, disk configuration, and application installs should match across all VMs.
  - With scale sets, all VM instances are created from the same base OS image and configuration. This approach lets you easily manage hundreds of VMs without extra configuration tasks or network management.
  - Scale sets support the use of the Azure load balancer for basic layer-4 traffic distribution, and Azure Application Gateway for more advanced layer-7 traffic distribution and TLS termination.

- **Provides high availability and application resiliency**
  - Scale sets are used to run multiple instances of your application. If one of these VM instances has a problem, customers continue to access your application through one of the other VM instances with minimal interruption.
  - For more availability, you can use Availability Zones to automatically distribute VM instances in a scale set within a single datacenter or across multiple datacenters. A scale set alone can't protect you against data center failures. Deploying VMs across Availability Zones within a scale set can protect you against data center failure.

- **Allows your application to automatically scale as resource demand changes**
  - Customer demand for your application may change throughout the day or week. To match customer demand, scale sets can automatically increase the number of VM instances as application demand increases, then reduce the number of VM instances as demand decreases.
  - Autoscale helps reduce the number of unnecessary VMs when demand is low. As demand increases, the scale set automatically adds more VMs to maintain an acceptable level of performance for your application. This ability helps reduce costs and efficiently create Azure resources as required.

- **Works at large-scale**
  - Scale sets support up to 1,000 VM instances for standard marketplace images and custom images through the Azure Compute Gallery (formerly known as Shared Image Gallery). If you create a scale set using a managed image, the limit is 600 VM instances.
  - For the best performance with production workloads, use Azure managed disks.

- **Cost-effective service**
  - There's no extra cost for using scale sets. You're charged based on the compute, network, and storage resources that the scale set uses.
  - For virtual machine pricing information, see Azure pricing.

Next steps

Flexible orchestration mode for your scale sets with Azure portal.

Last updated on 03/21/2025

Scaling profile

Article • 01/10/2025

Virtual machine scale sets allow you to define a scaling profile or template, which specifies the properties of virtual machine instances. Examples of properties set in the virtual machine scaling profile include:

- VM Image
- Admin credentials
- Network interface settings
- Load balancer backend pool
- OS and Data disk settings

When you increase the capacity or instance count of the scale set, the scale set will add new virtual machines to the set based on the configuration defined in the profile. Scale sets with scaling profile are also eligible for orchestrations such as reimaging, rolling upgrades, instance repair, and automatic OS updates.

> **Important**
>
> Virtual machine scaling profile settings are required for scale sets in Uniform Orchestration Mode, and optional for scale sets in Flexible Orchestration Mode.

Create a scale set with a scaling profile

By default, scale sets are created with a virtual machine scaling profile. See Create an Azure Virtual Machine and Create and manage Azure Virtual Machines for examples.

Create a scale set without a scaling profile

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Virtual machine scale sets in Flexible Orchestration Mode can optionally be created without a virtual machine scaling profile. This configuration is similar to creating and deploying an Availability Set in that you add to the set by manually creating virtual machine instances and adding them to the set. It's useful to create a scale set without a scaling profile when:

- You need complete control over all VM properties.
- You need to follow your own VM naming conventions.
- You want to add different types of VMs to the same scale set.
- You need to control the placement of virtual machines into a specific availability zone or fault domain.

Expand table

Feature	Virtual machine scale sets (no scaling profile)	Availability Sets
Maximum capacity	1000	200
Supports Availability Zones	Yes	No
Maximum Aligned Fault Domains Count	3	3
Add new VM to set	Yes	Yes
Add VM to specific fault domain	Yes	No
Maximum Update Domain count	N/A. Update domains are deprecated	20

Once you have created the virtual machine scale set, you can manually attach virtual machines.

Azure portal

1. Navigate to the **Create a Virtual Machine Scale Set (VMSS)** blade.
2. Set your resource group, scale set name, region, availability zone, and orchestration mode.
3. In the **Scaling** section, select **No scaling profile**.
4. Complete the **Advanced** and **Tags** section before selecting the **Review + create** button.

Next steps

- Create a new Virtual Machine Scale Set with a scale profile using the Azure portal.
- If you have created a virtual machine scale set without a scaling profile, you can manually attach a virtual machine.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Attach or detach a Virtual Machine to or from a Virtual Machine Scale Set

Article • 04/23/2025

Attaching a Virtual Machine to a Virtual Machine Scale Set

> **Important**
>
> You can only attach Virtual Machines to a Virtual Machine Scale Set in **Flexible orchestration mode**. For more information, see Orchestration modes for Virtual Machine Scale Sets.

There are times where you need to attach a virtual machine (VM) to a Virtual Machine Scale Set to benefit from the scale, availability, and flexibility that comes with scale sets. There are two ways to attach VMs to scale sets: manually create a new standalone VM in the scale set or attach an existing VM to the scale set.

You can attach a new standalone VM to a scale set when you need a different configuration on a specific VM than what's defined in the scaling profile, or when the scale set doesn't have a scaling profile. Manually attaching VMs gives you full control over instance naming and placement into a specific availability zone or fault domain. The VM doesn't have to match the configuration in the scale set's scaling profile, so you can specify parameters like operating system, networking configuration, on-demand or Spot, and VM size.

You can attach an existing VM to an existing Virtual Machine Scale Set by specifying which scale set you would like to attach to. The VM doesn't have to be the same as the VMs already running in the scale set, meaning it can have a different operating system, network configuration, priority, disk, and more.

Attach a new Virtual Machine to a Virtual Machine Scale Set

Attach a virtual machine to a Virtual Machine Scale Set at the time of VM creation by specifying the `virtualMachineScaleSet` property.

> **Note**
>
> Attaching a virtual machine to Virtual Machine Scale Set doesn't by itself update any VM networking parameters such as load balancers. If you would like this virtual machine to receive traffic from any load balancer, you must manually configure the VM network interface to receive traffic from the load balancer. Learn more about Load balancers.

Azure portal

1. Go to **Virtual Machines**.
2. Select **Create**.
3. Select **Azure virtual machine**.
4. In the **Basics** tab, open the **Availability options** dropdown and select **Virtual Machine Scale Set**.
5. In the **Virtual Machine Scale Set** dropdown, select the scale set to which you want to add this virtual machine.
6. Optionally, specify the Availability zone or Fault domain to place the VM.

Exceptions to attaching a new Virtual Machine to a Virtual Machine Scale Set

- The VM must be in the same resource group as the scale set.
- Regional virtual machines (no availability zones specified) can be attached to regional scale sets.
- Zonal virtual machines can be attached to scale sets that specify one or more zone. The virtual machine must be in one of the zones spanned by the scale set. For example, you can't create a virtual machine in Zone 1, and place it in a scale set that spans Zones 2 and 3.
- The scale set must be in Flexible orchestration mode, and the `singlePlacementGroup` property must be `false`.

Attach an existing Virtual Machine to a Virtual Machine Scale Set

Attach an existing virtual machine to a Virtual Machine Scale Set after the time of VM creation by specifying the `virtualMachineScaleSet` property. Attaching an existing VM to a scale set with a fault domain count of 1 doesn't require downtime.

> **Note**
>
> Attaching a virtual machine to Virtual Machine Scale Set doesn't by itself update any VM networking parameters such as load balancers. If you would like this virtual machine to receive traffic from any load balancer, you must manually configure the VM network interface to receive traffic from the load balancer. Learn more about Load balancers.

Azure portal

1. Go to **Virtual Machines**.
2. Select the name of the virtual machine you'd like to attach to your scale set.
3. Under **Settings** select **Availability + scaling**.

4. In the **Scaling** section, select the **Get started** button. If the button is grayed out, your VM currently doesn't meet the requirements to be attached to a scale set.
5. In the **Attach to a VMSS** blade on the right side of the page, select the scale set you'd like to attach the VM to in the **Select a VMSS** dropdown.
6. Select the **Attach** button at the bottom to attach the VM.

Limitations for attaching an existing Virtual Machine to a scale set

- The scale set must use Flexible orchestration mode.
- The scale set must have a `platformFaultDomainCount` of **1**.
- The VM and scale set must be in the same resource group.
- The VM and target scale set must both be zonal, or they must both be regional. You can't attach a zonal VM to a regional scale set.
- The VM can't be in a self-defined availability set.
- The VM can't be in a `ProximityPlacementGroup`.
- The VM can't be in an Azure Dedicated Host.
- The VM must have a managed disk.
- The scale set must have `singlePlacementGroup` set to `False`.
- Scale sets created without a scaling profile default to `singlePlacementGroup` set to `null`. To attach VMs to a scale set without a scaling profile, `singlePlacementGroup` needs to be set to `False` at the time of the scale set's creation.
- The VM can't be a Remote Direct Memory Access (RDMA) capable HB-series or N-series VM (Preview feature).

Detaching a Virtual Machine from a Virtual Machine Scale Set

Should you need to detach a VM from a scale set, you can follow the below steps to remove the VM from the scale set.

Azure portal

1. Go to **Virtual Machines**.
2. Select the name of the virtual machine you'd like to attach to your scale set.
3. Under **Settings** select **Availability + scaling**.
4. Select the **Detach from the VMSS** button at the top of the page.
5. When prompted to confirm, select the **Detach** button.
6. Portal sends a notification when the VM is detached.

Limitations for detaching a Virtual Machine from a scale set

- The scale set must use Flexible orchestration mode.
- The scale set must have a `platformFaultDomainCount` of **1**.
- Scale sets created without a scaling profile default to `singlePlacementGroup` set to `null`. To detach VMs from a scale set without a scaling profile, `singlePlacementGroup` needs to be set to `False`.
- The VM can't be an RDMA capable HB-series or N-series VM (Preview feature).

Moving Virtual Machines between scale sets

To move a VM from one scale set to another, use the following steps:

1. Detach the VM from scale set A.
2. Once the detach completes, attach the VM to scale set B.

Limitations

The limitations for VMs to be attached or detached to or from a scale set remain the same.

Troubleshooting

Attach an existing Virtual Machine to an existing scale set troubleshooting

Expand table

Error Message	Description	Troubleshooting Options
The Virtual Machine Scale Set '{vmssUri}' referenced by the Virtual Machine does not exist.	The scale set resource doesn't exist, or isn't in Flexible Orchestration Mode.	Check to see if the scale set exists. If it does, check if it's using Uniform Orchestration Mode.
This operation is not allowed because referenced Virtual Machine Scale Set '{vmssName}' does not have orchestration mode set to 'Flexible'.	The scale set isn't in Flexible Orchestration Mode.	Try attaching to another scale set with Flexible Orchestration Mode enabled.
Referenced Virtual Machine '{vmName}' belongs to an Availability Set and attaching to a Virtual Machine Scale Set is not	`VmssDoesNotSupportAttachingExistingAvsetVM`: The VM that you attempted to attach is part of an Availability Set and can't be attached to a scale set.	VMs in an Availability Set can't be attached to a scale set.

Error Message	Description	Troubleshooting Options
supported. For more information, see https://aka.ms/vmo/attachdetach .		
Referenced Virtual Machine Scale Set '{vmssName}' does not support attaching an existing Virtual Machine to it because the Virtual Machine Scale Set has more than 1 fault domains. For more information, see https://aka.ms/vmo/attachdetach .	`VmssDoesNotSupportAttachingExistingVMMultiFD`: The operation to attach the VM failed because the VM was trying to attach to a scale set with a platform fault domain count of more than one.	VMs can only be attached to scale sets with a `platform fault domain count` of 1. Try attaching to a scale set with a platform fault domain count of one.
Using a Virtual Machine '{vmName}' with unmanaged disks and attaching it to a Virtual Machine Scale Set is not supported. For more information, see https://aka.ms/vmo/attachdetach .	`VmssDoesNotSupportAttachingExistingVMUnmanagedDisk`: VMs with unmanaged disks can't be attached to a scale set.	To attach a VM with a disk to the scale set, ensure that the VM is using a managed disk. Visit the documentation to learn how to migrate from an unmanaged disk to a managed disk.
Referenced Virtual Machine '{vmName}' belongs to a proximity placement group (PPG) and attaching to a Virtual Machine Scale Set is not supported. For more information, see https://aka.ms/vmo/attachdetach .	`VmssDoesNotSupportAttachingPPGVM`: The operation to attach the VM failed because the VM is part of a Proximity Placement Group.	VMs from a Proximity Placement Group can't be attached to a scale set. Remove the VM from the Proximity Placement Group and then try to attach to the scale set. See the documentation to learn about how to move a VM out of a Proximity Placement Group.
PropertyChangeNotAllowed Changing property virtualMachineScaleSet.id isn't allowed.	The Virtual Machine Scale Set ID can't be changed to a different Virtual Machine Scale Set ID without detaching the VM from the scale set first.	Detach the VM from the Virtual Machine Scale Set, and then attach to the new scale set.
Virtual Machine Scale Set '{0}' does not support attaching an existing Virtual Machine to it because the Virtual Machine Scale Set has single placement group set to true or does not have single placement	`VmssDoesNotSupportAttachingWithSpg`: The operation for attaching the VM failed because the scale set is part of a Single Placement Group.	VMs can only be attached to scale sets with `singlePlacementGroup` set to `false`.

Error Message	Description	Troubleshooting Options
group explicitly set to false. Please see https://aka.ms/vmo/attachdetach for more information.		
Virtual Machine Scale Set does not support attaching Virtual Machine {0} because it uses VM Size {1} which can be only be used with a single placement group enabled Virtual Machine Scale Set. Please see https://aka.ms/vmo/attachdetach for more information.	The VM being attached is of a size that requires the scale set to use a Single Placement Group.	VMs requiring Single Placement Group can't be attached to a scale set.
Virtual Machine Scale Set does not support attaching RDMA capable VM Sizes such as {0}. Please see https://aka.ms/vmo/attachdetach for more information.	RDMA capable VMs can't be detached from the scale set. The detach failed because the VM is RDMA capable.	Only VMs that aren't RDMA enabled can be detached from the scale set. This functionality is in Preview.

Detach a Virtual Machine from a scale set troubleshooting

Expand table

Error Message	Description	Troubleshooting options
The Virtual Machine Scale Set '{vmssUri}' referenced by the Virtual Machine does not exist.	The scale set resource doesn't exist, or isn't in Flexible Orchestration Mode.	Check to see if the scale set exists. If it does, check if it's using Uniform Orchestration Mode.
This operation is not allowed because referenced Virtual Machine Scale Set '{vmssName}' does not have orchestration mode set to 'Flexible'.	The scale set isn't in Flexible Orchestration Mode.	Only scale sets with Flexible Orchestration Mode can have VMs detached from them.
Virtual Machine Scale Set '{vmssName}' does not support detaching an existing Virtual Machine from it because the Virtual Machine Scale Set has more than 1 fault domains. For more information, see https://aka.ms/vmo/attachdetach .	The detach of the VM failed because the scale set it's in has more than one platform fault domain.	VMs can only be detached from scale sets with a `platform fault domain count` of one.
OperationNotAllowed, Message: This operation is not allowed because referenced Virtual Machine Scale Set	The scale set you attempted to attach to or detach from is a scale set with Uniform Orchestration Mode.	Only scale sets with Flexible Orchestration Mode can have VMs detached from them.

Error Message	Description	Troubleshooting options
'{armId}' does not have orchestration mode set to 'Flexible'		
PropertyChangeNotAllowed Changing property virtualMachineScaleSet.id is not allowed.	The Virtual Machine Scale Set ID can't be changed to a different Virtual Machine Scale Set ID without detaching the VM from the scale set first.	Detach the VM from the Virtual Machine Scale Set, and then attach to the new scale set. Ensure the `virtualMachineScaleSet.id` is set to the value of `null`. Incorrect values include: `""` and `"null"`.
Virtual Machine Scale Set '{0}' does not support detaching Virtual Machine from it because the Virtual Machine Scale Set has single placement group set to true. Please see https://aka.ms/vmo/attachdetach for more information.	`VmssDoesNotSupportAttachingWithSpg`: The detach of the VM failed because the scale set is part of a Single Placement Group.	VMs can only be detached from scale sets with `singlePlacementGroup` set to `false`.
Virtual Machine Scale Set does not support detaching RDMA capable VM Sizes such as {0}. Please see https://aka.ms/vmo/attachdetach for more information.	RDMA capable VMs can't be detached from the scale set. The detach failed because the VM is RDMA capable.	Only VMs that aren't RDMA enabled can be detached from the scale set. This functionality is in Preview.

What's next

Learn how to manage updates and maintenance using Maintenance notification, Maintenance configurations, and Scheduled Events.

Standby pools for Virtual Machine Scale Sets

Article • 05/12/2025

> **Important**
>
> For standby pools to successfully create and manage resources, it requires access to the associated resources in your subscription. Ensure the correct permissions are assigned to the standby pool resource provider in order for your standby pool to function properly. For detailed instructions, see configure role permissions for standby pools.

Standby pools for Virtual Machine Scale Sets enables you to increase scaling performance by creating a pool of pre-provisioned virtual machines. The virtual machines in the standby pool complete all post provisioning processes such as installing applications, downloading data packages, etc. Once the virtual machines are fully provisioned, they are be maintained in a running, stopped (deallocated), or hibernated (preview) state. When your scale set requires more instances, the instances in the standby pool are automatically moved into the scale set. A standby pool significantly reduces the time it takes to scale-out a Virtual Machine Scale Set.

If maintaining a standby pool of running virtual machines, the machines are immediately ready to receive traffic after being moved into the scale set. If maintaining a standby pool of stopped (deallocated) or hibernated virtual machines, the virtual machines are automatically started or resumed after moving into the scale set.

Feature Registration

Register the standby pool resource provider with your subscription using Azure Cloud Shell. Registration can take up to 30 minutes to successfully show as registered. You can rerun the below commands to determine when the feature is successfully registered.

> **Note**
>
> To enable standby pool functionality in the Azure portal, the `StandbyPoolVMPoolPreview` feature flag is required. This is requirement is temporary and is in progress of being removed.

Azure PowerShell

```
Register-AzResourceProvider -ProviderNamespace Microsoft.StandbyPool
Register-AzProviderFeature -FeatureName StandbyVMPoolPreview -ProviderNameSpace
```

```
Microsoft.StandbyPool
```

Scaling

Moving virtual machines between the standby pool into the scale set happens automatically whenever a scale-out event is triggered. There's no extra configuration required. As long as there's an available instance in the standby pool that has completed all provisioning steps, the scale set by default uses that instance when scaling up.

When scaling back down, the instances are deleted from your scale set based on the scale-in policy and the standby pool refills to meet the max ready capacity configured. If at any point in time your scale set needs to scale beyond the number of instances you have in your standby pool, the scale set defaults to standard scale-out methods and creates new instances.

Standby pools only give out virtual machines from the pool that match the desired power state configured. For example, if your desired power state is set as stopped (deallocated), the standby pool only gives the scale set instances matching that current power state. If virtual machines are in a creating, failed or any other state than the expected state the scale set defaults to new virtual machine creation instead.

Standby pool size

There are three settings that determine how many instances are in your standby pool at any given point in time. This includes the scale set instance count, the minimum ready capacity and the maximum ready capacity.

The **scale set instance count** is how many instances are currently deployed in your scale set. This is a scale set level property that can be changed at any point in time by either scaling up or scaling down. Regardless of how you're managing the scaling rules for your scale set, the standby pool keeps track of how many instances are deployed and adjust accordingly.

The **minimum ready capacity** is a user defined parameter. By default, the minimum ready capacity for any new standby pool is zero. By setting the minimum ready capacity, it informs the standby pool that it should maintain that many instances at minimum. For example, if you have a minimum ready capacity of 5, anytime a virtual machine is moved from the pool into the scale set which reduces the minimum ready capacity to less than 5, the standby pool automatically creates another instance and begins preparing it for scale-out.

The **maximum ready capacity** is a user defined parameter. This setting tells the standby pool how many instances at most should be maintained in the pool. Maximum ready capacity is directly tied to the scale set instance count. If you have a maximum ready capacity of 20 and

you currently have 10 instances in your scale set, the pool size would equal 10. If your scale set scales down to 5, the pool size would increase to 15. This continues to dynamically adjust as the scale set increases and decreases instance count.

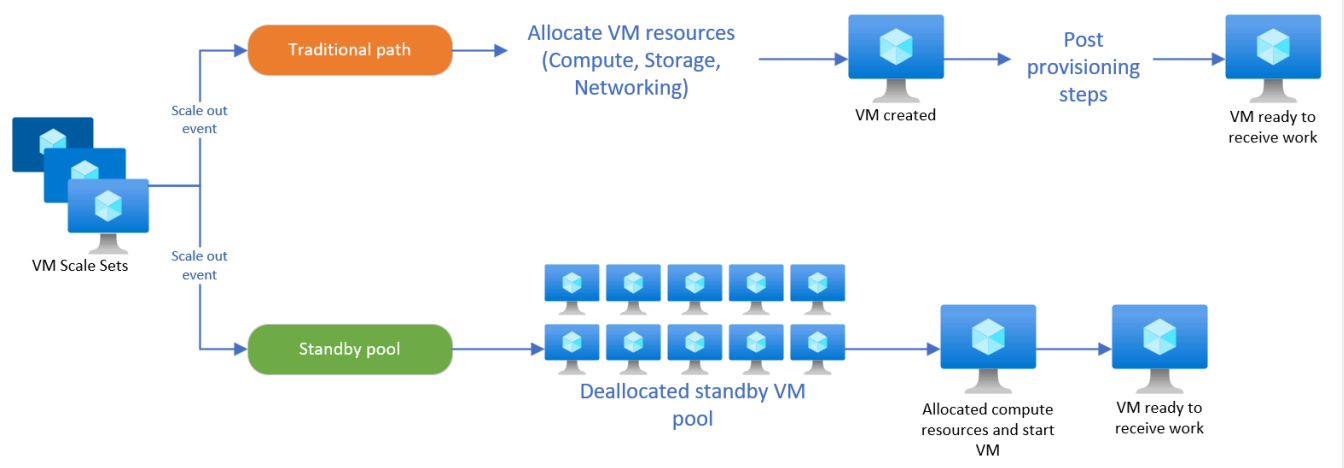
Expand table

Setting	Description
maxReadyCapacity	The maximum number of virtual machines to be created in the pool.
minReadyCapacity	The minimum number of virtual machines to be maintained in the pool.
instanceCount	The current number of virtual machines already deployed in the scale set.
Standby pool size	Standby pool size = `maxReadyCapacity` – `instanceCount`

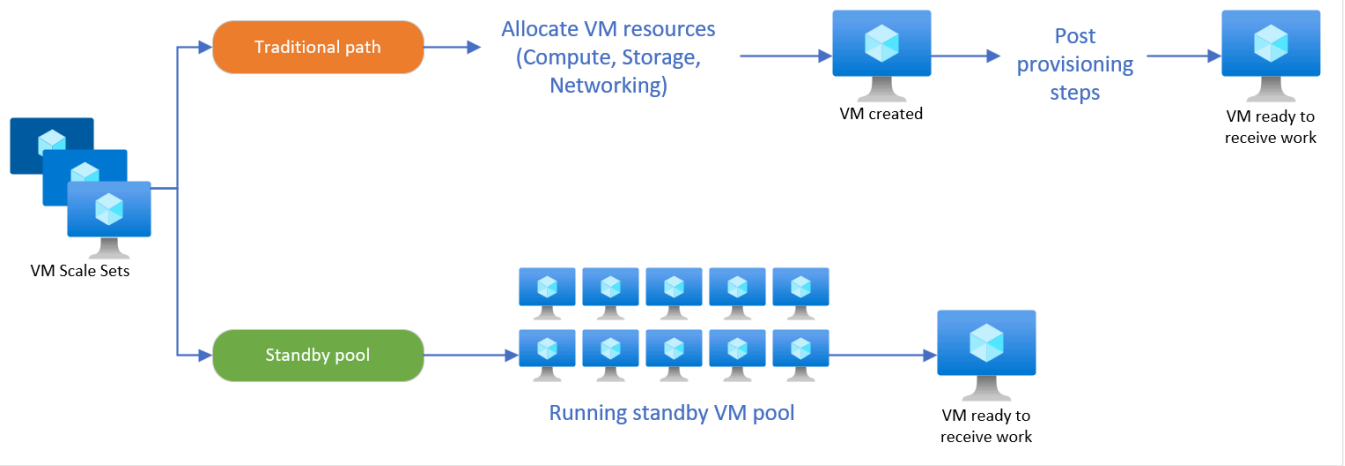
Virtual machine states

The virtual machines in the standby pool can be kept in a running, hibernated, or stopped (deallocated) state.

Deallocated: Deallocated virtual machines are shut down and keep any associated disks, network interfaces, and any static IPs. Ephemeral OS disks don't support the deallocated state.

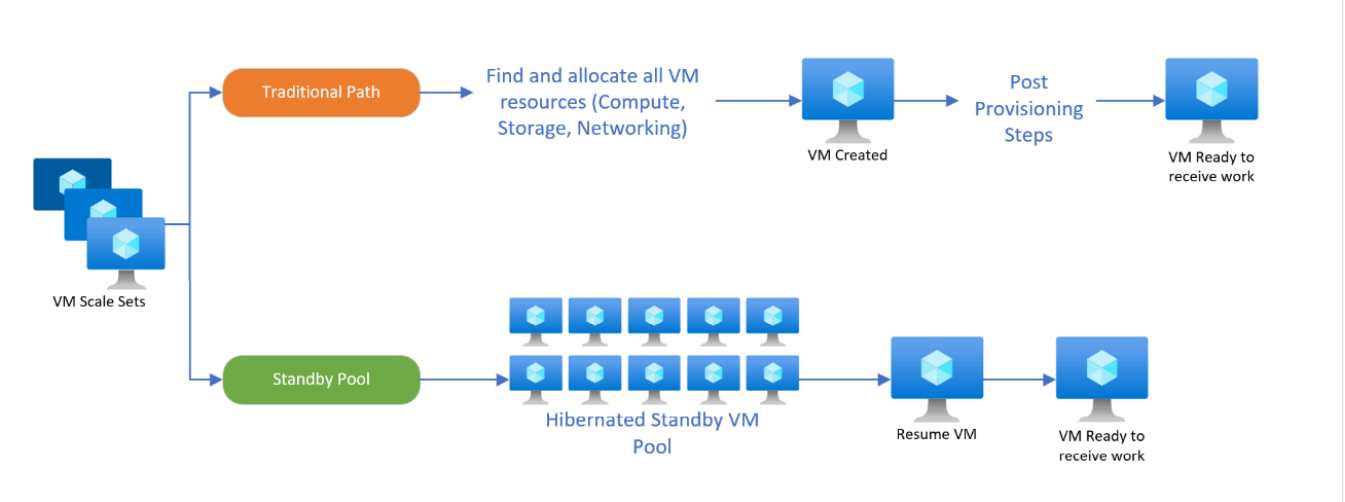


Running: Using virtual machines in a running state is recommended when latency and reliability requirements are strict. Virtual machines in a running state remain fully provisioned.



Hibernated (Preview): Using virtual machines in a hibernated state is recommended for systems with long boot times due to memory-intensive applications. These applications can be initialized on VMs and hibernated. These hibernated VMs can then be quickly started when needed, with the applications already up and running in the desired state.

> **Note**
>
> To use the hibernated VM state in your standby pool, the instances must be fully eligible and configured to support hibernation. For more information, see the Azure VM hibernation documentation.



Availability zones

When using standby pools with a Virtual Machine Scale Set spanning availability zones, the instances in the pool are spread across the same zones the Virtual Machine Scale Set is using.

When a scale-out is triggered in one of the zones, a virtual machine in the pool in that same zone is used. If a virtual machine is needed in a zone where you no longer have any pooled virtual machines left, the scale set creates a new virtual machine directly in the scale set.

Pricing

Users are charged based on the resources deployed in the standby pool. For example, virtual machines in a running state incur compute, networking, and storage costs. Virtual machines in a deallocated state doesn't incur any compute costs, but any persistent disks or networking configurations continue incur cost. Thus, a pool of running virtual machines incur more cost than a pool of deallocated virtual machines. For more information on virtual machine billing, see states and billing status of Azure Virtual Machines.

Limitations and considerations

When using standby pools with Virtual Machine Scale Sets, there are specific limitations and considerations to keep in mind:

- **Azure Autoscale**: Standby pools can't be created or attached to a Virtual Machine Scale Set with Azure autoscale enabled. Ensure autoscale is disabled before configuring a standby pool.
- **Fault Domains**: Standby pools aren't supported with Virtual Machine Scale Sets configured with a fault domain count greater than 1.
- **Region and Subscription**: The standby pool must be in the same region and subscription as the Virtual Machine Scale Set it's attached to. Cross-region or cross-subscription configurations aren't supported.
- **Single Standby Pool**: A Virtual Machine Scale Set can only have one standby pool attached at a time. Attaching multiple standby pools to the same scale set is not supported.
- **Orchestration Mode**: Standby pools are only supported with Virtual Machine Scale Sets using Flexible Orchestration. Uniform Orchestration is not supported.

Next steps

Learn how to create a standby pool.

Proximity placement groups

Article • 08/22/2024

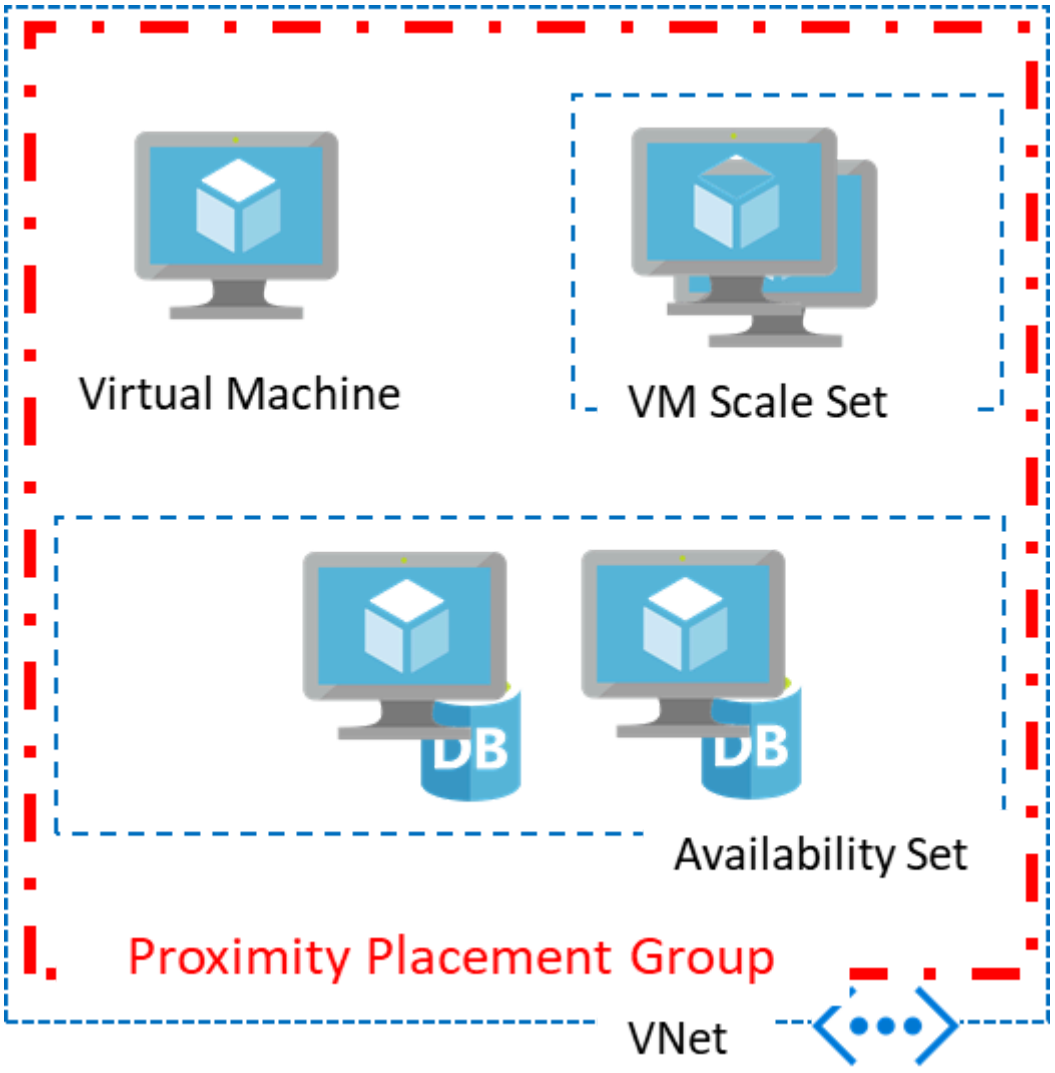
Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Placing VMs in a single region reduces the physical distance between the instances. Placing them within a single availability zone will also bring them physically closer together. However, as the Azure footprint grows, a single availability zone may span multiple physical data centers, which may result in a network latency impacting your application.

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a proximity placement group.

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

- Low latency between stand-alone VMs.
- Low Latency between VMs in a single availability set or a virtual machine scale set.
- Low latency between stand-alone VMs, VMs in multiple Availability Sets, or multiple scale sets. You can have multiple compute resources in a single placement group to bring together a multi-tiered application.
- Low latency between multiple application tiers using different hardware types. For example, running the backend using M-series in an availability set and the front end on a D-series instance, in a scale set, in a single proximity placement group.



Using Proximity Placement Groups

A proximity placement group is a resource in Azure. You need to create one before using it with other resources. Once created, it could be used with virtual machines, availability sets, or virtual machine scale sets. You specify a proximity placement group when creating compute resources providing the proximity placement group ID.

You can also move an existing resource into a proximity placement group. When moving a resource into a proximity placement group, you should stop (deallocate) the asset first since it will be redeployed potentially into a different data center in the region to satisfy the colocation constraint.

In the case of availability sets and virtual machine scale sets, you should set the proximity placement group at the resource level rather than the individual virtual machines.

A proximity placement group is a colocation constraint rather than a pinning mechanism. It's pinned to a specific data center with the deployment of the first resource to use it. Once all resources using the proximity placement group have been

stopped (deallocated) or deleted, it's no longer pinned. Therefore, whenever you use a proximity placement group with multiple VM series, it's important to specify all the required types upfront in a template if possible or follow a deployment sequence, which will improve your chances for a successful deployment. If your deployment fails, restart the deployment with the VM size, which has failed as the first size to be deployed.

Use intent to specify VM sizes

You can use the optional `intent` parameter to provide the intended VM Sizes to be part of the proximity placement group. This parameter can be specified at the time of creating a proximity placement group or it can be added/modified while updating a proximity placement group after deallocating all of the VMs.

When specifying `intent`, you can also add the optional `zone` parameter to specify an availability zone, indicating that the proximity placement group must be created within a specific availability zone. Note the following points when providing the `zone` parameter:

- The availability zone parameter can only be provided during the creation of the proximity placement group and can't be modified later.
- The `zone` parameter can only be use with `intent`, it can't be used alone.
- Only one availability zone can be specified.

Proximity Placement Group creation or update will succeed only when at least one data center supports all the VM Sizes specified in the intent. Otherwise, the creation or update will fail with "OverconstrainedAllocationRequest", indicating that the combination of VM Sizes can't be supported within a proximity placement group. The **intent does not provide any capacity reservation or guarantee**. The VM Sizes and zone given in `intent` are used to select an appropriate data center, reducing the chances of failure if the desired VM size isn't available in a data center. Allocation failures can still occur if there is no more capacity for a VM size at the time of deployment.

> **Note**
>
> To use intent for your proximity placement groups, ensure that the API version is 2021-11-01 or higher

Best Practices while using intent

- Provide an availability zone for your proximity placement group only when you provide an intent. Providing an availability zone without an intent will result in an error when creating the proximity placement group.
- If you provide an availability zone in the intent, ensure that the availability zone of the VMs you deploy match with what that specified in the intent, to avoid errors while deploying VMs.
- Creating or adding VMs with sizes that are not included in the intent is allowed, but not recommended. The size may not exist in the selected datacenter and can result in failures at the time of VM deployment.
- For existing placement groups, we recommend you include the sizes of the existing VMs when updating the intent, in order to avoid failure when redeploying the VMs.

Intent can be affected with decommissioning

- It is possible that after creating a proximity placement group with intent and before deploying VMs, planned maintenance events such as hardware decommissioning at an Azure datacenter could occur, resulting in the combination of VM Sizes specified in the intent not being available in the data center. In such cases, an "OverconstrainedAllocationRequest" error will occur, even while deploying VMs of sizes specified in the intent. You can try deallocating all the resources in the proximity placement group and recreate them to get a data center that can accommodate the intent. If there is no datacenter with the specified VM Sizes after the decommissioning, you may have to modify the intent to use a different combination of VM Sizes, since the combination of VM sizes is no longer supported.
- Azure may retire an entire VM family or a specific set of VM sizes. If you have such a VM size in the intent, you may have to either remove it or replace it with a different size before the retirement date for the original VM size. Otherwise, the intent will no longer be valid.

What to expect when using Proximity Placement Groups

Proximity placement groups offer colocation in the same data center. However, because proximity placement groups represent an additional deployment constraint, allocation failures can occur. There are few use cases where you may see allocation failures when using proximity placement groups:

- When you ask for the first virtual machine in the proximity placement group, the data center is automatically selected. In some cases, a second request for a

different VM size, may fail if it doesn't exist in that data center. In this case, an **OverconstrainedAllocationRequest** error is returned. To avoid this error, try changing the order in which you deploy your VM sizes or have both resources deployed using a single ARM template.

- If the proximity placement group is created with intent, the VMs are not required to be deployed in any particular order and are not required to be batched using a single ARM template, since the intent is used to select a datacenter that supports all VM sizes indicated in the intent.
- In the case of elastic workloads, where you add and remove VM instances, having a proximity placement group constraint on your deployment may result in a failure to satisfy the request resulting in **AllocationFailure** error.
- Stopping (deallocate) and starting your VMs as needed is another way to achieve elasticity. Since the capacity is not kept once you stop (deallocate) a VM, starting it again may result in an **AllocationFailure** error.
- VM start and redeploy operations will continue to respect the Proximity Placement Group once successfully configured.

Planned maintenance and Proximity Placement Groups

Planned maintenance events, like hardware decommissioning at an Azure datacenter, could potentially affect the alignment of resources in proximity placement groups. Resources may be moved to a different data center, disrupting the collocation and latency expectations associated with the proximity placement group.

Check the alignment status

You can do the following to check the alignment status of your proximity placement groups.

- Proximity placement group colocation status can be viewed using the portal, CLI, and PowerShell.
    - PowerShell - colocation status can be obtained through Get-AzProximityPlacementGroup cmdlet by including the optional parameter '-ColocationStatus`.
    - CLI - colocation status can be obtained through `az ppg show` by including the optional parameter '--include-colocation-status`.

- For each proximity placement group, a **colocation status** property provides the current alignment status summary of the grouped resources.
    - **Aligned**: Resource is within the same latency envelop of the proximity placement group.
    - **Unknown**: At least one of the VM resources are deallocated. After re-starting them successfully, the status should go back to **Aligned**.
    - **Not aligned**: At least one VM resource is not aligned with the proximity placement group. The specific resources that are not aligned will also be called out separately in the membership section
- For Availability Sets, you can see information about alignment of individual VMs in the Availability Set Overview page.
- For scale sets, information about alignment of individual instances can be seen in the **Instances** tab of the **Overview** page for the scale set.

Realign resources

If a proximity placement group is `Not Aligned`, you can stop\deallocate, and then restart the affected resources. If the VM is in an availability set or a scale set, all VMs in the availability set or scale set must be stopped\deallocated first before restarting them.

If there is an allocation failure due to deployment constraints, you may have to stop\deallocate all resources in the affected proximity placement group (including the aligned resources) first, and then restart them to restore alignment.

Best practices

- For the lowest latency, use proximity placement groups together with accelerated networking. For more information, see Create a Linux virtual machine with Accelerated Networking or Create a Windows virtual machine with Accelerated Networking.
- In order to avoid landing on hardware that doesn't support all the VM SKUs and sizes you require, use intent for proximity placement groups. If it is an already existing proximity placement group without intent, you can use a single ARM template with all VM sizes specified to avoid this issue.
- When reusing an existing placement group from which VMs were deleted, wait for the deletion to fully complete before adding VMs to it.

- If latency is your first priority, put VMs in a proximity placement group and the entire solution in an availability zone. But, if resiliency is your top priority, spread your instances across multiple availability zones (a single proximity placement group cannot span zones).

Next steps

- Deploy a VM to a proximity placement group using the Azure CLI or PowerShell.
- Learn how to test network latency.
- Learn how to optimize network throughput.
- Learn how to use proximity placement groups with SAP applications.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy VMs to proximity placement groups using Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a proximity placement group.

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

Create the proximity placement group

Create a proximity placement group using az ppg create.

Azure CLI

```
az group create --name myPPGGroup --location eastus
az ppg create \
   -n myPPG \
   -g myPPGGroup \
   -l eastus \
   -t standard \
   --intent-vm-sizes Standard_E64s_v4 Standard_M416ms_v2 \
   -z 1
```

List proximity placement groups

You can list all of your proximity placement groups using az ppg list.

Azure CLI

```
az ppg list -o table
```

Show proximity placement group

You can see the proximity placement group details and resources using az ppg show

Azure CLI

```
az ppg show --name myPPG --resource-group myPPGGroup
{  "availabilitySets": [],
   "colocationStatus": null,
   "id":
"/subscriptions/[subscriptionId]/resourceGroups/myPPGGroup/providers/Microso
ft.Compute/proximityPlacementGroups/MyPPG",
   "intent": {
    "vmSizes": [
      "Standard_E64s_v4",
      "Standard_M416ms_v2"
    ]
   },
   "location": "eastus",
   "name": "MyPPG",
   "proximityPlacementGroupType": "Standard",
   "resourceGroup": "myPPGGroup",
   "tags": {},
   "type": "Microsoft.Compute/proximityPlacementGroups",
   "virtualMachineScaleSets": [],
   "virtualMachines": [],
   "zones": [
    "1"
   ]
}
```

Create a VM

> **ⓘ Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Create a VM within the proximity placement group using new az vm.

Azure CLI

```
az vm create \
   -n myVM \
   -g myPPGGroup \
   --image Ubuntu2204 \
   --orchestration-mode "Uniform"
   --ppg myPPG  \
   --generate-ssh-keys \
```

```
    --size Standard_E64s_v4 \
    -l eastus
```

You can see the VM in the proximity placement group using az ppg show.

Azure CLI

```
az ppg show --name myppg --resource-group myppggroup --query
"virtualMachines"
```

Availability Sets

You can also create an availability set in your proximity placement group. Use the same `--ppg` parameter with az vm availability-set create to add all VMs in the availability set to the same proximity placement group.

Scale sets

You can also create a scale set in your proximity placement group. Use the same `--ppg` parameter with az vmss create to create a scale set and all of the instances will be created in the same proximity placement group.

Next steps

Learn more about the Azure CLI commands for proximity placement groups.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy VMs to proximity placement groups using Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a proximity placement group.

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

Create a proximity placement group

Create a proximity placement group using the New-AzProximityPlacementGroup cmdlet.

Azure PowerShell

```
$resourceGroup = "myPPGResourceGroup"
$location = "East US"
$ppgName = "myPPG"
$zone = "1"
$vmSize1 = "Standard_E64s_v4"
$vmSize2 = "Standard_M416ms_v2"
New-AzResourceGroup -Name $resourceGroup -Location $location
$ppg = New-AzProximityPlacementGroup `
   -Location $location `
   -Name $ppgName `
   -ResourceGroupName $resourceGroup `
   -ProximityPlacementGroupType Standard `
   -Zone $zone `
   -IntentVMSizeList $vmSize1, $vmSize2
```

List proximity placement groups

You can list all of the proximity placement groups using the Get-AzProximityPlacementGroup cmdlet.

Azure PowerShell

```
Get-AzProximityPlacementGroup -ResourceGroupName $resourceGroup -Name
$ppgName

ResourceGroupName           : myPPGResourceGroup
ProximityPlacementGroupType : Standard
Id                          :
/subscriptions/[subscriptionId]/resourceGroups/myPPGResourceGroup/providers/
Microsoft.Compute/proximityPlacementGroups/myPPG
Name                        : myPPG
Type                        : Microsoft.Compute/proximityPlacementGroups
Location                    : eastus
Tags                        : {}
Intent                      :
  VmSizes[0]                : Standard_E64s_v4
  VmSizes[1]                : Standard_M416ms_v2
Zones[0]                    : 1
```

Create a VM

Create a VM in the proximity placement group using `-ProximityPlacementGroup $ppg.Id` to refer to the proximity placement group ID when you use New-AzVM to create the VM.

Azure PowerShell

```
$vmName = "myVM"

New-AzVm `
  -ResourceGroupName $resourceGroup `
  -Name $vmName `
  -Location $location `
  -ProximityPlacementGroup $ppg.Id
```

You can see the VM in the placement group using Get-AzProximityPlacementGroup.

Azure PowerShell

```
Get-AzProximityPlacementGroup -ResourceId $ppg.Id |
    Format-Table -Property VirtualMachines -Wrap
```

Move an existing VM into a proximity placement group

You can also add an existing VM to a proximity placement group. You need to stop\deallocate the VM first, then update the VM and restart.

Azure PowerShell

```
$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPGResourceGroup -
Name myPPG
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM
Stop-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
Update-AzVM -VM $vm -ResourceGroupName $vm.ResourceGroupName -
ProximityPlacementGroupId $ppg.Id
Start-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
```

Move an existing VM out of a proximity placement group

To remove a VM from a proximity placement group, you need to stop\deallocate the VM first, then update the VM and restart.

Azure PowerShell

```
$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPGResourceGroup -
Name myPPG
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM
Stop-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
$vm.ProximityPlacementGroup = ""
Update-AzVM -VM $vm -ResourceGroupName $vm.ResourceGroupName
Start-AzVM -Name $vm.Name -ResourceGroupName $vm.ResourceGroupName
```

Availability Sets

You can also create an availability set in your proximity placement group. Use the same `-ProximityPlacementGroup` parameter with the New-AzAvailabilitySet cmdlet to create an availability set and all of the VMs created in the availability set will also be created in the same proximity placement group.

To add or remove an existing availability set to a proximity placement group, you first need to stop all of the VMs in the availability set.

Move an existing availability set into a proximity placement group

Azure PowerShell

```
$resourceGroup = "myResourceGroup"
$avSetName = "myAvailabilitySet"
$avSet = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup -Name
$avSetName
```

```
$vmIds = $avSet.VirtualMachinesReferences
foreach ($vmId in $vmIDs){
    $string = $vmID.Id.Split("/")
    $vmName = $string[8]
    Stop-AzVM -ResourceGroupName $resourceGroup -Name $vmName -Force
    }

$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPG -Name myPPG
Update-AzAvailabilitySet -AvailabilitySet $avSet -ProximityPlacementGroupId
$ppg.Id
foreach ($vmId in $vmIDs){
    $string = $vmID.Id.Split("/")
    $vmName = $string[8]
    Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
    }
```

Move an existing availability set out of a proximity placement group

Azure PowerShell

```
$resourceGroup = "myResourceGroup"
$avSetName = "myAvailabilitySet"
$avSet = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup -Name
$avSetName
$vmIds = $avSet.VirtualMachinesReferences
foreach ($vmId in $vmIDs){
    $string = $vmID.Id.Split("/")
    $vmName = $string[8]
    Stop-AzVM -ResourceGroupName $resourceGroup -Name $vmName -Force
    }

$avSet.ProximityPlacementGroup = ""
Update-AzAvailabilitySet -AvailabilitySet $avSet
foreach ($vmId in $vmIDs){
    $string = $vmID.Id.Split("/")
    $vmName = $string[8]
    Start-AzVM -ResourceGroupName $resourceGroup -Name $vmName
    }
```

Scale sets

You can also create a scale set in your proximity placement group. Use the same `-ProximityPlacementGroup` parameter with New-AzVmss to create a scale set and all of the instances will be created in the same proximity placement group.

To add or remove an existing scale set to a proximity placement group, you first need to stop the scale set.

Move an existing scale set into a proximity placement group

Azure PowerShell

```
$ppg = Get-AzProximityPlacementGroup -ResourceGroupName myPPG -Name myPPG
$vmss = Get-AzVmss -ResourceGroupName myVMSSResourceGroup -VMScaleSetName
myScaleSet
Stop-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName
$vmss.ResourceGroupName
Update-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName
$vmss.ResourceGroupName -ProximityPlacementGroupId $ppg.Id
Start-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName
$vmss.ResourceGroupName
```

Move an existing scale set out of a proximity placement group

Azure PowerShell

```
$vmss = Get-AzVmss -ResourceGroupName myVMSSResourceGroup -VMScaleSetName
myScaleSet
Stop-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName
$vmss.ResourceGroupName
$vmss.ProximityPlacementGroup = ""
Update-AzVmss -VirtualMachineScaleSet $vmss -VMScaleSetName $vmss.Name -
ResourceGroupName $vmss.ResourceGroupName
Start-AzVmss -VMScaleSetName $vmss.Name -ResourceGroupName
$vmss.ResourceGroupName
```

Next steps

You can also use the Azure CLI to create proximity placement groups.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a proximity placement group using the Azure portal

Article • 12/04/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

To get VMs as close as possible, achieving the lowest possible latency, you should deploy them within a proximity placement group.

A proximity placement group is a logical grouping used to make sure that Azure compute resources are physically located close to each other. Proximity placement groups are useful for workloads where low latency is a requirement.

> ⓘ **Note**
>
> Proximity placement groups cannot be used with dedicated hosts.
>
> Intent for proximity placement groups is not supported on Azure portal. Use ARM templates or other client tools like Powershell or CLI to provide intent for proximity placement groups.
>
> If you want to use availability zones together with placement groups, you need to make sure that the VMs in the placement group are also all in the same availability zone.
>
> Accelerated networking will only work if the virtual machines are deployed to an availability set or a virtual machine scale set that is linked to a proximity placement group. Having only a proximity placement group will disable the hardware acceleration feature which affects network performance.

Create the proximity placement group

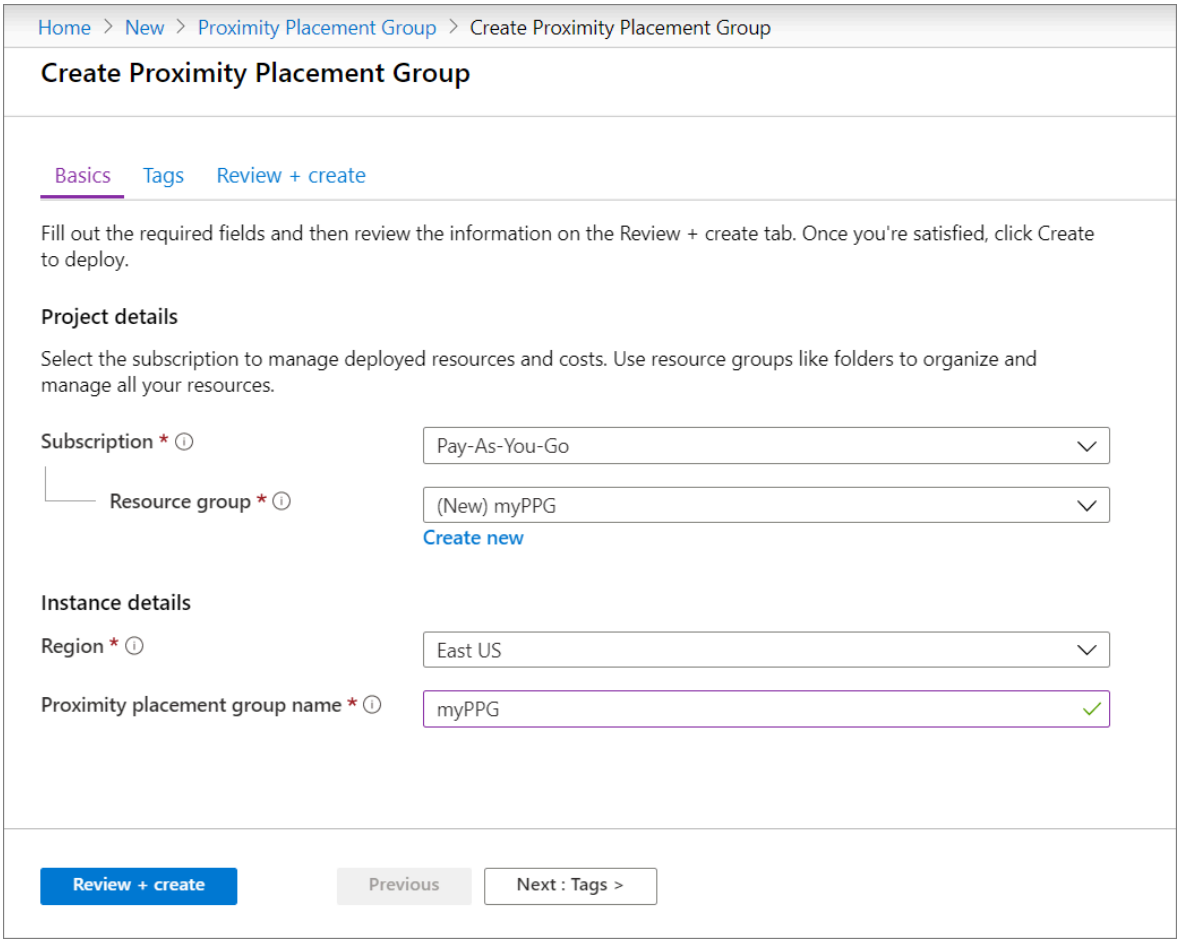
1. Type **proximity placement group** in the search.
2. Under **Services** in the search results, select **Proximity placement groups**.
3. In the **Proximity placement groups** page, select **Add**.
4. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected.

5. In **Resource group** either select **Create new** to create a new group or select an empty resource group that already exists, from the drop-down.

6. In **Region** select the location where you want the proximity placement group to be created.

7. In **Proximity placement group name** type a name and then select **Review + create**.

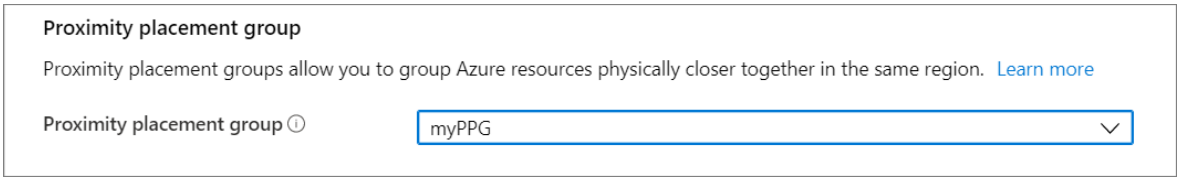
8. After validation passes, select **Create** to create the proximity placement group.



Create a VM

1. While creating a VM in the portal, go to the **Advanced** tab.

2. In the **Proximity placement group** selection, select the correct placement group.



3. When you are done making all of the other required selections, select **Review + create**.

4. After it passes validation, select **Create** to deploy the VM in the placement group.

Add VMs in an availability set to a proximity placement group

If the VM is part of the Availability set, you need to add the availability set into the placement group, before adding the VMs.

1. In the portal search for *Availability sets* and select your availability set from the results.
2. Stop\deallocate each VM in the availability set by selecting the VM, then selecting **Stop** on the page for the VM, and then select **OK** to stop the VM.
3. On the page for your availability set, make sure all of the VMs have the **Status** listed as **Stopped (deallocated)**.
4. In the left menu, select **Configuration**.
5. Under **Proximity placement group**, select a placement group from the drop-down, and then select **Save**.
6. Select **Overview** from the left menu to see the list of VMs again.
7. Select each VM in the availability set, and then select **Start** on the page for each VM.

Add existing VM to placement group

1. On the page for the VM, select **Stop**.
2. Once the status of the VM is listed as **Stopped (deallocated)**, select **Configuration** on the left menu.
3. Under **Proximity placement group**, select a placement group from the drop-down, and then select **Save**.
4. Select **Overview** from the left menu, then select **Start** to restart the VM.

Next steps

You can also use the Azure PowerShell to create proximity placement groups.

Feedback

Was this page helpful? Yes No

Compute throttling limits

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Microsoft Compute implements throttling mechanism to help with the overall performance of the service and to give a consistent experience to the customers. API requests that exceed the maximum allowed limits are throttled and users get an HTTP 429 error. All Compute throttling policies are implemented on a per-region basis.

How do the throttling policies work?

Microsoft Compute implements throttling policies that limit the number of API requests made per resource and per subscription per region per minute. If the number of API requests exceeds these limits, the requests are throttled. Here's how these limits work:

- **Per Resource Limit** – Each resource, such as a virtual machine (VM), has a specific limit for API requests. For instance, let us assume that a user creates 10 VMs in a subscription. The user can invoke up to 12 update requests for each VM in one minute. If the user exceeds the limit for the VM, API requests are throttled. This limit ensures that a few resources don’t consume the subscription level limits and throttle other resources.
- **Subscription Limit** – In addition to resource limits, there's an overarching limit on the number of API requests across all resources within a subscription. Any API requests beyond this limit are throttled, regardless of whether the limit for an individual resource has been reached. For instance, let us assume that a user has 200 VMs in a subscription. Even though user is entitled to initiate up to 12 Update VM requests for each VM, the aggregate limit for Update VM API requests is capped at 1500 per min. Any Update VM API requests for the subscription exceeding 1500 are throttled.

How does Microsoft Compute determine throttling limits?

To determine the limits for each resource and subscription, Microsoft Compute uses **Token Bucket Algorithm.** This algorithm creates buckets for each limit and holds a specific number of tokens in each bucket. The number of tokens in a bucket represent the throttling limit at any given minute.

At the start of throttling window, when the resource is created, the bucket is filled to its *Maximum Capacity*. Each API request initiated by the user consumes one token. When the token count depletes to zero, subsequent API requests are throttled. Bucket is replenished with new tokens every minute at a consistent rate called *Bucket Refill Rate* for a resource and a subscription.

For Instance: Let us consider the 'throttling policy for VM Update API' that stipulates a Bucket Refill Rate of four tokens per minute, and a Maximum Bucket Capacity of 12 tokens. The user invokes the Update VM API request for a virtual machine (VM) as per the following table. Initially, the bucket is filled with 12 tokens at the start of the throttling window. By the fourth minute, the user utilizes all 12 tokens, leaving the bucket empty. In the fifth minute, the bucket is replenished with four new tokens in accordance with the Bucket Refill Rate. So, four API requests can be made in the fifth minute, while Microsoft Compute throttles one API request due to insufficient tokens.

Expand table

(min)	1st	2nd	3rd	4th	5th	6th
Number of tokens in the beginning (A)	12	12	8	12	4	4
Requests per minute (B)	0	8	0	13	5	0
Throttled requests (C)	0	0	0	1	1	0
Remaining tokens at the end of period D = Max(A-B, 0)	12	4	8	0	0	4

Similar process is followed for determining the throttling limits at subscription level. The following sections detail the Bucket refill rate and Maximum bucket capacity that is used to determine throttling limits for Virtual Machines, Virtual Machine Scale Sets and Virtual Machines Scale Set VMs.

Throttling limits for Virtual Machines

API requests for Virtual Machines are categorized into seven distinct policies. Each policy has its own limits, depending upon how resource intensive the API requests under that policy are. Following table contains a comprehensive list of these policies, the corresponding REST APIs, and their respective throttling limits:

Expand table

Policy category	REST APIs	Resource Level	Resource Level	Subscription Level	Subscription Level
		Bucket refill rate (Per Min)	Maximum Bucket capacity (Per Min)	Bucket refill rate (Per Min)	Maximum Bucket capacity (Per Min)
Put VM (Create new VMs)	Create	4	12	500	1,500
Update VM (Update existing VMs)	Update Reapply Restart Power Off Start Generalize Convert To Managed Disks Redeploy Perform Maintenance Capture Run Command Create Or Update Extensions - Update Extensions - Delete Reimage Update Run Commands - Update Run Commands - Delete Run Commands - Create Or Update	4	12	500	1,500
Delete VM (Delete VMs)	Delete Simulate Eviction Deallocate	4	12	500	1,500
Low Cost Get VM (Get information on single VM)	Get Instance View Extensions - Get List Available Sizes Retrieve Boot	12	36	8,000	24,000

Policy category	REST APIs	Resource Level	Resource Level	Subscription Level	Subscription Level
	Diagnostics Data Run Commands - Get By Virtual Machine Run Commands - List By Virtual Machine				
High Cost Get VM[1] (Get information on multiple VMs)	List List All List By Location	NA	NA	300	900
Get Operation (Get information on async VM operations)	Status of asynchronous operations	15	45	5,000	15,000
VM Guest Patch Operations (Assess & install guest patches)	Assess Patches Install Patches	2	6	200	600

[1] Only subscription level policies are applicable.

Throttling limits for Virtual Machine Scale Sets

API requests for Virtual Machine Scale Set(Uniform & Flex) are categorized into 5 distinct policies. Each policy has its own limits, depending upon how resource intensive the API requests under that policy are. These policies are applicable to both Flex and Uniform orchestration modes. Following table contains a comprehensive list of these policies, the corresponding REST APIs, and their respective throttling limits:

Expand table

Policy category	REST APIs	Resource Level	Resource Level	Subscription Level	Subscription Level
		Bucket refill rate	Maximum Bucket	Bucket refill rate (Per Min)	Maximum Bucket

Policy category	REST APIs	Resource Level (Per Min)	Resource Level capacity (Per Min)	Subscription Level	Subscription Level capacity (Per Min)
Put (Create new scale set)	Create	4	12	125	375
Update (Update existing scaleset)	Update Start[2] Restart[2] Redeploy[2] Perform Maintenance[2] Reimage[2] Reimage All[2] Create Or Update Rolling Upgrades - Cancel Extensions - Create Extensions - Update Extensions - Delete Force Recovery Service Fabric Platform Update Domain Walk Convert To Single Placement Group Set Orchestration Service State	4	12	500	1,500
Delete (Delete scale set)	Delete Power Off[2] Deallocate	4	12	175	525
Low Cost Get (Get information on single scale set)	Get List Skus Rolling Upgrades - Get Latest Get OS Upgrade History	12	36	800	2,400
High Cost Get (Get resource	Get Instance View List[2]	10	30	360	1,080

Policy category	REST APIs	Resource Level	Resource Level	Subscription Level	Subscription Level
intensive information)	List All[2] List By Location[2]				

[2] Only subscription level policies are applicable.

Throttling limits for Virtual Machine Scale Set Virtual Machines

API requests for Virtual Machine Scale Set Virtual Machines are categorized into 3 distinct policies. Each policy has its own limits, depending upon how resource intensive the API requests under that policy are. Following table contains a comprehensive list of these policies, the corresponding REST APIs, and their respective throttling limits:

Expand table

Policy category	REST APIs	Resource Level	Resource Level	Subscription Level	Subscription Level
		Bucket refill rate (Per Min)	Maximum Bucket capacity (Per Min)	Bucket refill rate (Per Min)	Maximum Bucket capacity (Per Min)
Update scale set VMs (Update existing VMs in a scale set)	Start Restart Reimage ReimageAll Update SimulateEviction Extensions- Create Or Update RunCommands - Create Or Update RunCommands - Update	4	12	500	1,500
Delete scale set VMs (Delete scale set VMs)	Delete PowerOff Deallocate Extensions- Delete RunCommands - Delete	4	12	500	1,500

Policy category	REST APIs	Resource Level	Resource Level	Subscription Level	Subscription Level
Get scale set VMs (Get information on scale set VMs)	Get GetInstance View Extensions- Get RunCommands - Get RetrieveBoot Diagnostics Data	12	36	2,000	6,000

Troubleshooting guidelines

In case users are still facing challenges due to Compute throttling, refer to Troubleshooting throttling errors in Azure - Virtual Machines. It has details on how to troubleshoot throttling issues, and best practices to avoid being throttled.

FAQs

Is there any action required from users?

Users don't need to change anything in their configuration or workloads. All existing APIs continue to work as is.

What benefits do the throttling policies provide?

The throttling policies offer several benefits:

- All Compute resources have a uniform window of 1 min. Users can successfully invoke API calls, 1 min after getting throttled.
- No single resource can use up all the limits under a subscription as limits are defined at resource level.
- Microsoft Compute is introducing a new algorithm, Token Bucket Algorithm, for determining the limits. The algorithm provides extra buffer to the customers, while making high number of API requests.

Does the customer get an alert when they're about to reach their throttling limits?

As part of every response, Microsoft Compute returns **x-ms-ratelimit-remaining-resource** which can be used to determine the throttling limits against the policies. A list of applicable throttling policies is returned as a response to Call rate informational headers.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Regions for virtual machines in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

It is important to understand how and where your virtual machines (VMs) operate in Azure, along with your options to maximize performance, availability, and redundancy. This article provides you with an overview of the availability and redundancy features of Azure.

What are Azure regions?

Azure operates in multiple datacenters around the world. These datacenters are grouped in to geographic regions, giving you flexibility in choosing where to build your applications.

You create Azure resources in defined geographic regions like 'West US', 'North Europe', or 'Southeast Asia'. You can review the list of regions and their locations. Within each region, multiple datacenters exist to provide for redundancy and availability. This approach gives you flexibility as you design applications to create VMs closest to your users and to meet any legal, compliance, or tax purposes.

Special Azure regions

Azure has some special regions that you may wish to use when building out your applications for compliance or legal purposes. These special regions include:

- **US Gov Virginia** and **US Gov Iowa**
  - A physical and logical network-isolated instance of Azure for US government agencies and partners, operated by screened US persons. Includes additional compliance certifications such as FedRAMP and DISA. Read more about Azure Government.
- **China East** and **China North**
  - These regions are available through a unique partnership between Microsoft and 21Vianet, whereby Microsoft does not directly maintain the datacenters.
- **Germany Central** and **Germany Northeast**
  - These regions are available via a data trustee model whereby customer data remains in Germany under control of T-Systems, a Deutsche Telekom company, acting as the German data trustee.

Region pairs

Each Azure region is paired with another region within the same geography (such as US, Europe, or Asia). This approach allows for the replication of resources, such as VM storage, across a geography that should reduce the likelihood of natural disasters, civil unrest, power outages, or physical network outages affecting both regions at once. Additional advantages of region pairs include:

- In the event of a wider Azure outage, one region is prioritized out of every pair to help reduce the time to restore for applications.
- Planned Azure updates are rolled out to paired regions one at a time to minimize downtime and risk of application outage.
- Data continues to reside within the same geography as its pair (except for Brazil South) for tax and law enforcement jurisdiction purposes.

Examples of region pairs include:

Expand table

Primary	Secondary
West US	East US
North Europe	West Europe
Southeast Asia	East Asia

You can see the full list of regional pairs here.

Feature availability

Some services or VM features are only available in certain regions, such as specific VM sizes or storage types. There are also some global Azure services that do not require you to select a particular region, such as Microsoft Entra ID, Traffic Manager, or Azure DNS. To assist you in designing your application environment, you can check the availability of Azure services across each region. You can also programmatically query the supported VM sizes and restrictions in each region.

Storage availability

Understanding Azure regions and geographies becomes important when you consider the available storage replication options. Depending on the storage type, you have

different replication options.

Azure Managed Disks

- Locally redundant storage (LRS)
  - Replicates your data three times within the region in which you created your storage account.

Storage account-based disks

- Locally redundant storage (LRS)
  - Replicates your data three times within the region in which you created your storage account.
- Zone redundant storage (ZRS)
  - Replicates your data three times across two to three facilities, either within a single region or across two regions.
- Geo-redundant storage (GRS)
  - Replicates your data to a secondary region that is hundreds of miles away from the primary region.
- Read-access geo-redundant storage (RA-GRS)
  - Replicates your data to a secondary region, as with GRS, but also then provides read-only access to the data in the secondary location.

The following table provides a quick overview of the differences between the storage replication types:

Expand table

Replication strategy	LRS	ZRS	GRS	RA-GRS
Data is replicated across multiple facilities.	No	Yes	Yes	Yes
Data can be read from the secondary location and from the primary location.	No	No	No	Yes
Number of copies of data maintained on separate nodes.	3	3	6	6

You can read more about Azure Storage replication options here. For more information about managed disks, see Azure Managed Disks overview.

Next steps

For more information, see Azure regions.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Availability sets overview

Applies to: ✔ Linux VMs ✔ Windows VMs

This article provides an overview of the availability features of Azure virtual machines (VMs).

> ⓘ **Note**
>
> We recommend that customers choose Virtual Machine Scale Sets with flexible orchestration mode for high availability with the widest range of features. Virtual Machine Scale Sets:
>
> - Allow VM instances to be centrally managed, configured, and updated.
> - Automatically increase or decrease the number of VM instances in response to demand or a defined schedule.
>
> Availability sets provide high availability but don't offer the same level of resiliency as availability zones.

What is an availability set?

Availability sets are logical groupings of VMs that reduce the chance of correlated failures bringing down related VMs at the same time. Availability sets distribute VMs across multiple fault domains, improving reliability. This configuration is especially useful in regions that don't support availability zones.

When you use availability sets, create two or more VMs within an availability set. Using two or more VMs in an availability set helps keep applications highly available and meets the 99.95% Azure service-level agreement (SLA). There's no extra cost for using availability sets. You only pay for each VM instance that you create.

Availability sets provide lower VM-to-VM latency than availability zones because the VMs are placed in closer physical proximity. Availability sets have fault isolation for many possible failures, to minimize single points of failure and to offer high availability. Availability sets are still susceptible to certain shared infrastructure failures, such as datacenter-level network outages, which can affect multiple fault domains.

For more reliability than availability sets offer, use availability zones. Availability zones have the highest reliability. Each VM is deployed in multiple datacenters to help protect you from loss of power, networking, or cooling in an individual datacenter. If your highest priority is the best reliability for your workload, replicate your VMs across multiple availability zones.

How do availability sets work?

The underlying Azure platform assigns an *update domain* and a *fault domain* to each virtual machine in your availability set. Each availability set can have up to 3 fault domains and 20 update domains. These settings can't be changed after the availability set is created.

Update domains

Update domains indicate groups of virtual machines and underlying physical hardware that can be restarted at the same time.

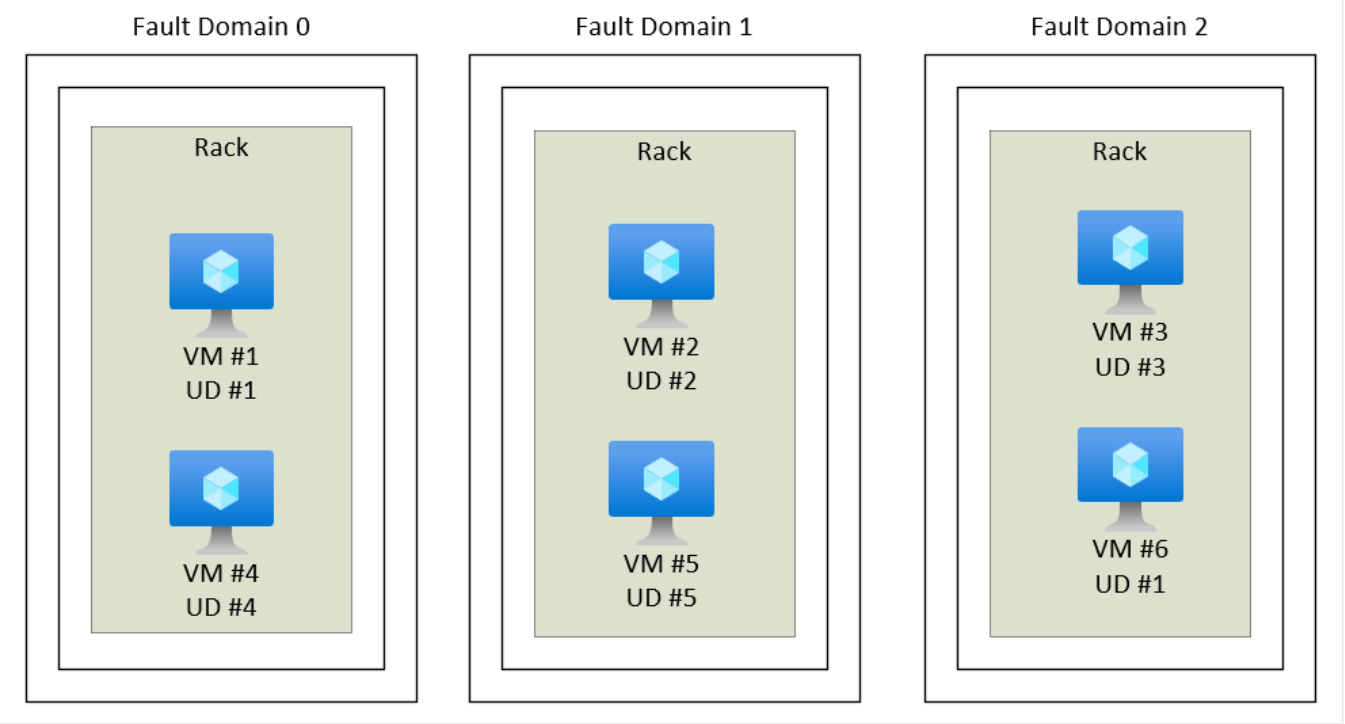
If more than five VMs are configured in an availability set with five update domains, the sixth VM is placed in the same update domain as the first, the seventh with the second, and so on.

The order of update domains being restarted might not proceed sequentially during planned maintenance, but only one update domain is restarted at a time. A restarted update domain has 30 minutes to recover before maintenance starts on a different update domain.

Fault domains

Fault domains define the group of virtual machines that share a common power source and network switch. By default, the virtual machines configured within your availability set are separated across up to three fault domains.

Placing your virtual machines into an availability set doesn't protect your application from operating system or application-specific failures. But it does limit the impact of potential physical hardware failures, network outages, or power interruptions.



Disk fault domains

VMs are also aligned with disk fault domains. This alignment ensures that all the managed disks attached to a VM are within the same fault domains.

Only VMs with managed disks can be created in a managed availability set. The number of managed-disk fault domains varies by region: either two or three managed-disk fault domains per region.

The following command retrieves a list of fault domains per region:

Azure CLI

```
az vm list-skus --resource-type availabilitySets --query '[?name==`Aligned`].
{Location:locationInfo[0].location, MaximumFaultDomainCount:capabilities[0].value}'
-o Table
```

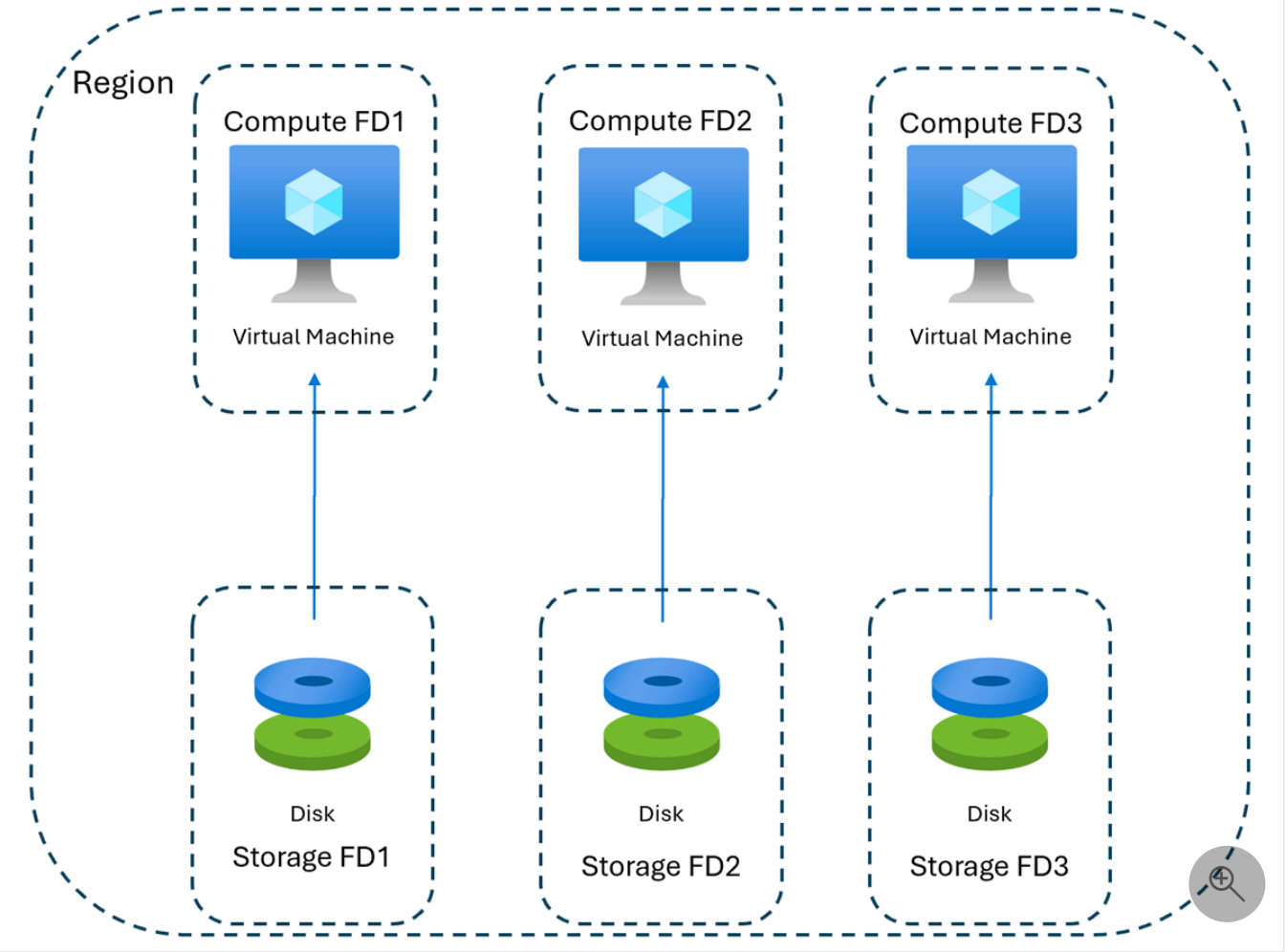
Shared fault domains

Under certain circumstances, two VMs in the same availability set might share a fault domain. You can confirm a shared fault domain by going to your availability set and checking the **Fault Domain** column.

If your VMs have a shared fault domain, it could be because you took these actions when deploying your VMs:

1. Deploy the first VM.
2. Stop or deallocate the first VM.
3. Deploy the second VM.

Under these circumstances, the OS disk of the second VM might be created on the same fault domain as the first VM, so the two VMs will be on same fault domain. To avoid this problem, don't stop or deallocate VMs between deployments.



Related content

- For best practices related to Azure availability, see Resiliency checklist for specific Azure services.

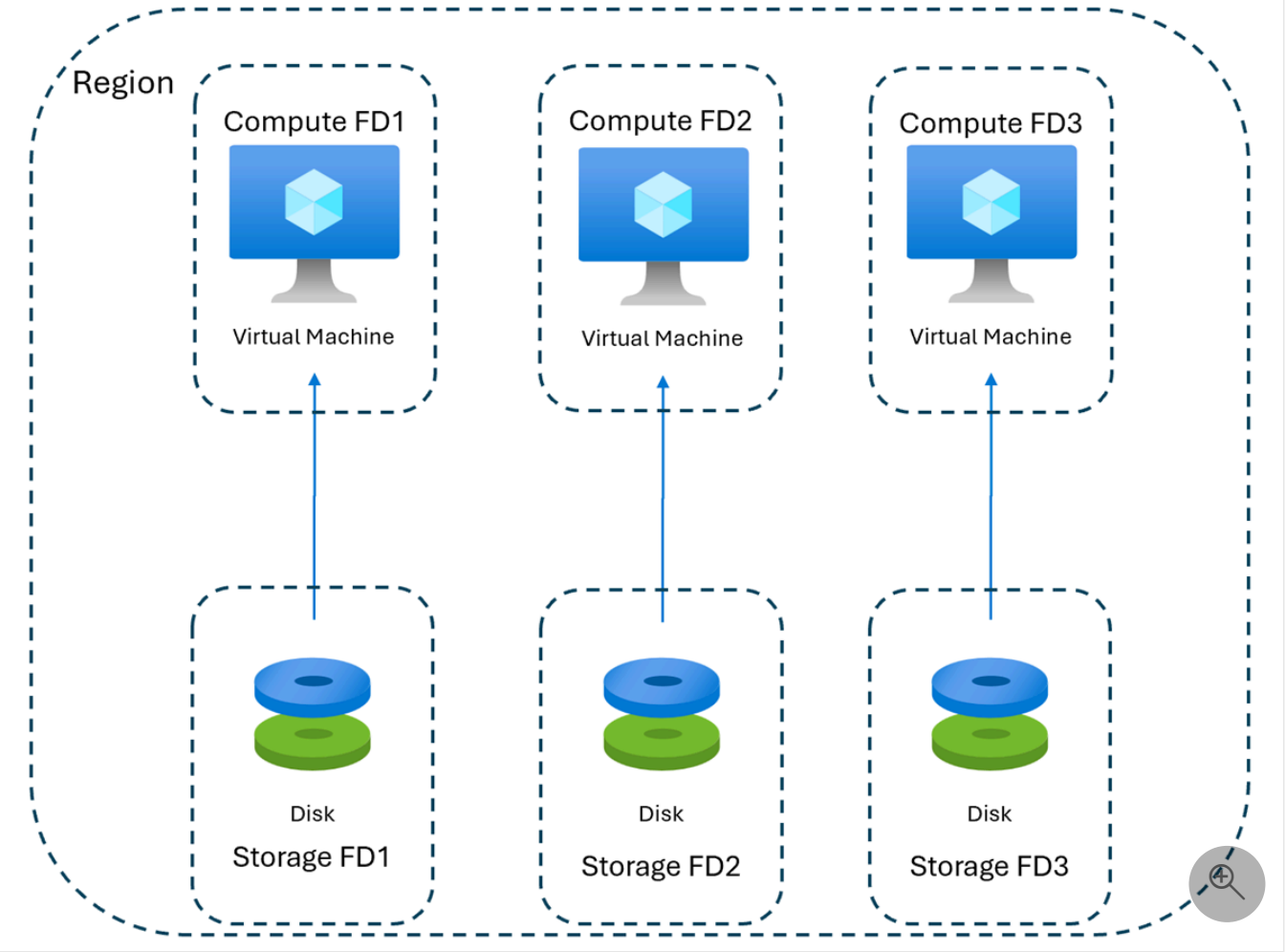
Last updated on 04/07/2026

Use premium SSD v2 with VMs in availability set

05/20/2025

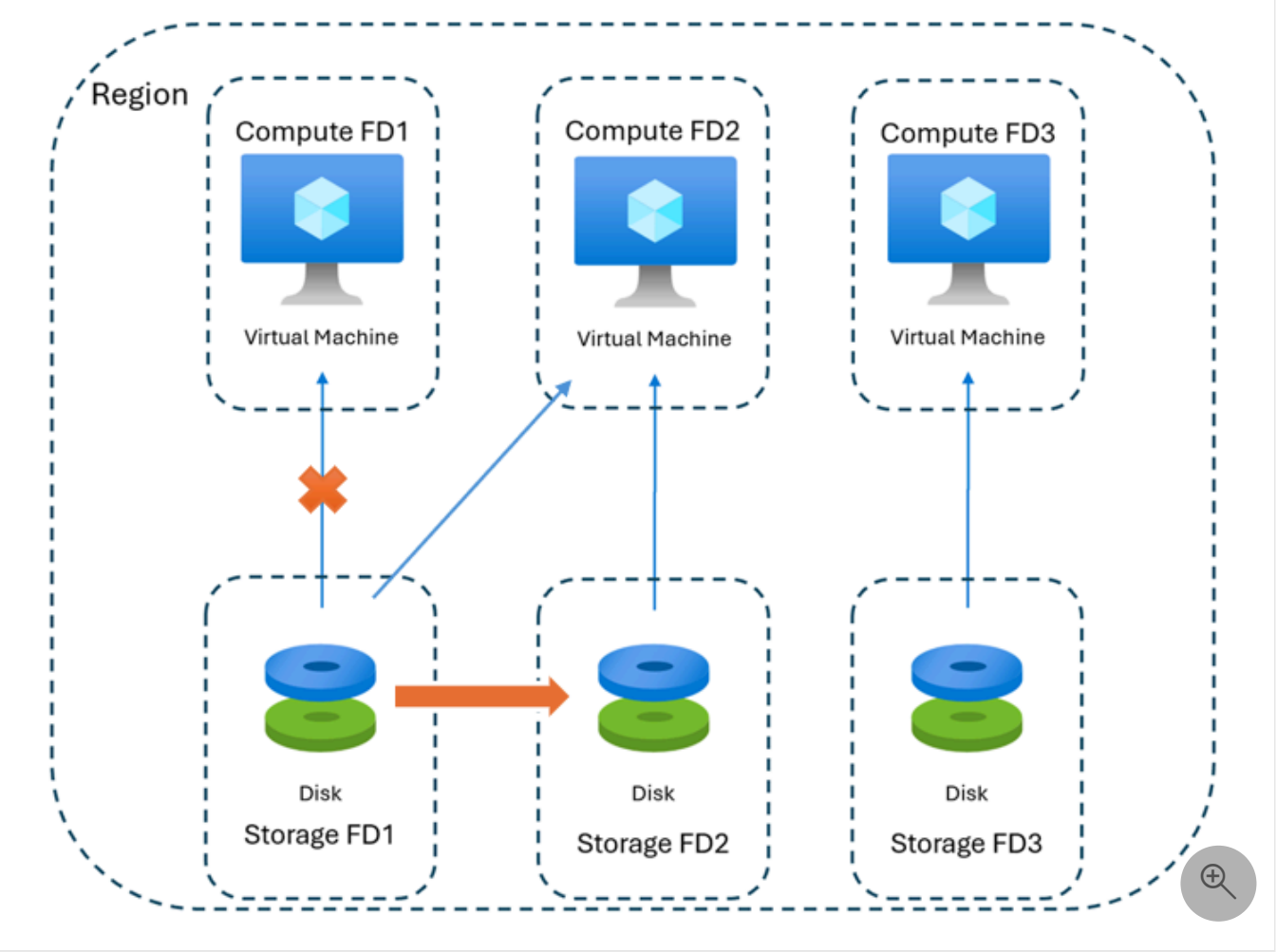
Introduction

Premium SSD v2 managed disks are supported with Azure Virtual Machines in availability sets to enhance the high availability of your applications. When VMs using Premium SSD v2 are part of an availability set, the platform ensures that their disks are automatically distributed across multiple storage fault domains. This distribution minimizes the risk of a single point of failure.



Availability sets have fault isolation for many possible failures, to minimize single points of failure and to offer high availability. If there's a failure in one storage fault domain, only the virtual machine (VM) instances with Premium SSD v2 disks on that specific fault domain are affected. Other VM instances, whose disks are placed on separate fault domains, remain unaffected and continue to operate normally. Availability sets are susceptible to certain shared infrastructure failures, like datacenter network failures, physical hardware failures or power interruptions which can affect multiple fault domains.

When a Premium SSD v2 disk located in one fault domain is attached to a VM in another fault domain, the system triggers a background copy. This process moves the disk to match the VM's fault domain, helping ensure consistent alignment between compute and storage for better reliability and availability.



As the previous diagram illustrates, if a disk located in fault domain 1 is attached to a VM in fault domain 1 and is later detached and attached to a VM in fault domain 2, the system automatically triggers a background copy of the disk to move it from fault domain 1 to fault domain 2 for compute and storage fault domain alignment. This background move process can take up to 24 hours to complete.

Regional availability

Premium SSD v2 support for VMs in an availability set is currently limited to the following regions that lack availability zone support:

- Australia Southeast
- Canada East
- North Central US
- UK West

- West Central US
- West US

Limitations

- Premium SSD v2 is supported for VMs in availability set in select regions without availability zones.
- You must register your subscription to use this feature. Follow the steps in Register Premium SSD v2 with VMs in availability sets in regions without availability zones.
- Only one background data copy can run per disk at a time. When attaching a disk to a VM in an availability set, the system might start a background copy to align with the fault domain. If you try to detach and reattach the disk while this move is in progress, the operation fails with an error. To prevent operation failure, wait until the move finishes, or set the OptimizedForFrequentAttach property on the disk. This setting skips fault domain-alignment background copies for future attachments. For more information on OptimizedForFrequentAttach, follow the instructions Optimize background data copy of the disk.
- You can't attach a disk created from a snapshot to VMs in an availability set while it's still copying data in the background. Wait until the copy process finishes before attaching the disk. To check the status of background data copy from a snapshot, follow the instructions here.
- Disk size increase and changing customer-managed key (CMK) are not supported while a background data copy for Fault Domain alignment is in progress.
- Premium SSD v2 managed disks have their own separate set of limitations, as well.

Register Premium SSD v2 with VMs in availability sets in supported regions without availability zones

The feature is only available in regions that do not support Availability Zones. If you're targeting a supported region without availability zones, ensure your subscription is registered for the required feature.

To proceed, register the feature manually:

- Use the following command to register the feature with your subscription:

Azure CLI

```
az feature registration create --namespace Microsoft.Compute --name
PV2WithAVSetRegionWithoutZone
```

- Use the command below to verify if the feature is registered :

Azure CLI

```
az feature registration show --provider Microsoft.Compute --name
PV2WithAVSetRegionWithoutZone
```

Deploy a VM and a Premium SSD v2 disk within an availability set

Azure CLI

- Create a resource group:

Azure CLI

```
az group create --name myResourceGroup --location myLocation
```

- The number of storage fault domains varies by region. The following command retrieves a list of maximum suported fault domains per region:

Azure CLI

```
az vm list-skus --resource-type availabilitySets --query '[?name==`Aligned`].
{Location:locationInfo[0].location,
MaximumFaultDomainCount:capabilities[0].value}' -o Table
```

- Create an availability set:

Azure CLI

```
az vm availability-set create -n myAvSet -g myResourceGroup --platform-fault-
domain-count myFDCount --platform-update-domain-count myUDCount

## The value for platform-fault-domain-count should be determined based on the
number of maximum supported fault domains in a given region. See previous step
to check the available fault domains per region.
```

- Create a VM:

Azure CLI

```
az vm create -n myVMname -g myResourceGroupName --availability-set myAvSetName
```

```
--image Win2016Datacenter --count MyCount
```

- Attach a new Premium SSD v2 disk to existing VMs in an availability set

Azure CLI

```
az vm disk attach -g MyResourceGroupName --vm-name MyVMname --name MyDiskName
--new --sku PremiumV2_LRS --size-gb MySize
```

- Attach existing Premium SSD v2 disk to existing VMs in an Availability Set:

Azure CLI

```
az vm disk attach -g MyResourceGroupName --vm-name MyVMname --disks MyDiskName
```

Optimize background data copy of the disk

Change optimized-for-frequent-attach disk property

If your workload often moves disks between VMs in the same or different availability sets, turn on the `optimized-for-frequent-attach` property. Setting this property to true prevents the system from triggering a background copy of the disk for fault domain alignment during reattachments. `optimized-for-frequent-attach` can be set when creating a new unattached disk or update it later for an existing disk. If the disk is currently attached to a VM, first detach the disk. Update the `optimized-for-frequent-attach` disk property and then reattach the disk to the VM.

To set the property while creating a new unattached disk:

Azure CLI

```
az disk create --name myDiskName --resource-group myResourceGroup --location
myLocation --sku PremiumV2_LRS --size-gb myGB --optimized-for-frequent-attach true
```

To update the property on an existing unattached disk:

Azure CLI

```
az disk update --name myDiskName --resource-group myResourceGroup --set
optimizedForFrequentAttach=true
```

Next steps

- Deploy a Premium SSD v2
- Availability sets overview
- Best practices for achieving high availability with Azure virtual machines and managed disks

Introduction to Azure managed disks

Article • 04/01/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure managed disks are block-level storage volumes managed by Azure and used with Azure Virtual Machines. Managed disks are like physical disks in an on-premises server, but they're virtualized. With managed disks, you only have to specify the disk type and the disk size, then provision the disk. After you provision the disk, Azure handles the rest.

There are five types of managed disks: Ultra Disks, Premium solid-state drives (SSD) v2, Premium SSD, Standard SSD, and Standard hard disk drives (HDD). To learn about each disk type and decide which fits your needs, see Azure managed disk types.

An alternative is to use Azure Elastic SAN as the storage for your virtual machine (VM). With Elastic SAN, you can consolidate the storage for all your workloads into a single storage back end. Elastic SAN can be more cost effective if you have many large-scale, I/O-intensive workloads and top-tier databases. To learn more, see What is Azure Elastic SAN?.

High durability and availability

Managed disks are designed for 99.999% availability, to achieve this availability, managed disks provide three replicas of your data. If one or two replicas experience problems, the remaining replicas help ensure persistence of your data and high tolerance against failures.

This architecture helps Azure consistently deliver high durability for infrastructure as a service (IaaS) disks, with a 0% annualized failure rate. Locally redundant storage (LRS) disks provide at least 99.999999999% (11 9's) of durability over a year. Zone-redundant storage (ZRS) disks provide at least 99.9999999999% (12 9's) of durability over a year.

Simple and scalable VM deployment

With managed disks, you can create up to 50,000 disks of each disk type in a subscription per region. You can then create thousands of VMs in a single subscription.

Managed disks increase the scalability of virtual machine scale sets. You can create up to 1,000 VMs in a virtual machine scale set by using an Azure Marketplace image or an

Azure Compute Gallery image with managed disks.

Failure isolation

Integration with availability sets

Managed disks are integrated with availability sets to help ensure that the disks of VMs in an availability set are sufficiently isolated from each other to avoid a single point of failure.

Disks are automatically placed in different storage scale units (stamps). If a stamp fails due to hardware or software failure, only the VM instances with disks on those stamps fail.

For example, say you have an application running on five VMs that are in an availability set. The disks for those VMs aren't all stored in the same stamp. So if one stamp goes down, the other instances of the application continue to run.

Integration with availability zones

Managed disks support availability zones, which help protect your applications from datacenter failures. Availability zones are unique physical locations within an Azure region. Each zone consists of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions.

For information about the service-level agreement (SLA) for VM uptime with availability zones, see the page for Azure SLAs.

Performance options

The demands and needs of your workload can shift over time, either due to high demand during a holiday, sudden bursts of traffic, or scaling up to meet client needs. Azure managed disks have several capabilities you can take advantage of to improve their performance and match the shifting needs of your workloads. Different disk types offer different capabilities, some disk types have capabilities you can use to ensure their performance automatically shifts to meet the changing demands of your workload, others require manual adjustment, and other disk types can't do either.

To learn about the options each disk type has, see Overview of options to improve Azure managed disk performance

Backup and disaster recovery options

Managed disks support several backup and disaster recovery options. These options include built-in redundancy options (locally redundant storage, and zone-redundant storage), Azure Backup, managed disk snapshots, restore points, and Azure Site Recovery. The ideal configuration of backup and disaster recovery options for your needs can vary. To decide which works best for your needs, see Backup and disaster recovery for Azure managed disks.

Snapshots

A managed disk snapshot is a read-only, crash-consistent full copy of a managed disk that's stored as a standard managed disk by default. With snapshots, you can back up your managed disks at any point in time. These snapshots exist independently of the source disk, and you can use them to create new managed disks.

To learn how to create managed disk snapshots, see Create a snapshot of a virtual hard disk.

Images

Managed disks support creating managed custom images. You can create an image from your custom VHD in a storage account or directly from a generalized (via Sysprep) VM. The image contains all managed disks associated with a VM, including both the OS and data disks. A managed custom image lets you create hundreds of VMs without the need to copy or manage any storage accounts.

For information on creating images, see Create a legacy managed image of a generalized VM in Azure.

Images vs snapshots

It's important to understand the difference between images and snapshots. With managed disks, you can take an image of a generalized VM that you deallocated. This image includes all of the disks attached to the VM. You can use this image to create a VM.

A snapshot is a copy of a disk at a point in time. It applies only to one disk. If you have a VM that has one disk (the OS disk), you can take a snapshot or an image of it and create a VM from either the snapshot or the image.

A snapshot doesn't have awareness of any disk except the one that it contains. Using snapshots in scenarios that require the coordination of multiple disks, such as striping, is problematic. Snapshots would need to be able to coordinate with each other, and that's not supported.

Upload your VHD or VHDX

You can reduce costs by uploading data to managed disks directly, without attaching them to VMs. With direct upload, you can upload VHDs up to 32 TiB in size. To learn how to upload your VHD to Azure, see the Azure CLI or Azure PowerShell articles.

Security

Control access to managed disk imports and exports

You have several options for protecting your managed disks from being imported or exported. You can create a custom Azure role-based access control (RBAC) role with a limited permission set, you can use Microsoft Entra ID, Private Links, Azure Policy, or configure the `NetworkAccessPolicy` parameter on your disk resources. To learn more, see Restrict managed disks from being imported or exported.

Encryption

Several kinds of encryption are available for your managed disks, including Server-Side Encryption (SSE), Azure Disk Encryption (ADE), encryption at host, and confidential disk encryption. You can use either platform-managed keys or customer-managed keys with these encryption options. To learn more about your encryption options, see Overview of managed disk encryption options

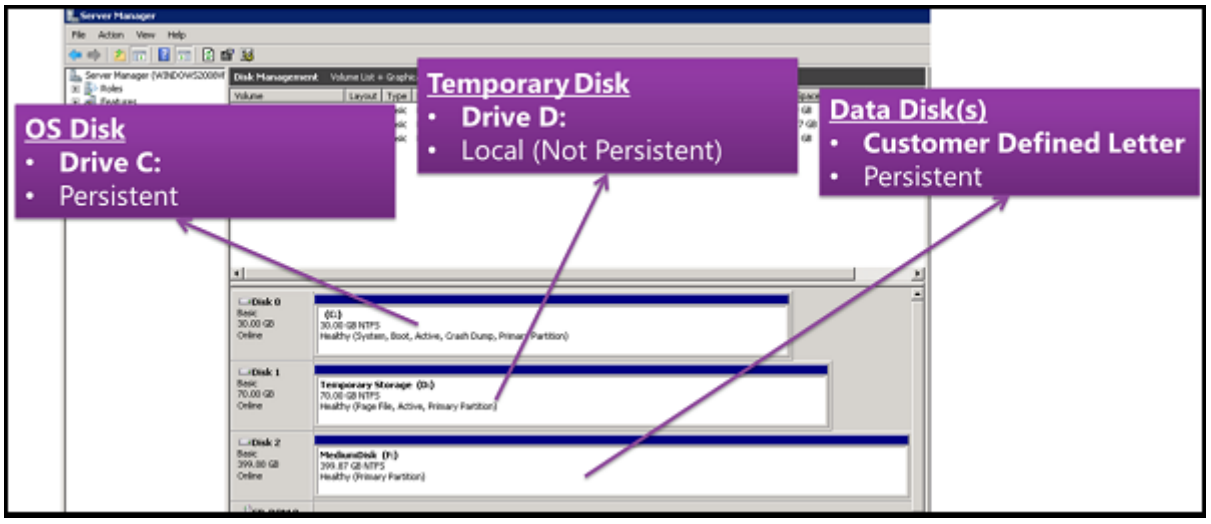
Shared disks

For use with cluster applications, you can attach an individual managed disk to multiple VMs simultaneously, allowing you to either deploy new or migrate existing clustered applications to Azure. This configuration requires a cluster manager, like Windows Server Failover Cluster (WSFC), or Pacemaker, that handles cluster node communication and write locking. To learn more about this configuration, see Share an Azure managed disk.

Disk roles

There are three main disk roles in Azure: the OS disk, the data disk, and the temporary disk. These roles map to disks that are attached to your virtual machine.

Performance for each disk role works differently. To learn more about how performance works for each role, see Disk allocation and performance.



OS disk

Every virtual machine has one attached OS disk. This disk has a preinstalled operating system, which you selected when creating the VM. This disk contains the boot volume.

Generally, you should store only your OS information on the OS disk. The data disk is where you should store all applications and data. If cost is a concern, you can use the OS disk instead of creating a data disk.

The OS disk has a maximum capacity of 4,095 gibibytes (GiB). However, many operating systems are partitioned with master boot records (MBRs) by default. An MBR limits the usable size to 2 TiB. If you need more than 2 TiB, create and attach data disks and use them for data storage. If you need to store data on the OS disk and require extra space, convert it to a GUID partition table (GPT). To learn about the differences between an MBR and a GPT on Windows deployments, see Windows and GPT FAQ.

On Azure Windows VMs, drive C is your OS disk and is persistent storage, unless you're using ephemeral OS disks.

Data disk

A data disk is a managed disk attached to a virtual machine to store application data or other data. Data disks are registered as SCSI drives and are labeled with a letter that you

choose. The size and type of the virtual machine determines how many data disks you can attach to the VM and the disk types you can use with the VM.

Generally, you should use data disks to store your applications and data, instead of storing them on an OS disk. Using data disks to store applications and data offers the following benefits over using OS disks:

- Improved backup and disaster recovery
- More flexibility and scalability
- Performance isolation
- Easier maintenance
- Improved security and access control

For more information on these benefits, see Why should I use the data disk to store applications and data instead of the OS disk?.

Temporary disk

Most VMs contain a temporary disk, which isn't a managed disk. The temporary disk provides short-term storage for applications and processes. It's intended for storing only data such as page files, swap files, or SQL Server tempdb files.

Data on the temporary disk might be lost during a maintenance event, when you redeploy a VM, or when you stop the VM. During a successful standard restart of the VM, data on the temporary disk persists. For more information about VMs without temporary disks, see Azure VM sizes with no local temporary disk.

On Azure Linux VMs, the temporary disk is typically */dev/sdb*. On Windows VMs, the temporary disk is drive D by default. The temporary disk isn't encrypted unless:

- You're using an Azure VM that is version 5 and above (such as Dsv5 or Dsv6). Azure VMs version 5 and above automatically encrypt their temporary disks and (if in use) their ephemeral OS disks with encryption-at-rest.
- For server-side encryption, you enable encryption at host.
- For Azure Disk Encryption, you set the `VolumeType` parameter to All on Windows or EncryptFormatAll on Linux.

Related content

- Learn more about the individual disk types that Azure offers, which type is fits your needs, and their performance targets, see Select a disk type for IaaS VMs
- Learn about how Virtual machine and disk performance works

- Learn the Best practices for achieving high availability with Azure virtual machines and managed disks
- Learn more about how managed disks are billed, see Understand Azure Disk Storage billing

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure managed disk types

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure managed disks currently offers five disk types, each intended to address a specific customer scenario:

- Ultra Disks
- Premium SSD v2
- Premium SSDs (solid-state drives)
- Standard SSDs
- Standard HDDs (hard disk drives)

Disk type comparison

The following table provides a comparison of the five disk types to help you decide which to use.

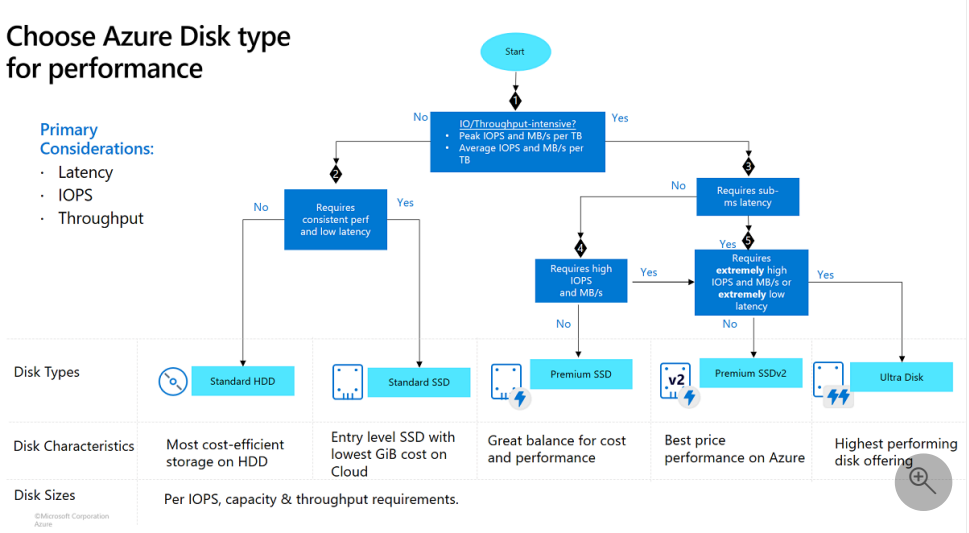
> ⓘ Note
>
> On September 8, 2028, the ability to use Standard HDDs as OS disks will be retired. To learn more, see Migrate your Standard HDD OS disks by September 08, 2028.

Expand table

	Ultra Disk	Premium SSD v2	Premium SSD	Standard SSD	Standard HDD
Disk type	SSD	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads.	Production and performance-sensitive workloads that consistently require low latency and high IOPS and throughput	Production and performance sensitive workloads	Web servers, lightly used enterprise applications and dev/test	Backup, non-critical, infrequent access
Max disk size	65,536 GiB	65,536 GiB	32,767 GiB	32,767 GiB	32,767 GiB
Max throughput	10,000 MB/s	1,200 MB/s	900 MB/s	750 MB/s	500 MB/s
Max IOPS	400,000	80,000	20,000	6,000	2,000, 3,000*
Usable as OS Disk?	No	No	Yes	Yes	Yes (retiring Sept 8, 2028)

* Only applies to disks with performance plus enabled.

For more help deciding which disk type suits your needs, this decision tree should help with typical scenarios:



For a video that covers some high level differences for the different disk types, and some ways for determining what impacts your workload requirements, see Block storage options with Azure Disk Storage and Elastic SAN.

Ultra Disks

Azure Ultra Disks are the highest-performing storage option for Azure virtual machines (VMs). You can change the performance parameters of an Ultra Disk without having to restart your VMs. Ultra Disks are suited for data-intensive workloads such as SAP HANA, top-tier databases, and transaction-heavy workloads.

Ultra Disks must be used as data disks and can only be created as empty disks. When using Ultra Disks as data disks, use Premium solid-state drives (SSDs) as operating system (OS) disks.

Ultra Disk size

Ultra Disk sizes range from 4 GiB up to 64 TiB, in 1 GiB increments. Ultra Disk capacity works like Premium SSD v2 disks and you're billed on a per GiB ratio.

By default, for each individual region, you can deploy a total of 100 TiB of capacity as Ultra Disks per subscription. If you need more than 100 TiB of Ultra Disk capacity per region per subscription, request a quota increase or contact Azure Support.

The following table provides an example of performance caps an Ultra Disk has depending on the size you set for the disk.

Expand table

Disk Size (GiB)	IOPS Cap	Throughput Cap (MB/s)
4	4,000 (1,200)*	1,000 (300)*
8	8,000 (2,400)*	2,000 (600)*
16	16,000 (4,800)*	4,000 (1,200)*
32	32,000 (9,600)*	8,000 (2,400)*
64	64,000 (19,200)*	10,000 (4,900)*
128	128,000 (38,400)*	10,000 (9,800)*
256	256,000 (76,000)*	10,000
512	400,000 (153,000)*	10,000

Disk Size (GiB)	IOPS Cap	Throughput Cap (MB/s)
1,024	400,000 (307,200)*	10,000
2,048-65,536	400,000	10,000

* Only applies during deployment of Virtual Machine Scale Sets with Uniform orchestration mode. Setting a higher value during deployment results in a failed deployment. After deployment completes you can increase the performance of your disks.

Ultra Disk performance

Ultra Disks are designed to provide consistently low sub millisecond latencies and the highest limits for IOPS and throughput. With Ultra Disks, you can individually set the capacity, throughput, and IOPS of a disk based on your workload needs, increasing flexibility and reducing costs. Each of these values determine the cost of your disk. Within a 24 hour period, you can change the performance of a disk four times. It can take up to an hour for a performance change to take effect.

Ultra Disk IOPS

Ultra Disks support IOPS limits of 1000 IOPS/GiB, up to a maximum of 400,000 IOPS per disk. To achieve the target IOPS for the disk, ensure that the selected disk IOPS are less than the VM IOPS limit. Ultra Disks with greater IOPS can be used as shared disks to support multiple VMs. The minimum baseline IOPS per disk is 100.

> **Note**
>
> Only during deployment of Ultra Disks while using Uniform Virtual Machine Scale Sets: The minimum IOPS per disk are 1 IOPS/GiB, with an overall baseline minimum of 100 IOPS. The maximum IOPS per disk are 300 IOPS/GiB, up to a maximum of 400,000 IOPS per disk. If you exceed these limits at deployment, the deployment fails. You can **increase the performance** of these disks once deployment completes.

For more information about IOPS, see Virtual machine and disk performance.

Ultra Disk throughput

The maximum throughput limit of an Ultra Disk is .25 MB/s for each provisioned IOPS, up to a maximum of 10,000 MB/s per disk (where MB/s = 10^6 Bytes per second). The minimum guaranteed throughput of an Ultra Disk is 1 MB/s.

> **Note**
>
> Only during deployment of Ultra Disks while using Uniform Virtual Machine Scale Sets: The minimum throughput of an Ultra Disk is 4-KB/s per provisioned IOPS. So if you provision a 500 GiB, 500 IOPS Ultra Disk, the minimum throughput for that disk would be 2 MB/s and the maximum throughput that can be provisioned is 125 MB/s.

You can adjust Ultra Disk IOPS and throughput performance at runtime without detaching the disk from the virtual machine. After a performance resize operation has been issued on a disk, it can take up to an hour for the change to take effect. Up to four performance resize operations are permitted during a 24-hour window.

It's possible for a performance resize operation to fail because of a lack of performance bandwidth capacity.

Ultra Disk limitations

The following list contains Ultra Disk's limitations:

- Only during deployment with Virtual Machine Scale Sets with Uniform orchestration mode: At time of deployment, Ultra Disks used with Uniform Virtual Machine Scale Sets use different formulas for IOPS and throughput for their performance. Exceeding the limits of these formulas results in a deployment failure. Once the deployment succeeds you can increase the performance of these disks. See Ultra Disk IOPS and Ultra Disk throughput for details.
- Ultra Disks can't be used as an OS disk or with Azure Compute Gallery.
- Currently, Ultra Disks only support Single VM and Availability zone infrastructure options as locally redundant storage (LRS). Ultra Disks don't support zone redundant storage (ZRS).
- Ultra Disks don't support availability sets.
- Ultra Disks don't support disk caching.

Ultra Disks support a 4k physical sector size by default but also supports a 512E sector size. Most applications are compatible with 4k sector sizes, but some require 512-byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks. For older versions of Oracle DB, 512-byte sector size is required.

The following table outlines the regions Ultra Disks are available in, and their corresponding availability options.

> **Note**
>
> If a region in the following list lacks availability zones that support Ultra Disks, then a VM in that region must be deployed without infrastructure redundancy to attach an Ultra Disk.

Expand table

Redundancy options	Regions
Regional	Australia Central, Australia Central 2 Brazil Southeast Canada East Korea South Norway West UK West North Central US, West US US Gov Arizona, US Gov Texas
One availability zone	Brazil South Central India East Asia Germany West Central Japan West Korea Central South Central US Spain Central US Gov Virginia
Two availability zones	Indonesia Central Malaysia West New Zealand North Qatar Central

Redundancy options	Regions
Three availability zones	Australia East Canada Central China North 3 North Europe, West Europe France Central Italy North Japan East Poland Central South Africa North Southeast Asia Sweden Central Switzerland North UAE North UK South Central US, East US, East US 2, West US 2, West US 3

Not every VM size is available in every supported region with Ultra Disks. The following table lists VM series that are compatible with Ultra Disks.

Expand table

VM Type	Sizes	Description
General purpose	DSv3-series, Ddsv4-series, Dsv4-series, Dasv4-series, Dsv5-series, Ddsv5-series, Dasv5-series, Dplsv6-series, Dpldsv6-series, Dpsv6-series, Dpdsv6-series	Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute optimized	FSv2-series, Famsv6-series	High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	ESv3-series, Easv4-series, Edsv4-series, Esv4-series, Esv5-series, Edsv5-series, Easv5-series, Ebsv5 series, Ebdsv5 series, M-series, Mv2-series, Msv2, Mdsv2-series, Mbsv3, Mbdsv3-series, Epsv6-series, Epdsv6-series	High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	LSv2-series, Lsv3-series, Lasv3-series	High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases.
GPU optimized	NCv3-series, NCasT4_v3-series, ND-series, NDv2-series, NVv3-series, NVv4-series, NVadsA10 v5-series	Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with deep learning. Available with single or multiple GPUs.
Performance optimized	HC-series, HBv2-series	The fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

If you would like to start using Ultra Disks, see the article on using Azure Ultra Disks.

Premium SSD v2

Premium SSD v2 offers higher performance than Premium SSDs while also generally being less costly. You can individually tweak the performance (capacity, throughput, and IOPS) of Premium SSD v2 disks at any time, allowing workloads to be cost efficient while meeting shifting performance needs. For example, a transaction-intensive database may need a large amount of IOPS at a small size, or a gaming application may need a large amount of IOPS but

only during peak hours. Because of this, for most general purpose workloads, Premium SSD v2 can provide the best price performance.

Premium SSD v2 is suited for a broad range of workloads such as SQL server, Oracle, MariaDB, SAP, Cassandra, Mongo DB, big data/analytics, and gaming, on virtual machines or stateful containers.

Premium SSD v2 support a 4k physical sector size by default, but can be configured to use a 512E sector size as well. While most applications are compatible with 4k sector sizes, some require 512 byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks.

Differences between Premium SSD and Premium SSD v2

Unlike Premium SSDs, Premium SSD v2 doesn't have dedicated sizes. You can set a Premium SSD v2 to any supported size you prefer, and make granular adjustments to the performance without downtime. Premium SSD v2 doesn't support host caching but, benefits significantly from lower latency, which addresses some of the same core problems host caching addresses. The ability to adjust IOPS, throughput, and size at any time also means you can avoid the maintenance overhead of having to stripe disks to meet your needs.

Premium SSD v2 limitations

- Premium SSD v2 disks can't be used as an OS disk or with Azure Compute Gallery.
- For regions that support availability zones, Premium SSD v2 disks can only be attached to zonal VMs. When creating a new VM, specify the availability zone you want before adding Premium SSD v2 disks to your configuration.
- Premium SSD v2 doesn't support host caching.

Regional availability

Currently only available in the following regions:

Expand table

Regions without Availability Zone	*Two Availability Zones*	*Three Availability Zones*
- Australia Central 2 - Australia Southeast - Brazil Southeast - Canada East - Germany North - North Central US - Norway West - South India - Switzerland West - Taiwan North - UK West - US Gov Arizona - West Central US - West US	- Japan West	- Austria East - Australia East - Brazil South - Canada Central - Central India - Central US - China North 3 - East Asia - East US - East US 2 - France Central - Germany West Central - Indonesia Central - Israel Central - Italy North - Japan East - Korea Central - Malaysia West - Mexico Central - New Zealand North - North Europe - Norway East

Regions without Availability Zone	*Two Availability Zones*	*Three Availability Zones*
		- Poland Central - Spain Central - South African North - South Central US - Southeast Asia - Sweden Central - Switzerland North - UAE North - UK South - US Gov Virginia - West Europe - West US 2 - West US 3

To learn when support for particular regions was added, see either Azure Updates or What's new for Azure Disk Storage.

Premium SSD v2 performance

Premium SSD v2 disks are designed to provide provisioned IOPS and throughput 99.9% of the time. Premium SSD v2 disks are also designed to provide sub millisecond latencies. With Premium SSD v2 disks, you can individually set the capacity, throughput, and IOPS of a disk based on your workload needs, providing you with more flexibility and reduced costs. Each of these values determines the cost of your disk. You can adjust the performance of a Premium SSD v2 four times within a 24 hour period. Creating a disk counts as one of these times, so for the first 24 hours after creating a Premium SSD v2 you can only adjust its performance up to three times.

Premium SSD v2 capacities

Premium SSD v2 capacities range from 1 GiB to 64 TiBs, in 1-GiB increments. You're billed on a per GiB ratio, see the pricing page for details.

Premium SSD v2 offers up to 100 TiB per region per subscription by default, but supports higher capacity by request. To request an increase in capacity, request a quota increase or contact Azure Support.

Premium SSD v2 IOPS

All Premium SSD v2 disks have a baseline IOPS of 3000 that is free of charge. After 6 GiB, the maximum IOPS a disk can have increases at a rate of 500 per GiB, up to 80,000 IOPS. So an 8 GiB disk can have up to 4,000 IOPS, and a 10 GiB can have up to 5,000 IOPS. To be able to set 80,000 IOPS on a disk, that disk must have at least 160 GiBs. Increasing your IOPS beyond 3000 increases the price of your disk.

Premium SSD v2 throughput

All Premium SSD v2 disks have a baseline throughput of 125 MB/s that is free of charge. After 6 GiB, the maximum throughput that can be set increases by 0.25 MB/s per set IOPS. If a disk has 3,000 IOPS, the max throughput it can set is 750 MB/s. To raise the throughput for this disk beyond 750 MB/s, its IOPS must be increased. For example, if you increased the IOPS to 4,000, then the max throughput that can be set is 1,000. 1,200 MB/s is the maximum throughput supported for disks that have 5,000 IOPS or more. Increasing your throughput beyond 125 increases the price of your disk.

Premium SSD v2 Sector Sizes

Premium SSD v2 supports a 4k physical sector size by default. A 512E sector size is also supported. While most applications are compatible with 4k sector sizes, some require 512-byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks.

Summary

The following table provides an overview of disk capacities and performance maximums to help you decide which to use.

Expand table

Disk Size	Maximum available IOPS	Maximum available throughput (MB/s)
1 GiB-64 TiBs	3,000-80,000 (Increases by 500 IOPS per GiB)	125-1,200 (increases by 0.25 MB/s per set IOPS)

To deploy a Premium SSD v2, see Deploy a Premium SSD v2.

Premium SSDs

Azure Premium SSDs deliver high-performance and low-latency disk support for virtual machines (VMs) with input/output (IO)-intensive workloads. To take advantage of the speed and performance of Premium SSDs, you can migrate existing VM disks to Premium SSDs. Premium SSDs are suitable for mission-critical production applications, but you can use them only with compatible VM series. Premium SSDs support the 512E sector size.

To learn more about individual Azure VM types and sizes for Windows or Linux, including size compatibility for premium storage, see Sizes for virtual machines in Azure. You'll need to check each individual VM size article to determine if it's premium storage-compatible.

Premium SSD size

Expand table

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P7
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,
**Expanded provisioned IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8,000	16,000	20,000	20,000	20,
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	75(
**Expanded provisioned throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	90(

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P7
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,
Max burst throughput per disk	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,0 MB
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Un
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes on

*Applies only to disks with on-demand bursting enabled.
** Only applies to disks with performance plus enabled.

Capacity, IOPS, and throughput are guaranteed when a premium storage disk is provisioned. For example, if you create a P50 disk, Azure provisions 4,095-GB storage capacity, 7,500 IOPS, and 250-MB/s throughput for that disk. Your application can use all or part of the capacity and performance. Premium SSDs are designed to provide the single-digit millisecond latencies, target IOPS, and throughput described in the preceding table 99.9% of the time.

Premium SSD bursting

Premium SSDs offer disk bursting, which provides better tolerance on unpredictable changes of IO patterns. Disk bursting is especially useful during OS disk boot and for applications with spiky traffic. To learn more about how bursting for Azure disks works, see Disk-level bursting.

Premium SSD transactions

For Premium SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB.

Standard SSDs

Azure Standard SSDs are optimized for workloads that need consistent performance at lower IOPS levels. They're an especially good choice for customers with varying workloads supported by on-premises hard disk drive (HDD) solutions. Compared to Standard HDDs, Standard SSDs deliver better availability, consistency, reliability, and latency. Standard SSDs are suitable for web servers, low IOPS application servers, lightly used enterprise applications, and non-production workloads. Like Standard HDDs, Standard SSDs are available on all Azure VMs. Standard SSDs support the 512E sector size.

Standard SSD size

Expand table

Standard SSD sizes	E1	E2	E3	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to	Up to	Up to	Up to	Up to	Up to	Up to	Up to	Up to	Up to	Up to	Up to	Up to 4,000	Up to 6,000

Standard SSD sizes	E1	E2	E3	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
	500	500	500	500	500	500	500	500	500	500	500	2,000		
*Expanded IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Up to 1,500	Up to 3,000	Up to 6,000	Up to 6,000	Up to 6,000	Up to 6,000
Base throughput per disk	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 400 MB/s	Up to 600 MB/s	Up to 750 MB/s
*Expanded throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 600 MB/s	Up to 750 MB/s	Up to 750 MB/s	Up to 750 MB/s
Max burst IOPS per disk	600	600	600	600	600	600	600	600	1000					
Max burst throughput per disk	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	250 MB/s					
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min					

* Only applies to disks with performance plus enabled.

Standard SSDs are designed to provide single-digit millisecond latencies and the IOPS and throughput up to the limits described in the preceding table 99% of the time. Actual IOPS and throughput may vary sometimes depending on the traffic patterns. Standard SSDs provide more consistent performance than the HDD disks with the lower latency.

Standard SSD transactions

For Standard SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB. These transactions incur a billable cost but, there's an hourly limit on the number of transactions that can incur a billable cost. If that hourly limit is reached, additional transactions during that hour no longer incur a cost. For details, see the blog post.

Standard SSD Bursting

Standard SSDs offer disk bursting, which provides better tolerance for the unpredictable IO pattern changes. OS boot disks and applications prone to traffic spikes will both benefit from disk bursting. To learn more about how bursting for Azure disks works, see Disk-level bursting.

Standard HDDs

Azure Standard HDDs are suitable for latency-tolerant workloads using large, sequential IO, such as light data processing workloads that read or write large files sequentially. Standard HDDs are not suitable for workloads requiring low latency, or frequent small-block access - such as OS disks or transactional databases. Use either Standard SSDs or Premium SSD managed disks for these workloads. Standard HDDs are designed to deliver write latencies of less than 10 ms and read latencies of less than 20 ms for most IO operations. Performance may vary more widely than that of SSD-based disks, and actual performance may vary depending

on IO size and workload pattern. Standard HDDs are available in all Azure regions and can be used with all Azure VMs. Standard HDDs support the 512E sector size.

Standard HDD size

Expand table

Standard Disk Type	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000
*Expanded IOPS per disk	N/A	N/A	N/A	N/A	N/A	Up to 1,500	Up to 3,000	Up to 3,000	Up to 3,000	Up to 3,000	Up to 3,000
Base throughput per disk	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 300 MB/s	Up to 500 MB/s	Up to 500 MB/s
*Expanded throughput per disk	N/A	N/A	N/A	N/A	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 500 MB/s	Up to 500 MB/s	Up to 500 MB/s	Up to 500 MB/s

* Only applies to disks with performance plus enabled.

Standard HDD Transactions

Standard HDD transactions incur a billable cost for every 10,000 disk operations.

Billing

When using managed disks, the following billing considerations apply:

- Disk type
- Managed disk Size
- Snapshots
- Outbound data transfers
- Number of transactions

Managed disk size: managed disks are billed according to their provisioned size. Azure maps the provisioned size (rounded up) to the nearest offered disk size. For details of the disk sizes offered, see the previous tables. Each disk maps to a supported provisioned disk-size offering and is billed accordingly. For example, if you provisioned a 200-GiB Standard SSD, it maps to the disk size offer of E15 (256 GiB). Billing for any provisioned disk is prorated hourly by using the monthly price for the storage offering. For example, you provision an E10 disk and delete it after 20 hours of use. In this case, you're billed for the E10 offering prorated to 20 hours, regardless of the amount of data written to the disk.

Snapshots: Snapshots are billed based on the size used. For example, you create a snapshot of a managed disk with provisioned capacity of 64 GiB and actual used data size of 10 GiB. In this case, the snapshot is billed only for the used data size of 10 GiB.

For more information on snapshots, see Create a snapshot of a virtual hard disk.

Outbound data transfers: Outbound data transfers (data going out of Azure data centers) incur billing for bandwidth usage.

Transactions: You're billed for the number of transactions performed on a standard managed disk. For Standard SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB. Standard HDD transactions incur a billable cost for every 10,000 disk operations.

For detailed information on pricing for managed disks (including transaction costs), see managed disk pricing.

Ultra Disks VM reservation fee

Azure VMs have the capability to indicate if they're compatible with Ultra Disks. An Ultra Disk-compatible VM allocates dedicated bandwidth capacity between the compute VM instance and the block storage scale unit to optimize the performance and reduce latency. When you add this capability on the VM, it results in a reservation charge. The reservation charge is only imposed if you enabled Ultra Disk capability on the VM without an attached Ultra Disk. When an Ultra Disk is attached to the Ultra Disk compatible VM, the reservation charge wouldn't be applied. This charge is per vCPU provisioned on the VM.

> **Note**
>
> For **constrained core VM sizes**, the reservation fee is based on the actual number of vCPUs and not the constrained cores. For Standard_E32-8s_v3, the reservation fee will be based on 32 cores.

For Ultra Disk pricing details, see the Azure Disks pricing page.

Azure disk reservation

Disk reservation provides you with a discount on the advance purchase of one year's of disk storage, reducing your total cost. When you purchase a disk reservation, you select a specific disk SKU in a target region. For example, you may choose five P30 (1 TiB) Premium SSDs in the Central US region for a one year term. The disk reservation experience is similar to Azure reserved VM instances. You can bundle VM and Disk reservations to maximize your savings. For now, Azure Disks Reservation offers one year commitment plan for Premium SSD SKUs from P30 (1 TiB) to P80 (32 TiB) in all production regions. For more information about reserved disks pricing, see Azure Disks pricing page.

Next steps

- Learn more about how managed disks are billed, see Understand Azure Disk Storage billing.
- Learn the Best practices for achieving high availability with Azure virtual machines and managed disks.
- Learn about the Performance options for managed disks.

Last updated on 04/09/2026

Understand Azure Disk Storage billing

This article helps you understand how Azure managed disks are billed and how billing is laid out in your Azure Disk Storage bill. Some disks have unique attributes that affect their billing, but most disk types have the same set of attributes and are affected differently by these attributes depending on the disk type. You can also take snapshots of your disks, which are reflected in your bill.

For detailed Azure Disk Storage pricing information, see Azure Disks pricing page.

Snapshots

There are two kinds of snapshots offered for Azure managed disks: Full snapshots and incremental snapshots. Full snapshots can be stored on standard hard disk drives (HDDs) or premium solid-state drives (SSDs) while incremental snapshots are only stored on Standard HDDs. With snapshots, you're billed based on the used size of data. So if you take a full snapshot of a disk with 500-GiB size but only 50 GiB of that size is being used, then your snapshot is billed only for the used size of 50 GiB. Incremental snapshots are more cost efficient than full snapshots, as each snapshot you take only consists of the differences since the last snapshot.

Managed disk snapshots have two redundancy options, locally redundant storage (LRS) and zone-redundant storage (ZRS). For snapshots, the pricing for each redundancy option is the same.

Ultra Disks

The price of an Azure Ultra Disk is determined by the combination of how large the disk is (its disk size) and what performance you select (IOPS and throughput) for your disk. If you share an Ultra Disk between multiple VMs that can affect its price as well. The following sections focus on these factors as they relate to the price of your Ultra Disk. For more information on how these factors work, see the Ultra Disks section of the Azure managed disk types article.

Ultra Disk size

Ultra Disk capacities range from 4 GiB to 64 TiBs, in 1-GiB increments. You're billed on a per GiB ratio.

Ultra Disk IOPS

Pricing of an Azure Ultra Disk increases as you provision more IOPS to your disk. Ultra Disks supports 1000 IOPS/GiB, up to a maximum of 400,000 IOPS per disk.

Ultra Disk throughput

Pricing of an Ultra Disk increases as you increase the disk's throughput limit. Ultra Disk supports .25 MB/s for each provisioned IOPS, up to a maximum of 10,000 MB/s per disk (where MB/s = 10^6 Bytes per second). The minimum guaranteed throughput of an Ultra Disk is 1 MB/s.

Shared Ultra Disks

Ultra Disks can be used as shared disks, where you attach one disk to multiple VMs. For Ultra Disks, there isn't an extra charge for each VM that the disk is mounted to. Ultra Disks that are shared are billed on the total IOPS and MB/s that the disk is configured for. Normally, an Ultra Disk has two performance throttles that determine its total IOPS/MB/s. However, when configured as a shared Ultra Disk, two more performance throttles are exposed, for a total of four. These two extra throttles allow for increased performance at an extra expense and each meter has a default value, which raises the performance and cost of the disk. For more information, see Share an Azure managed disk.

Ultra Disk billing example

In this example, we provisioned an Ultra Disk with LRS redundancy with a total provisioned capacity of 3 TiB, a target performance of 100,000 IOPS and 2,000 MB/s of throughput. We also created and stored incremental snapshots for our used capacity. We're billed for the provisioned capacity of the disk, the extra IOPS and throughput past the baseline values, and the used snapshot capacity that shows as the following tier and meters in our bill:

Expand table

Tier	Meter
Ultra Disks	Ultra LRS provisioned capacity
Ultra Disks	Ultra LRS provisioned IOPS
Ultra Disks	Ultra LRS provisioned throughput (MB/s)

Tier	Meter
Standard HDD managed disks	ZRS snapshots

Premium SSD v2

The price of an Azure Premium SSD v2 is determined by the combination of how large the disk is (its capacity) and what performance you select (IOPS and throughput) for your disk. If you share a Premium SSD v2 between multiple VMs that can affect its price as well. The following sections focus on these factors as they relate to the price of your Premium SSD v2. For more information on how these factors work, see the Premium SSD v2 section of the Azure managed disk types article.

Premium SSD v2 capacities

Premium SSD v2 capacities range from 1 GiB to 64 TiBs, in 1-GiB increments. You're billed on a per GiB ratio. See the pricing page for details.

Premium SSD v2 IOPS

All Premium SSD v2 disks have a baseline IOPS of 3,000 that is free of charge. After 6 GiB, the maximum IOPS a disk can have increases at a rate of 500 per GiB, up to 80,000 IOPS. So an 8-GiB disk can have up to 4,000 IOPS, and a 10 GiB can have up to 5,000 IOPS. To set 80,000 IOPS on a disk, that disk must have at least 160 GiBs. Increasing your IOPS beyond 3,000 increases the price of your disk.

Premium SSD v2 throughput

All Premium SSD v2 disks have a baseline throughput of 125 MB/s that is free of charge. After 6 GiB, the maximum throughput that can be provisioned increases by 0.25 MB/s per provisioned IOPS. If a disk has 3,000 IOPS, the max throughput it can set is 750 MB/s. To raise the throughput for this disk beyond 750 MB/s, its IOPS must be increased. For example, if you increased the IOPS to 4,000, then the max throughput that can be set is 1,000. 1,200 MB/s is the maximum throughput supported for disks that have 5,000 IOPS or more. Increasing your provisioned throughput beyond 125 MB/s increases the price of your disk.

Shared Premium SSD v2

Premium SSD v2 managed disks can be used as shared disks, where you attach one disk to multiple VMs. For Premium SSD v2 disks there isn't an extra charge for each VM that the disk is mounted to. Premium SSD v2 disks that are shared are billed on the total IOPS and MB/s that the disk is configured for. Normally, a Premium SSD v2 has two performance throttles that determine its total IOPS/MB/s. However, when configured as a shared Premium SSD v2, two more performance throttles are exposed, for a total of four. These two extra throttles allow for increased performance at an extra expense and each meter has a default value, which raises the performance and cost of the disk. For more information, see Share an Azure managed disk.

Premium SSD v2 billing example

In this example, we provision a Premium SSD v2 with LRS redundancy with a total provisioned capacity of 512 GiB, a target performance of 40,000 IOPS and 200 MB/s of throughput. We also create and store incremental snapshots for our current used capacity. We're billed for the provisioned capacity of the disk, the IOPS and throughput past the baseline values, and the used snapshot capacity that show as the following tier and meters in our bill:

Expand table

Tier	Meter
Azure Premium SSD v2	Premium LRS provisioned capacity
Azure Premium SSD v2	Premium LRS provisioned IOPS
Azure Premium SSD v2	Premium LRS provisioned throughput (MB/s)
Standard HDD managed disks	LRS snapshots

Premium SSD

The price of an Azure Premium SSD is determined by the performance tier of the disk, whether bursting is enabled, what redundancy options you select, and whether or not you share the disk between multiple VMs. The following sections focus on these factors as they relate to the price of your Premium SSD. For more information on how these factors work, see the Premium SSDs section of the Azure managed disk types article.

Performance tier

The initial billing of Premium SSDs is determined by the performance tier of the disk. Generally, the performance tier is set when you select the capacity you require (if you deploy a 1 TiB

Premium SSD, it has the P30 tier by default) but certain disk sizes can select higher performance tiers. When you select a higher performance tier, your disk is billed at that tier until you change its performance tier again. To learn more about performance tiers, see Performance tiers for managed disks.

Premium SSD bursting

Premium SSDs offer two bursting models, credit-based bursting and on-demand bursting. Only on-demand bursting has billing impact and you must explicitly enable on-demand bursting. Premium SSD managed disks using on-demand bursting are charged an hourly burst enablement flat fee, and transaction costs apply to any burst transactions beyond the provisioned target. Transaction costs are charged using a pay-as-you go model, based on uncached disk IOs, including reads and writes that exceed provisioned targets.

Premium SSD transactions

For Premium SSD managed disks, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB. Unless you enable on-demand bursting, there are no transaction costs for Premium SSDs.

Redundancy options

Premium SSD managed disks can be deployed either with locally redundant storage (LRS) or zone-redundant storage (ZRS). The redundancy you select for your disk changes its pricing. For details see the Azure pricing page.

Shared Premium SSD

Premium SSD managed disks can be used as shared disks, where you attach one disk to multiple VMs. For shared Premium SSDs, there's a charge that increases with each VM the SSD is mounted to. See managed disks pricing for details.

Premium SSD billing example

In this example, we provision a Premium SSD at 512 GiB with LRS redundancy with bursting enabled.

We're billed for the provisioned capacity of the Premium SSD, the burst enablement flat fee, and transaction costs apply to any burst transactions beyond the provisioned target that show

as the following tier and meters in our bill:

Expand table

Tier	Meter
Premium SSD managed disks	P20 LRS Disk
Premium SSD managed disks	LRS Burst Enablement*
Premium SSD managed disks	LRS Burst Transactions*

*To see a more detailed example of how bursting is billed, see Disk-level bursting.

Standard SSD

The price of an Azure Standard SSD is determined by the performance tier of the disk, number of transactions, what redundancy options you select, and whether or not you share the disk between multiple VMs. The following sections focus on these factors as they relate to the price of your Standard SSD. For more information on how these factors work, see the Standard SSDs section of the Azure managed disk types article.

Performance tier

The initial billing of Standard SSDs is determined by the performance tier. The performance tier is set when you select the capacity you require (if you deploy a 1 TiB Standard SSD, it has the E30 tier), your disk is billed at that tier. If you increase the capacity of your disk into the next tier, it's then billed at that tier. For example, if you increased your 1-TiB disk to a 3-TiB disk, it's billed at the E50 tier.

Standard SSD transactions

For Standard SSDs, each I/O operation less than or equal to 256 kB of throughput is considered a single I/O operation. I/O operations larger than 256 kB of throughput are considered multiple I/Os of size 256 kB. These transactions incur a billable cost but, there's an hourly limit on the number of transactions that can incur a billable cost. If that hourly limit is reached, extra transactions during that hour no longer incur a cost. For details, see the blog post.

Redundancy options

Standard SSDs can be deployed either with locally redundant storage (LRS) or zone-redundant storage (ZRS). The redundancy you select for your disk changes its pricing. For details see the Azure pricing page.

Shared Standard SSDs

Standard SSDs can be used as shared disks, where you attach one disk to multiple VMs. For shared Standard SSDs, there's a charge that increases with each VM the SSD is mounted to. See managed disks pricing for details.

Standard SSD billing example

In this example, we provision a 1 TiB Standard SSD with LRS redundancy, where we also have snapshots created on the current used data capacity of 120 GiB. You're billed for the provisioned capacity of the SSD disk, the transactions performed on the disk, and the used snapshot capacity that will show as the following tier and meters in our bill:

Expand table

Tier	Meter
Standard SSD managed disks	E30 LRS Disk
Standard SSD managed disks	E4 LRS Disk Operations
Standard HDD managed disks	LRS snapshots

Standard HDD

The price of an Azure Standard HDD is determined by the performance tier of the disk and the number of transactions. The following sections focus on these factors as they relate to the price of your Standard HDD. For more information on how these factors work, see the Standard HDDs section of the Azure managed disk types article.

Performance tier

The initial billing of Standard HDDs is determined by the performance tier. The performance tier is set when you select the capacity you require (if you deploy a 1 TiB Standard HDD, it has the S30 tier), your disk is billed at that tier. If you increase the capacity of your disk into the next tier, it's billed at that tier. For example, if you increased your 1-TiB disk to a 3-TiB disk, it's billed at the S50 tier.

Standard HDD Transactions

Standard HDD transactions incur a billable cost for every 10,000 disk operations.

Standard HDD billing example

In this example, we provision a 512 GiB Standard HDD with LRS redundancy. We're billed for the provisioned capacity of the HDD disk and the transactions performed on the disk, which shows as the following tier and meters in our bill:

Expand table

Tier	Meter
Standard HDD managed disks	S20 LRS Disk
Standard HDD managed disks	S4 LRS Disk Operations

See also

- Azure managed disks pricing page
- Shared disk billing implications

Last updated on 04/09/2026

Redundancy options for managed disks

07/02/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure managed disks offer two storage redundancy options, zone-redundant storage (ZRS), and locally redundant storage. ZRS provides higher availability for managed disks than locally redundant storage (LRS) does. However, the write latency for LRS disks is better than ZRS disks because LRS disks synchronously write data to three copies in a single data center.

Locally redundant storage for managed disks

Locally redundant storage (LRS) replicates your data three times within a single data center in the selected region. LRS protects your data against server rack and drive failures. LRS disks provide at least 99.999999999% (11 9's) of durability over a given year. To protect an LRS disk from a zonal failure like a natural disaster or other issues, take the following steps:

- Use applications that can synchronously write data to two zones, and automatically failover to another zone during a disaster.
  - An example would be SQL Server Always On.
- Take frequent backups of LRS disks with ZRS snapshots.
- Enable cross-zone disaster recovery for LRS disks via Azure Site Recovery. However, cross-zone disaster recovery doesn't provide zero Recovery Point Objective (RPO).

If your workflow doesn't support application-level synchronous writes across zones, or your application must meet zero RPO, then ZRS disks would ideal.

Zone-redundant storage for managed disks

Zone-redundant storage (ZRS) synchronously replicates your Azure managed disk across three Azure availability zones in the region you select. Each availability zone is a separate physical location with independent power, cooling, and networking. ZRS disks provide at least 99.9999999999% (12 9's) of durability over a given year.

A ZRS disk lets you recover from failures in availability zones. If a zone went down and your virtual machine (VM) wasn't affected, then your workloads continue running. But if your VM was affected by an outage and you want to recover before it's resolved, you can force detach your ZRS disks from the failed VM, freeing them to attach to another VM.

ZRS disks can also be shared between VMs for improved availability with clustered or distributed applications like SQL FCI, SAP ASCS/SCS, or GFS2. A shared ZRS disk can be

attached to primary and secondary VMs in different zones to take advantage of both ZRS and availability zones. If your primary zone fails, you can quickly fail over to the secondary VM using SCSI persistent reservation.

For more information on ZRS disks, see Zone Redundant Storage (ZRS) option for Azure Disks for high availability.

Limitations

ZRS for managed disks is only supported with Premium SSD and Standard SSD managed disks. ZRS for managed disks isn't supported with Premium SSD v2 managed disks or Ultra Disks. Force detach only supports data disks.

Regional availability

ZRS disks are available in all regions that have availability zone support. For a list of these regions, see Azure regions list.

Billing implications

For details see the Azure pricing page.

Comparison with other disk types

Except for more write latency, disks using ZRS are identical to disks using LRS, they have the same scale targets. Benchmark your disks to simulate the workload of your application and compare the latency between LRS and ZRS disks.

Next steps

- To learn how to create a ZRS disk, see Deploy a ZRS managed disk.
- To convert an LRS disk to ZRS, see Convert a disk from LRS to ZRS.

What's new for Azure Disk Storage

Summarize this article for me

Azure Disk Storage regularly receives updates for new features and enhancements. This article provides information about what's new in Azure Disk Storage.

Update summary

- What's new in 2026
  - Quarter 1 (January, February, March)
    - Expanded regional availability for Premium SSD v2
- What's new in 2025
  - Quarter 4 (October, November, December)
    - Expanded regional availability for Premium SSD v2
    - Public preview: Instant Access Snapshot for Premium SSD v2 and Ultra Disks
    - Generally available: Azure Site Recovery for Virtual Machines with Premium SSD v2 and Ultra Disks
  - Quarter 3 (July, August, September)
    - Generally available: Live Resize for Premium SSD v2 and Ultra Disks using NVMe controllers
    - Generally available: Ultra Disk price reduction in West US 2, Central US, and UK South
  - Quarter 2 (April, May, June)
    - Generally available: Troubleshoot Disk Performance with Copilot in Azure
    - Generally available: Availability Set support for Premium SSD v2 Storage
    - Expanded regional availability for Premium SSD v2
  - Quarter 1 (January, February, March)
    - Public Preview: Troubleshoot Disk Performance with Copilot in Azure
- What's new in 2024
  - Quarter 4 (October, November, December)
    - Generally available: Convert existing disks to Premium SSD v2 disks
    - Generally available: Expand Ultra Disks and Premium SSD v2 without downtime
    - Expanded regional availability for Premium SSD v2
  - Quarter 2 (April, May, June)
    - Generally available: New property for disks - LastOwnershipUpdateTime
  - Quarter 1 (January, February, March)
    - Generally available: Azure VM Backup support for Ultra Disks and Premium SSD v2
    - Generally available: Trusted launch support for Ultra Disks and Premium SSD v2
    - Expanded regional availability for Ultra Disks
    - Expanded regional availability for zone-redundant storage disks

What's new in 2026

Quarter 1 (January, February, March)

Expanded regional availability for Premium SSD v2

Premium SSD v2 disks are now available in Brazil Southeast, Germany North, Switzerland West and in a third Availability Zone in Indonesia Central, Malaysia West and New Zealand North.

What's new in 2025

Quarter 4 (October, November, December)

Expanded regional availability for Premium SSD v2

Premium SSD v2 disks are now available in Austria East and in a second Availability Zone in Japan West.

Public preview: Instant Access Snapshot for Premium SSD v2 and Ultra Disks

By using Instant Access Snapshots for Premium SSD v2 and Ultra Disks, you can restore new disks right after creating snapshots of Premium SSD v2 and Ultra Disks. Restore disks deliver high performance instantly, while data hydration continues rapidly in the background. For more information, see instant access for Azure Managed Disk.

Generally available: Azure Site Recovery for Virtual Machines with Premium SSD v2 and Ultra Disks

Azure Site Recovery support for Virtual Machines with Premium SSD v2 and Ultra disks is generally available. Azure Site Recovery provides seamless disaster recovery for Virtual Machines across Azure Regions and from on-premises to Azure, helping organizations maintain business continuity. It offers cost-effective replication, automated failover, and easy disaster recovery simulation, ensuring minimal production impact during disaster events. Learn more.

Quarter 3 (July, August, September)

Generally available: Live Resize for Premium SSD v2 and Ultra Disks using NVMe controllers

You can dynamically expand the storage capacity of your Premium SSD v2 and Ultra Disks using NVMe controllers without any disruption to your applications. To optimize costs, you can start with smaller disks and gradually increase their storage capacity as needed, without experiencing downtime. For more information, see expand with Ultra Disks and Premium SSD v2.

Generally available: Ultra Disk price reduction in West US 2, Central US, and UK South

To better support performance sensitive and mission critical workloads on Azure, Ultra Disk offers improved cost efficiency in West US 2, Central US, and UK South. These enhancements give you access to the same high-performance storage at a lower cost. For details, see West US 2, Central US, and UK South.

Quarter 2 (April, May, June)

Public preview: Azure Site Recovery for Virtual Machines with Premium SSD v2 and Ultra Disks

Azure Site Recovery support for Virtual Machines with Premium SSD v2 and Ultra disks is in public preview. Azure Site Recovery provides seamless disaster recovery for Virtual Machines across Azure Regions and from on-premises to Azure, helping organizations maintain business continuity. It offers cost-effective replication, automated failover, and easy disaster recovery simulation, ensuring minimal production impact during disaster events. With built-in security, compliance support, and native integration with Azure services, Azure Site Recovery helps your organization stay resilient and minimize downtime. Learn more.

Generally available: Troubleshoot Disk Performance with Copilot in Azure

The Disk Performance Troubleshooting Capability for Copilot in Azure is now generally available as a part of the Copilot in Azure General Availability. Now, you can use Copilot in Azure to analyze your disk metrics and resolve any performance degradation issues when your application requires higher performance than what you configured for your VMs and disks. To learn more, see Troubleshoot Disk Performance using Microsoft Copilot in Azure.

Generally available: Availability Set support for Premium SSD v2 Storage

Availability Set support for Premium SSD v2 storage is now generally available. Availability Set enhances application availability by distributing virtual machines and their associated Premium SSD v2 disks across multiple fault domains, reducing the risk of a single point of failure. Premium SSD v2 provides low latency, consistent performance, flexible scalability, and cost efficiency making it an ideal choice for enterprise workloads such as SAP, SQL Server, and Oracle. The combination of Availability Set and Premium SSD v2 enables customers to achieve higher availability, performance, and cost optimization for their critical applications. For more information, see Availability Sets with Premium SSD v2 documentation.

Expanded regional availability for Premium SSD v2

Premium SSD v2 disks are available in Australia Central 2, Australia Southeast, Canada East, Indonesia Central, Japan West, Malaysia West, New Zealand North, North Central US, Norway West, UK West, US West, and West Central US.

Quarter 1 (January, February, March)

Public preview: Troubleshoot disk performance with Copilot in Azure

The Disk Performance Troubleshooting Capability for Copilot in Azure is now available in Public Preview. Now, you can use Copilot in Azure to analyze your disk metrics and resolve any performance degradation problems when your application requires higher performance than what you configured for your VMs and disks. To learn more, see Troubleshoot Disk Performance using Microsoft Copilot in Azure.

What's new in 2024

Quarter 4 (October, November, December)

Generally available: Convert existing disks to Premium SSD v2 disks

Directly converting a disk to a Premium SSD v2 is generally available. This feature makes it easier to move your workloads from Standard HDD, Standard SSD, and Premium Disks, to Premium SSD v2 disks, and take advantage of its balance of price and performance capabilities. To learn more, see Convert Premium SSD v2 disks.

Generally available: Expand Ultra Disks and Premium SSD v2 without downtime

Expanding Ultra Disks and Premium SSD v2 disks without downtime is generally available. This feature allows you to dynamically increase the capacity of your storage without causing disruptions to existing applications. To learn more, see the Windows or Linux articles.

Expanded regional availability for Premium SSD v2

Premium SSD v2 disks are available in Germany West Central, Israel Central, Italy North, Spain Central, and Mexico Central regions. For more information, see the Azure Update.

Quarter 2 (April, May, June)

Generally available: New property for disks - LastOwnershipUpdateTime

The `LastOwnershipUpdateTime` property is now available for disks in the Azure portal, Azure PowerShell module, and Azure CLI. This property reflects the time when a disk's state was last changed. This property can be used with the `diskState` to identify the current state of a disk, and when it was last updated. To learn more, see the Azure Update post or the documentation.

Quarter 1 (January, February, March)

Generally available: Azure VM Backup support for Ultra Disks and Premium SSD v2

Azure Backup enabled support on Azure VMs using Ultra Disks and Premium SSD v2 that offers high throughput, high IOPS, and low latency. Azure VM Backup support allows you to ensure business continuity for your virtual machines and to recover from any disasters or ransomware attacks. Enabling backup on VMs using Ultra Disks and Premium SSD v2 is available in all regions where Ultra Disks and Premium SSD v2 disks are supported. For more information, see the documentation and enable backup on your Azure VMs.

Generally available: Trusted launch support for Ultra Disks and Premium SSD v2

Trusted launch VMs added support for Ultra Disks and Premium SSD v2, so you can combine the foundational compute security of Trusted Launch with the high throughput, high IOPS, and low latency of Ultra Disks and Premium SSD v2. For more information, see Trusted launch for Azure virtual machines or the Azure Update.

Expanded regional availability for Ultra Disks

Ultra Disks are available in the UK West and Poland Central regions.

Expanded regional availability for zone-redundant storage disks

Zone-redundant storage (ZRS) disks are available in West US 3 and Germany Central regions.

What's new in 2023

Quarter 4 (October, November, December)

Encryption at host is generally available for Premium SSD v2 and Ultra Disks

Encryption at host was previously only available for Standard HDDs, Standard SSDs, and Premium SSDs. Encryption at host is now also available as a GA offering for Premium SSD v2 and Ultra Disks. For more information on encryption at host, see Encryption at host - End-to-end encryption for your VM data.

Some additional restrictions apply to Premium SSD v2 and Ultra Disks that enable encryption at host. For more information, see Restrictions.

New latency metrics (preview)

Metrics dedicated to monitoring latency are now available as a preview feature. To learn more, see either the metrics article or the Azure Update.

Expanded regional availability for Premium SSD v2

Premium SSD v2 disks are available in Poland Central, China North 3, and US Gov Virginia. For more information, see the Azure Update.

Expanded regional availability for ZRS disks

ZRS disks are available in the Norway East and UAE North regions. For more information, see the Azure Update.

Quarter 3 (July, August, September)

Expanded regional availability for ZRS disks

In quarter 3, ZRS disks became available in the China North 3, East Asia, India Central, Switzerland North, South Africa North, and Sweden Central regions.

Expanded regional availability for Premium SSD v2

In quarter 3, Premium SSD v2 became available in the Australia East, Brazil South, Canada Central, Central India, Central US, East Asia, France Central, Japan East, Korea Central, Norway East, South Africa North, Sweden Central, Switzerland North, and UAE North regions.

General availability - Incremental snapshots for Premium SSD v2 and Ultra Disks

Incremental snapshots for Premium SSD v2 and Ultra Disks are now generally available (GA). For more information, see either the documentation or the Azure Update.

Quarter 2 (April, May, June)

Expanded regional availability for Premium SSD v2

In quarter 2, Premium SSD v2 disks became available in the Southeast Asia, UK South, South Central US, and West US 3 regions.

Expanded regional availability for ZRS disks

In quarter 2, ZRS disks became available in the Australia East, Brazil South, Japan East, Korea Central, Qatar Central, UK South, East US, East US 2, South Central US, and Southeast Asia regions.

Azure Backup support (preview) for Premium SSD v2

Azure Backup added preview support for Azure virtual machines using Premium SSD v2 disks in the East US and West Europe regions. For more information, see the Azure Update.

Quarter 1 (January, February, March)

Expanded regional availability for Premium SSD v2

In quarter 1, Premium SSD v2 disks became available in the East US 2, North Europe, and West US 2 regions.

Preview - Performance plus

Azure Disk Storage added a new preview feature, performance plus. Performance plus enhances the IOPS and throughput performance for Premium SSDs, Standard SSDs, and Standard HDDs that are 513 GiB and larger. For details, see Increase IOPS and throughput limits for Azure Premium SSDs and Standard SSD/HDDs.

Expanded regional availability for Ultra Disks

In quarter 1, Ultra Disks became available in the Brazil Southeast, China North 3, Korea South, South Africa North, Switzerland North, and UAE North regions.

More transactions at no extra cost - Standard SSDs

In quarter 1, an hourly limit was added to the number of transactions that can occur before a billable cost. Any transactions beyond that limit don't incur a cost. For more information, see the blog post or Standard SSD transactions.

GA: Create disks from snapshots encrypted with customer-managed keys across subscriptions

In quarter 1, support was added for creating disks from snapshots or other disks encrypted with customer-managed keys in different subscriptions while within the same tenant. For more information, see either the Azure Update or the documentation.

GA: Entra ID support for managed disks

In quarter 1, support was added for using Entra ID to secure uploads and downloads of managed disks. For details, see Secure downloads and uploads of Azure Managed Disks.

Next steps

- Azure Managed Disk types
- Introduction to Azure Managed Disks

Last updated on 02/19/2026

Using Azure Ultra Disks

07/02/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article explains how to deploy and use an Ultra Disk, for conceptual information about Ultra Disks, refer to What disk types are available in Azure?.

Azure Ultra Disks offer high throughput, high IOPS, and consistent low latency disk storage for Azure IaaS virtual machines (VMs). This new offering provides top of the line performance at the same availability levels as our existing disks offerings. One major benefit of Ultra Disks is the ability to dynamically change the performance of the SSD along with your workloads without the need to restart your VMs. Ultra Disks are suited for data-intensive workloads such as SAP HANA, top tier databases, and transaction-heavy workloads.

GA scope and limitations

The following list contains Ultra Disk's limitations:

- Ultra Disks can't be used as an OS disk or with Azure Compute Gallery.
- Currently, Ultra Disks only support Single VM and Availability zone infrastructure options as locally redundant storage (LRS). Ultra Disks don't support zone redundant storage (ZRS).
- Ultra Disks don't support availability sets.
- Ultra Disks don't support disk caching.
- (Preview) You can encrypt Ultra Disks with customer-managed keys using Azure Key Vaults stored in a different Microsoft Entra ID tenant.

Ultra Disks support a 4k physical sector size by default but also supports a 512E sector size. Most applications are compatible with 4k sector sizes, but some require 512-byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks. For older versions of Oracle DB, 512-byte sector size is required.

The following table outlines the regions Ultra Disks are available in, and their corresponding availability options.

> ⓘ **Note**
>
> If a region in the following list lacks availability zones that support Ultra disks, then a VM in that region must be deployed without infrastructure redundancy to attach an Ultra Disk.

Expand table

Redundancy options	Regions
Regional	Australia Central, Australia Central 2 Brazil Southeast Canada East Korea South Norway West UK West North Central US, West US US Gov Arizona, US Gov Texas
One availability zone	Brazil South Central India East Asia Germany West Central Korea Central South Central US Spain Central US Gov Virginia
Two availability zones	Indonesia Central Malaysia West New Zealand North Qatar Central
Three availability zones	Australia East Canada Central China North 3 North Europe, West Europe France Central Italy North Japan East Poland Central South Africa North Southeast Asia Sweden Central Switzerland North UAE North UK South Central US, East US, East US 2, West US 2, West US 3

Not every VM size is available in every supported region with Ultra Disks. The following table lists VM series that are compatible with Ultra Disks.

Expand table

VM Type	Sizes	Description
General purpose	DSv3-series, Ddsv4-series, Dsv4-series, Dasv4-series, Dsv5-series, Ddsv5-series, Dasv5-series, Dplsv6-series, Dpldsv6-series, Dpsv6-series, Dpdsv6-series	Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.
Compute optimized	FSv2-series, Famsv6-series	High CPU-to-memory ratio. Good for medium traffic web servers, network appliances, batch processes, and application servers.
Memory optimized	ESv3-series, Easv4-series, Edsv4-series, Esv4-series, Esv5-series, Edsv5-series, Easv5-series, Ebsv5 series, Ebdsv5 series, M-series, Mv2-series, Msv2, Mdsv2-series, Mbsv3, Mbdsv3-series, Epsv6-series, Epdsv6-series	High memory-to-CPU ratio. Great for relational database servers, medium to large caches, and in-memory analytics.
Storage optimized	LSv2-series, Lsv3-series, Lasv3-series	High disk throughput and IO ideal for Big Data, SQL, NoSQL databases, data warehousing, and large transactional databases.
GPU optimized	NCv3-series, NCasT4_v3-series, ND-series, NDv2-series, NVv3-series, NVv4-series, NVadsA10 v5-series	Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with deep learning. Available with single or multiple GPUs.
Performance optimized	HC-series, HBv2-series	The fastest and most powerful CPU virtual machines with optional high-throughput network interfaces (RDMA).

Determine VM size and region availability

VMs using availability zones

To use Ultra Disks, you need to determine which availability zone you are in. Not every region supports every VM size with Ultra Disks. To determine if your region, zone, and VM size support Ultra Disks, run either of the following commands, make sure to replace the **region**, **vmSize**, and **subscriptionId** values first:

CLI

Azure CLI

```
subscriptionId="<yourSubID>"
# Example value is southeastasia
region="<yourLocation>"
# Example value is Standard_E64s_v3
vmSize="<yourVMSize>"

az vm list-skus --resource-type virtualMachines --location $region --query "[?
name=='$vmSize'].locationInfo[0].zoneDetails[0].Name" --subscription
$subscriptionId
```

PowerShell

PowerShell

```
# Example value is southeastasia
$region = "<yourLocation>"
# Example value is Standard_E64s_v3
$vmSize = "<yourVMSize>"
$sku = (Get-AzComputeResourceSku | where {$_.Locations -icontains($region) -and
($_.Name -eq $vmSize) -and $_.LocationInfo[0].ZoneDetails.Count -gt 0})
if($sku){$sku[0].LocationInfo[0].ZoneDetails} Else {Write-host "$vmSize is not
supported with Ultra Disk in $region region"}
```

The response will be similar to the form below, where X is the zone to use for deploying in your chosen region. X could be either 1, 2, or 3.

Preserve the **Zones** value, it represents your availability zone and you'll need it in order to deploy an Ultra disk.

Expand table

ResourceType	Name	Location	Zones	Restriction	Capability	Value
disks	UltraSSD_LRS	eastus2	X			

> **Note**
>
> If there was no response from the command, then the selected VM size is not supported with Ultra Disks in the selected region.

Now that you know which zone to deploy to, follow the deployment steps in this article to either deploy a VM with an Ultra Disk attached or attach an Ultra Disk to an existing VM.

VMs with no redundancy options

Ultra disks deployed in select regions must be deployed without any redundancy options, for now. However, not every VM size that supports Ultra Disks are necessarily in these regions. To determine which VM sizes support Ultra Disks, use either of the following code snippets. Make sure to replace the `vmSize`, `region`, and `subscriptionId` values first:

Azure CLI

```
subscriptionId="<yourSubID>"
# Example value is westus
region="<yourLocation>"
# Example value is Standard_E64s_v3
vmSize="<yourVMSize>"

az vm list-skus --resource-type virtualMachines --location $region --query "[?
name=='$vmSize'].capabilities" --subscription $subscriptionId
```

PowerShell

```
# Example value is westus
$region = "<yourLocation>"
# Example value is Standard_E64s_v3
$vmSize = "<yourVMSize>"
(Get-AzComputeResourceSku | where {$_.Locations -icontains($region) -and ($_.Name
-eq $vmSize) })[0].Capabilities
```

The response will be similar to the following form, `UltraSSDAvailable True` indicates whether the VM size supports Ultra Disks in this region.

```
Name                                        Value
----                                        -----
MaxResourceVolumeMB                         884736
OSVhdSizeMB                                 1047552
vCPUs                                       64
HyperVGenerations                           V1,V2
MemoryGB                                    432
MaxDataDiskCount                            32
LowPriorityCapable                          True
PremiumIO                                   True
VMDeploymentTypes                           IaaS
vCPUsAvailable                              64
ACUs                                        160
vCPUsPerCore                                2
CombinedTempDiskAndCachedIOPS               128000
CombinedTempDiskAndCachedReadBytesPerSecond 1073741824
CombinedTempDiskAndCachedWriteBytesPerSecond 1073741824
CachedDiskBytes                             1717986918400
UncachedDiskIOPS                            80000
```

```
UncachedDiskBytesPerSecond              1258291200
EphemeralOSDiskSupported                True
AcceleratedNetworkingEnabled            True
RdmaEnabled                             False
MaxNetworkInterfaces                    8
UltraSSDAvailable                       True
```

Deploy an Ultra Disk using Azure Resource Manager

First, determine the VM size to deploy. For a list of supported VM sizes, see the GA scope and limitations section.

If you would like to create a VM with multiple Ultra Disks, refer to the sample Create a VM with multiple Ultra Disks.

If you intend to use your own template, make sure that **apiVersion** for `Microsoft.Compute/virtualMachines` and `Microsoft.Compute/Disks` is set as `2018-06-01` (or later).

Set the disk sku to **UltraSSD_LRS**, then set the disk capacity, IOPS, availability zone, and throughput in MBps to create an Ultra Disk.

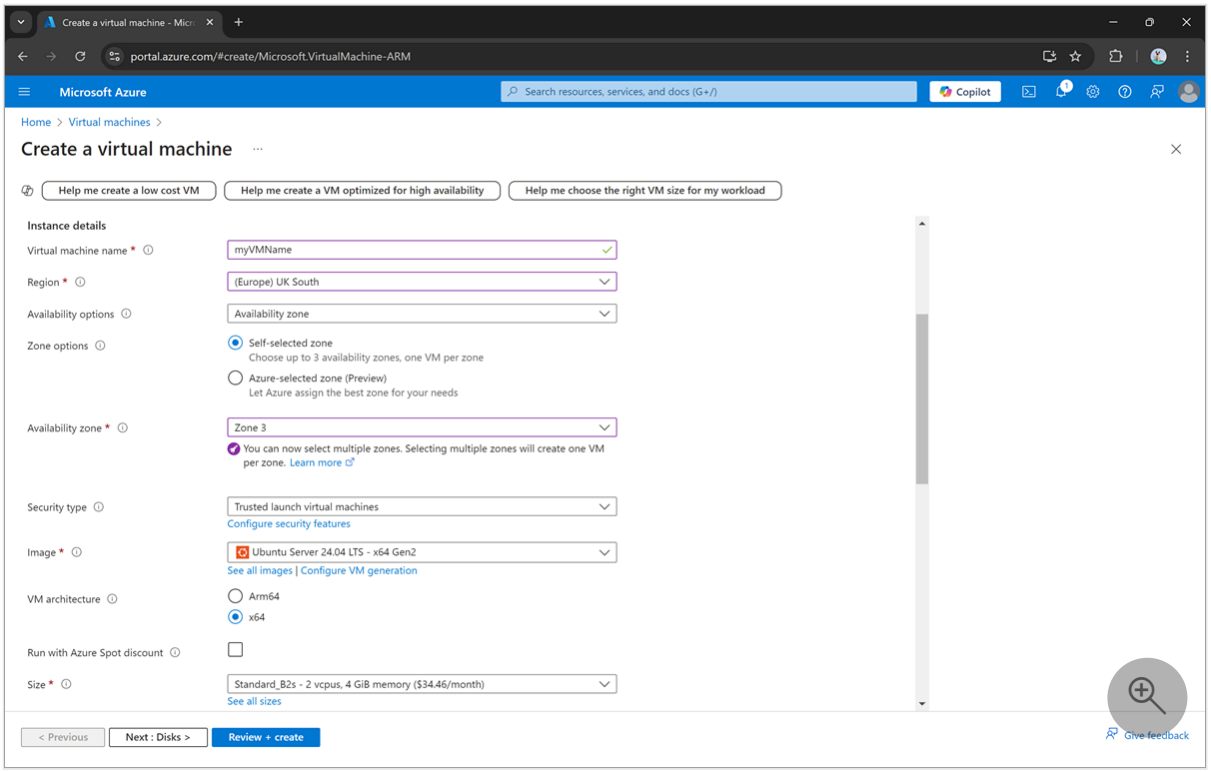
Once the VM is provisioned, you can partition and format the data disks and configure them for your workloads.

Deploy an Ultra Disk

Portal

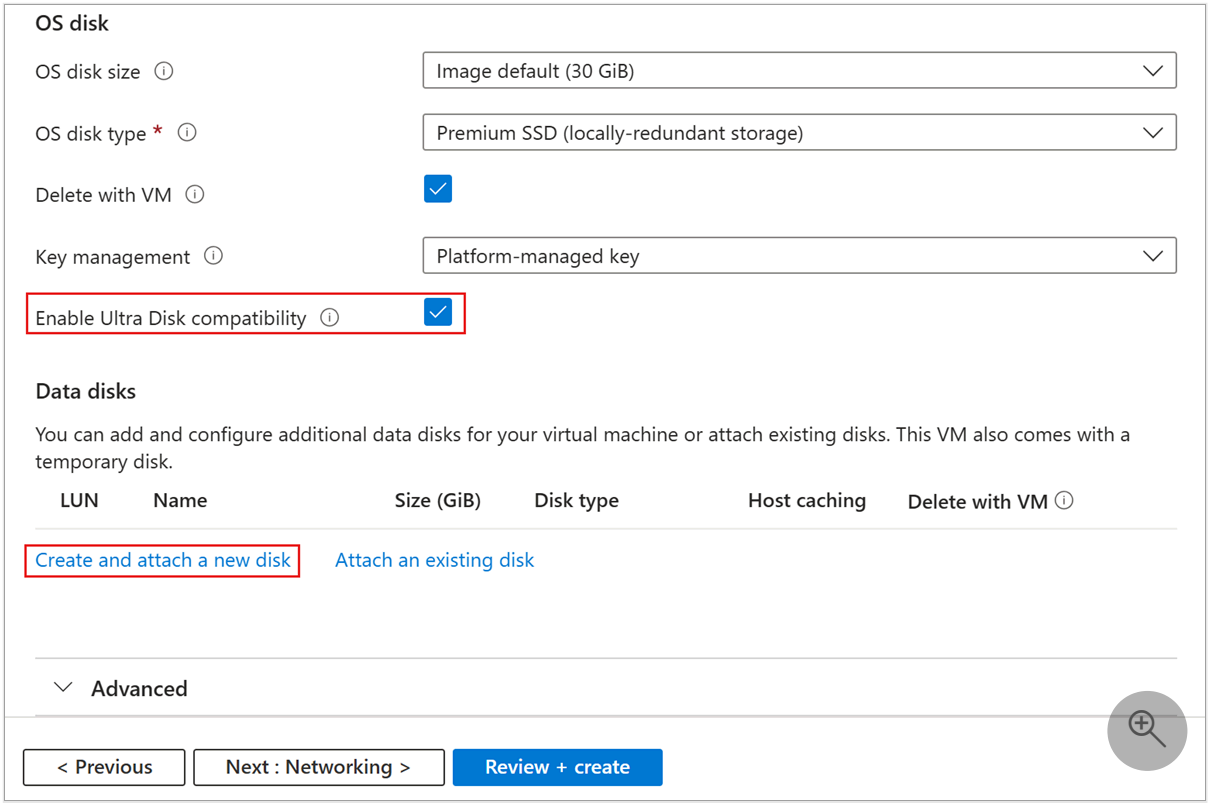
This section covers deploying a virtual machine equipped with an Ultra Disk as a data disk. It assumes you have familiarity with deploying a virtual machine, if you don't, see our Quickstart: Create a Windows virtual machine in the Azure portal.

1. Sign in to the Azure portal and navigate to deploy a virtual machine (VM).
2. Make sure to choose a supported VM size and region.
3. Select **Availability zone** in **Availability options**.
4. Fill in the remaining entries with selections of your choice.
5. Select **Disks**.

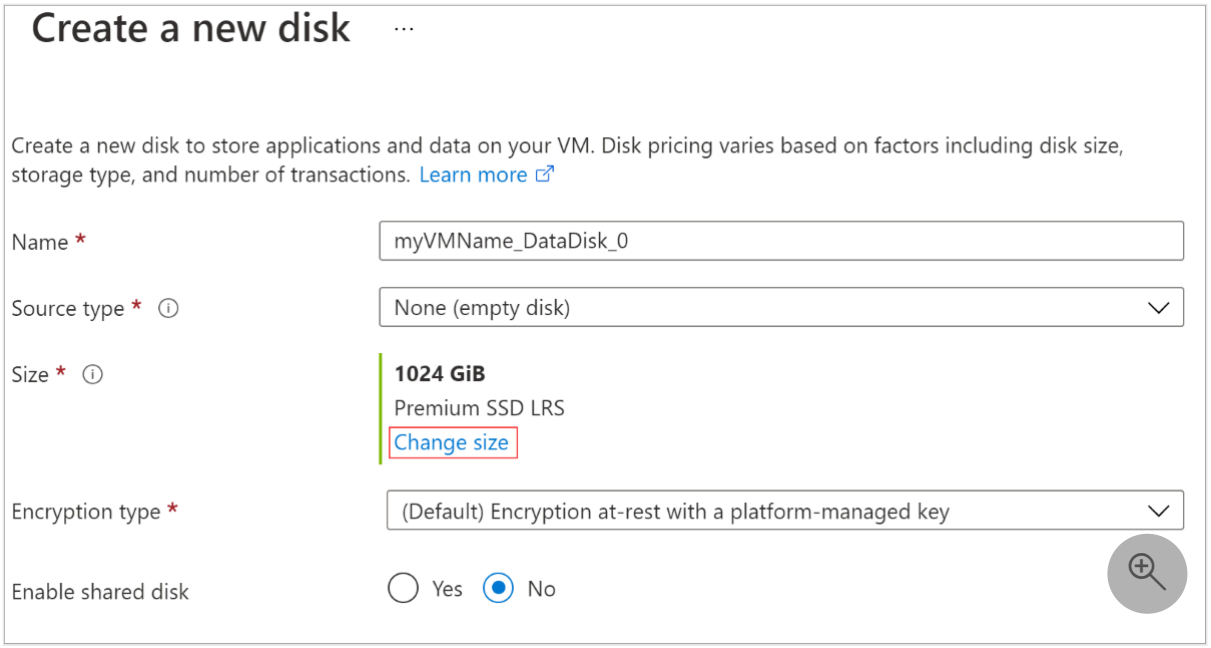


6. On the Disks blade, select **Yes** for **Enable Ultra Disk compatibility**.

7. Select **Create and attach a new disk** to attach an Ultra disk now.



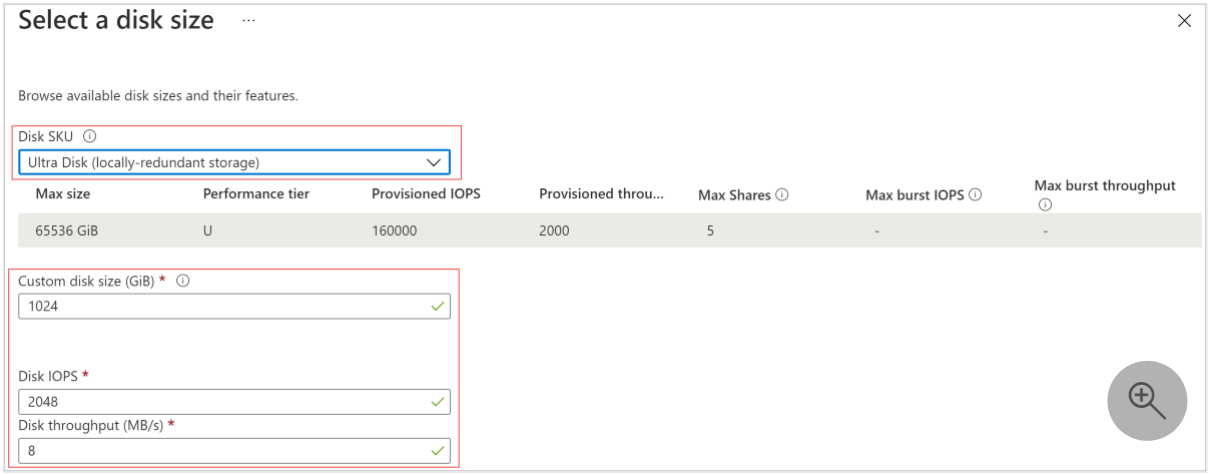
8. On the **Create a new disk** blade, enter a name, then select **Change size**.



9. Change the **Disk SKU** to **Ultra Disk**.

10. Change the values of **Custom disk size (GiB)**, **Disk IOPS**, and **Disk throughput** to ones of your choice.

11. Select **OK** in both blades.



12. Continue with the VM deployment, the same as you would deploy any other VM.

Deploy an Ultra Disk - 512-byte sector size

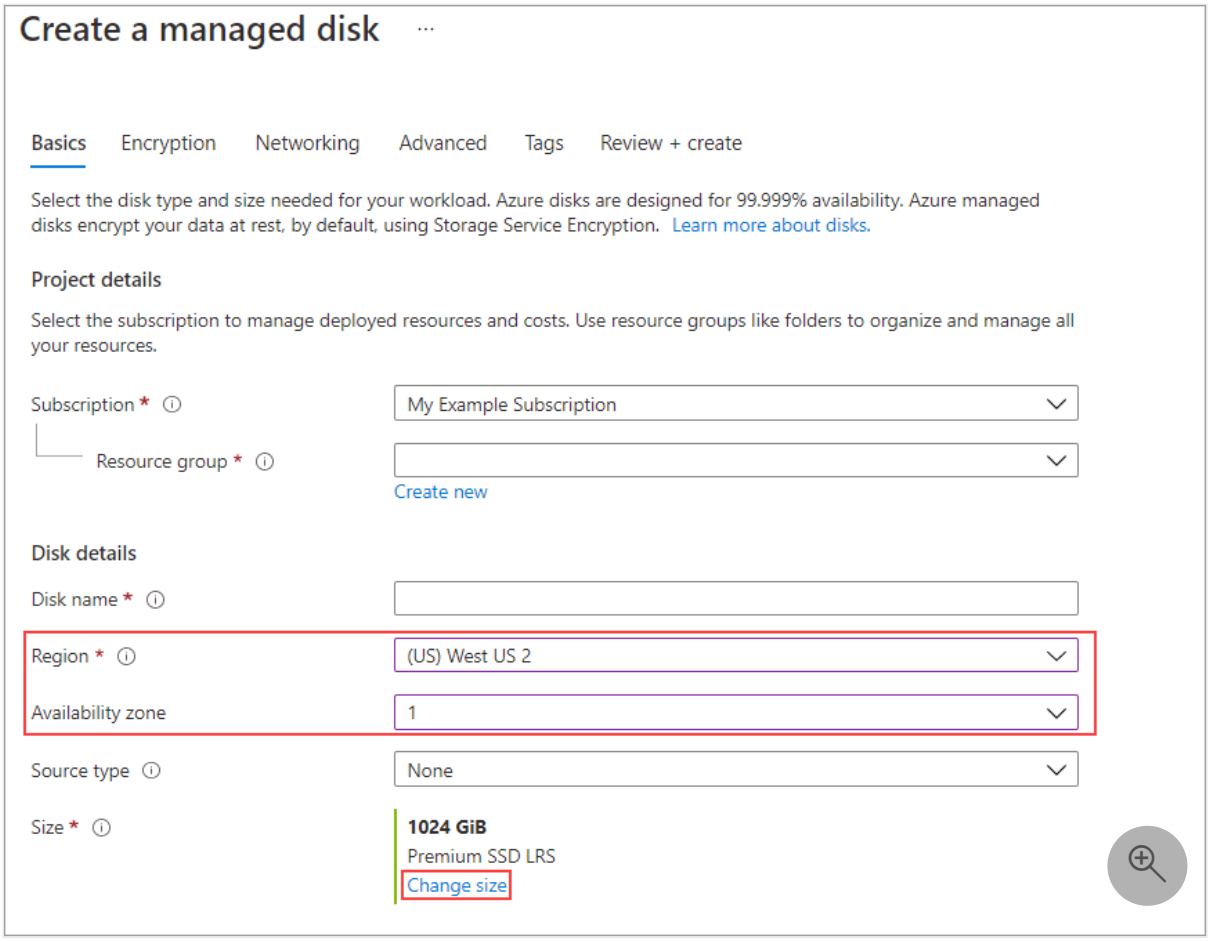
Portal

1. Sign in to the Azure portal, then search for and select **Disks**.

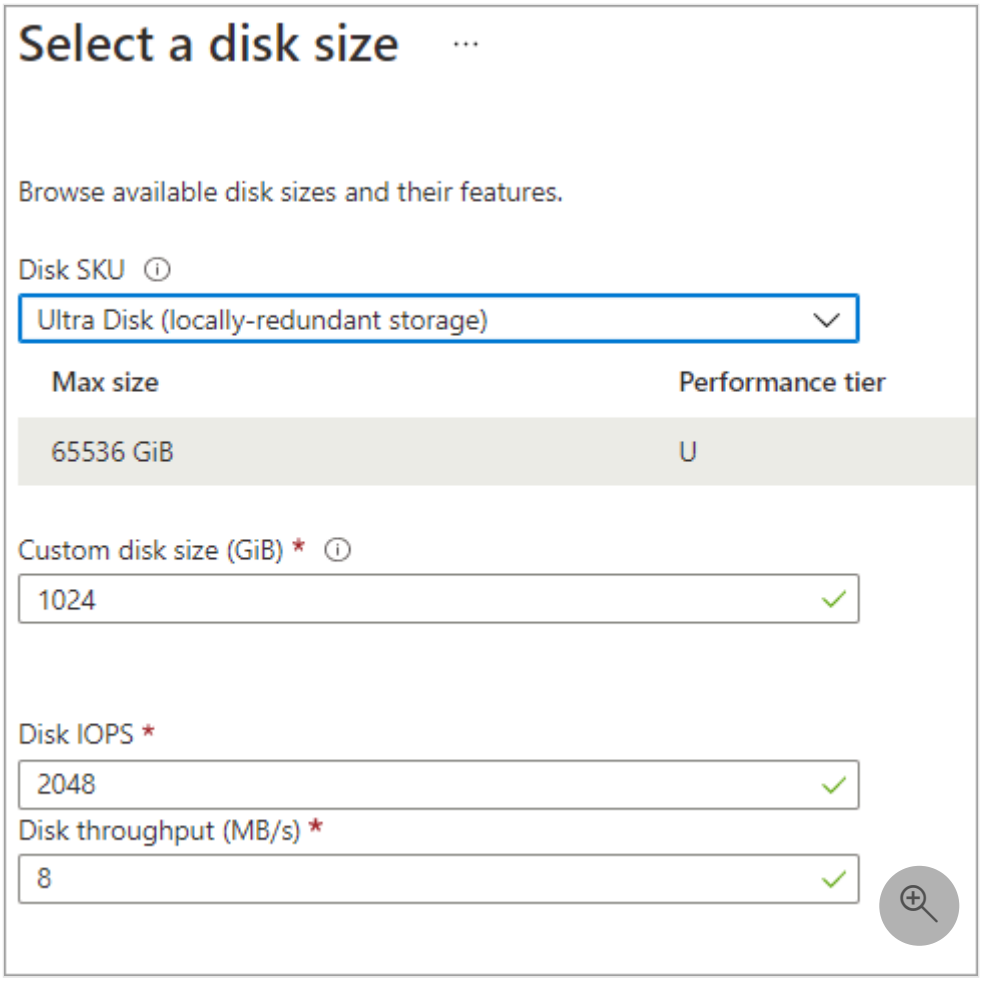
2. Select **+ New** to create a new disk.

3. Select a region that supports Ultra Disks and select an availability zone, fill in the rest of the values as you desire.

4. Select **Change size**.

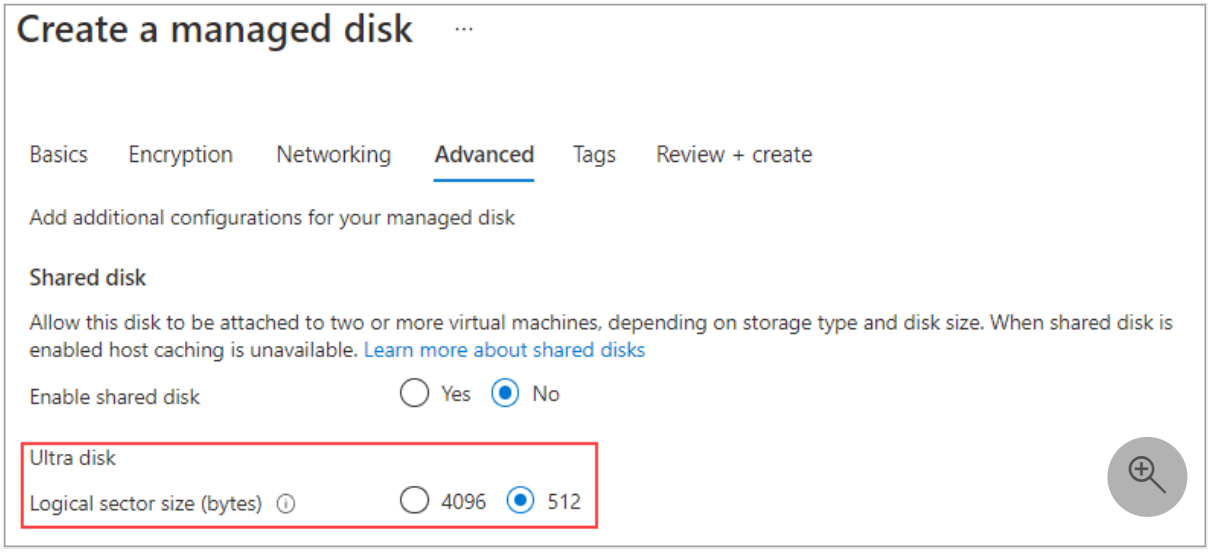


5. For **Disk SKU** select **Ultra Disk**, then fill in the values for the desired performance and select **OK**.



6. On the **Basics** blade, select the **Advanced** tab.

7. Select **512** for **Logical sector size**, then select **Review + Create**.

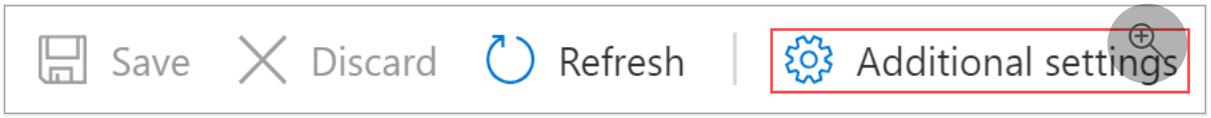


Attach an Ultra Disk

Portal

Alternatively, if your existing VM is in a region/availability zone that is capable of using Ultra Disks, you can make use of Ultra Disks without having to create a new VM. By enabling Ultra Disks on your existing VM, then attaching them as data disks. To enable Ultra Disk compatibility, you must stop the VM. After you stop the VM, you can enable compatibility, then restart the VM. Once compatibility is enabled, you can attach an Ultra Disk:

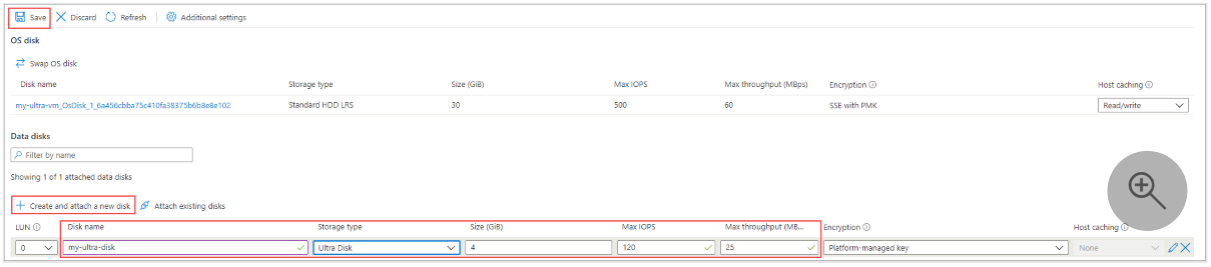
1. Navigate to your VM and stop it, wait for it to deallocate.
2. Once your VM has been deallocated, select **Disks**.
3. Select **Additional settings**.



4. Select **Yes** for **Enable Ultra Disk compatibility**.



5. Select **Save**.
6. Select **Create and attach a new disk** and fill in a name for your new disk.
7. For **Storage type** select **Ultra Disk**.
8. Change the values of **Size (GiB)**, **Max IOPS**, and **Max throughput** to ones of your choice.
9. After you're returned to your disk's blade, select **Save**.



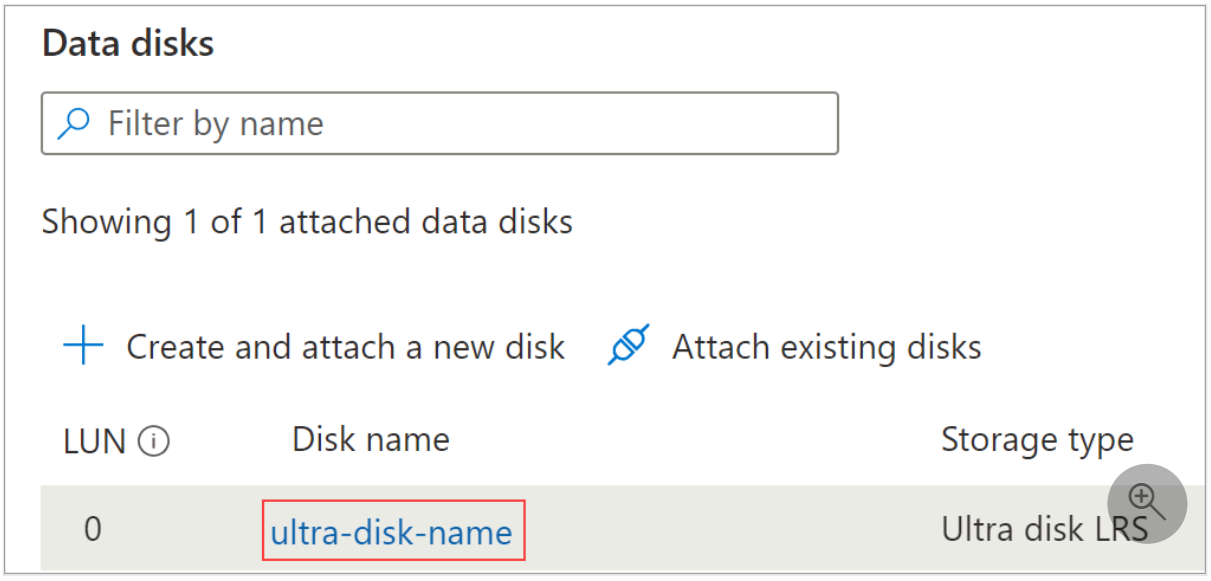
10. Start your VM again.

Adjust the performance of an Ultra Disk

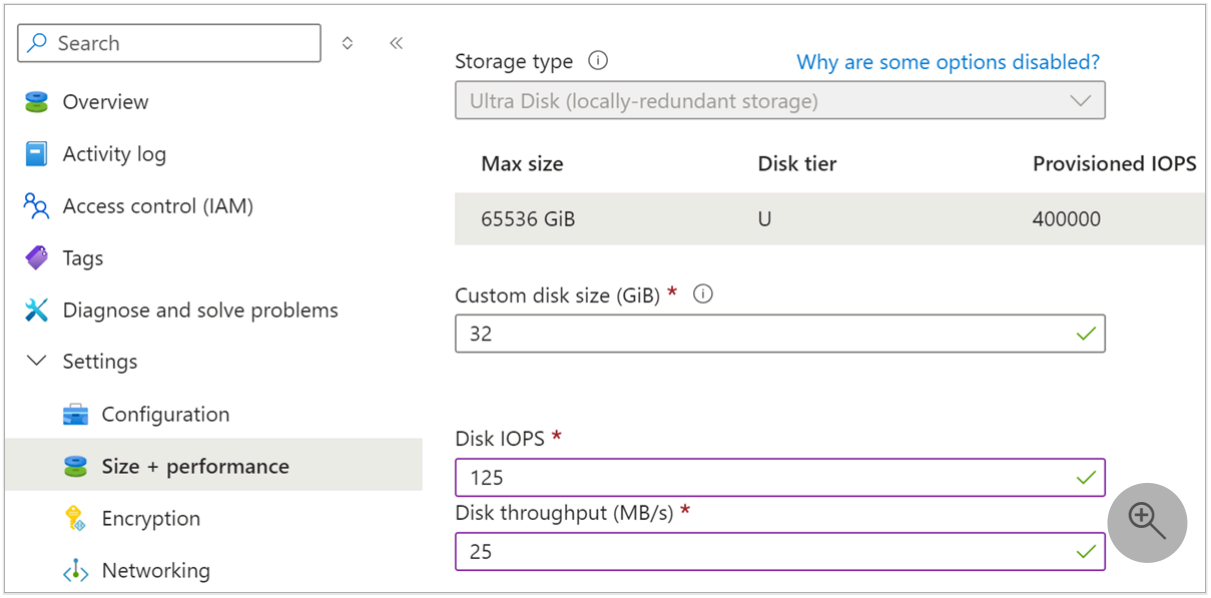
Portal

Ultra Disks offer a unique capability that allows you to adjust their performance. You can adjust the performance of an Ultra Disk four times within a 24 hour period.

1. Navigate to your VM and select **Disks**.
2. Select the Ultra Disk you'd like to modify the performance of.



3. Select **Size + performance** and then make your modifications.
4. Select **Save**.



Next steps

- Use Azure Ultra Disks on Azure Kubernetes Service (preview).

- Migrate log disk to an Ultra Disk.
- For more questions on Ultra Disks, see the Ultra Disks section of the FAQ.

Deploy a Premium SSD v2

Summarize this article for me

Deploy to Azure

Azure Premium SSD v2 is designed for IO-intense enterprise workloads that require sub-millisecond disk latencies and high IOPS and throughput at a low cost. Premium SSD v2 is suited for a broad range of workloads such as SQL server, Oracle, MariaDB, SAP, Cassandra, Mongo DB, big data/analytics, gaming, on virtual machines or stateful containers. For conceptual information on Premium SSD v2, see Premium SSD v2.

Premium SSD v2 disks support a 4k physical sector size by default, but can be configured to use a 512E sector size as well. While most applications are compatible with 4k sector sizes, some require 512 byte sector sizes. Oracle Database, for example, requires release 12.2 or later in order to support 4k native disks.

Limitations

- Premium SSD v2 disks can't be used as an OS disk or with Azure Compute Gallery.
- For regions that support availability zones, Premium SSD v2 disks can only be attached to zonal VMs. When creating a new VM, specify the availability zone you want before adding Premium SSD v2 disks to your configuration.
- Premium SSD v2 doesn't support host caching.

Regional availability

Currently only available in the following regions:

Expand table

Regions without Availability Zone	*One Availability Zone*	*Two Availability Zones*	*Three Availability Zones*
- Australia Central 2 - Australia Southeast - Brazil Southeast - Canada East - Germany North - North Central US - Norway West - Switzerland West - Taiwan North		- Japan West	- Austria East - Australia East - Brazil South - Canada Central - Central India - Central US - China North 3 - East Asia - East US

Regions without Availability Zone	*One Availability Zone*	*Two Availability Zones*	*Three Availability Zones*
- UK West - West Central US - West US			- East US 2 - France Central - Germany West Central - Indonesia Central - Israel Central - Italy North - Japan East - Korea Central - Malaysia West - Mexico Central - New Zealand North - North Europe - Norway East - Poland Central - Spain Central - South African North - South Central US - Southeast Asia - Sweden Central - Switzerland North - UAE North - UK South - US Gov Virginia - West Europe - West US 2 - West US 3

To learn when support for particular regions was added, see either Azure Updates or What's new for Azure Disk Storage.

Prerequisites

- Install either the latest Azure CLI or the latest Azure PowerShell module.

Determine region availability programmatically

Since not every region and zone support Premium SSD v2 disks, you can use the Azure CLI or PowerShell to determine region and zone supportability.

Azure CLI

To determine the regions and zones that support Premium SSD v2 disks, replace `yourSubscriptionId` with your subscription, and then run the az vm list-skus command:

Azure CLI

```
az login

subscriptionId="<yourSubscriptionId>"

az account set --subscription $subscriptionId

az vm list-skus --resource-type disks --query "[?name=='PremiumV2_LRS'].
{Region:locationInfo[0].location, Zones:locationInfo[0].zones}"
```

Now that you know the region and zone to deploy to, follow the deployment steps in this article to create a Premium SSD v2 and attach it to a VM.

Use Premium SSD v2 in Regions with Availability Zones

Currently, Premium SSD v2 disks are only available in select regions with Availability Zones (AZs).

Azure CLI

Create a Premium SSD v2 in an availability zone by using the az disk create command. Then create a VM in the same region and availability zone that supports Premium Storage and attach the disk to it by using the az vm create command.

The following script creates a Premium SSD v2 with a 4k sector size, to deploy one with a 512 sector size, update the `$logicalSectorSize` parameter. Replace the values of all the variables with your own, then run the following script:

Azure CLI

```
## Initialize variables
diskName="yourDiskName"
resourceGroupName="yourResourceGroupName"
region="yourRegionName"
zone="yourZoneNumber"
##Replace 4096 with 512 to deploy a disk with 512 sector size
logicalSectorSize=4096
vmName="yourVMName"
vmImage="Win2016Datacenter"
```

```
adminPassword="yourAdminPassword"
adminUserName="yourAdminUserName"
vmSize="Standard_D4s_v3"

## Create a Premium SSD v2
az disk create -n $diskName -g $resourceGroupName \
--size-gb 100 \
--disk-iops-read-write 5000 \
--disk-mbps-read-write 150 \
--location $region \
--zone $zone \
--sku PremiumV2_LRS \
--logical-sector-size $logicalSectorSize

## Create the VM
az vm create -n $vmName -g $resourceGroupName \
--image $vmImage \
--zone $zone \
--authentication-type password --admin-password $adminPassword --admin-username
$adminUserName \
--size $vmSize \
--location $region \
--attach-data-disks $diskName
```

Use a Premium SSD v2 in non-AZ Regions

Currently, Premium SSD v2 disks are only available in select regions without Availability Zones (AZs). Regions without AZ support may experience slightly higher average latency for Premium SSD v2 disks compared to regions with AZ support.

Azure CLI

Create a Premium SSD v2 in a region without availability zone support by using the az disk create command. Then create a VM in the same region that supports Premium Storage and attach the disk to it by using the az vm create command.

The following script creates a Premium SSD v2 with a 4k sector size. To create a disk with a 512 sector size, update the `$logicalSectorSize` parameter. Replace the values of all the variables with your own, then run the following script:

Azure CLI

```
## Initialize variables
diskName="yourDiskName"
resourceGroupName="yourResourceGroupName"
region="yourRegionName"
##Replace 4096 with 512 to deploy a disk with 512 sector size
```

```
logicalSectorSize=4096
vmName="yourVMName"
vmImage="Win2016Datacenter"
adminPassword="yourAdminPassword"
adminUserName="yourAdminUserName"
vmSize="Standard_D4s_v3"

## Create a Premium SSD v2
az disk create -n $diskName -g $resourceGroupName \
--size-gb 100 \
--disk-iops-read-write 5000 \
--disk-mbps-read-write 150 \
--location $region \
--sku PremiumV2_LRS \
--logical-sector-size $logicalSectorSize

## Create the VM
az vm create -n $vmName -g $resourceGroupName \
--image $vmImage \
--zone $zone \
--authentication-type password --admin-password $adminPassword --admin-username
$adminUserName \
--size $vmSize \
--location $region \
--attach-data-disks $diskName
```

Adjust disk performance

You can adjust the performance of a Premium SSD v2 four times within a 24 hour period. Creating a disk counts as one of these times, so for the first 24 hours after creating a Premium SSD v2 you can only adjust its performance up to three times.

For conceptual information on adjusting disk performance, see Premium SSD v2 performance.

Azure CLI

Use the az disk update command to change the performance configuration of your Premium SSD v2. For example, you can use the `disk-iops-read-write` parameter to adjust the max IOPS limit, and the `disk-mbps-read-write` parameter to adjust the max throughput limit of your Premium SSD v2.

The following command adjusts the performance of your disk. Update the values in the command, and then run it:

Azure CLI

```
az disk update --subscription $subscription --resource-group $rgname --name
$diskName --disk-iops-read-write=5000 --disk-mbps-read-write=200
```

Next steps

Add a data disk by using either the Azure portal, Azure CLI, or PowerShell.

Provide feedback on Premium SSD v2.

Last updated on 03/02/2026

Deploy a managed disk that uses zone-redundant storage

07/02/2025

This article covers how to deploy a disk that uses zone-redundant storage (ZRS) as a redundancy option. ZRS replicates your Azure managed disk synchronously across three Azure availability zones in the selected region. Each availability zone is a separate physical location with independent power, cooling, and networking.

For conceptual information on ZRS, see Zone-redundant storage for managed disks

Limitations

ZRS for managed disks is only supported with Premium SSD and Standard SSD managed disks. ZRS for managed disks isn't supported with Premium SSD v2 managed disks or Ultra Disks. Force detach only supports data disks.

Regional availability

ZRS disks are currently available in the following regions:

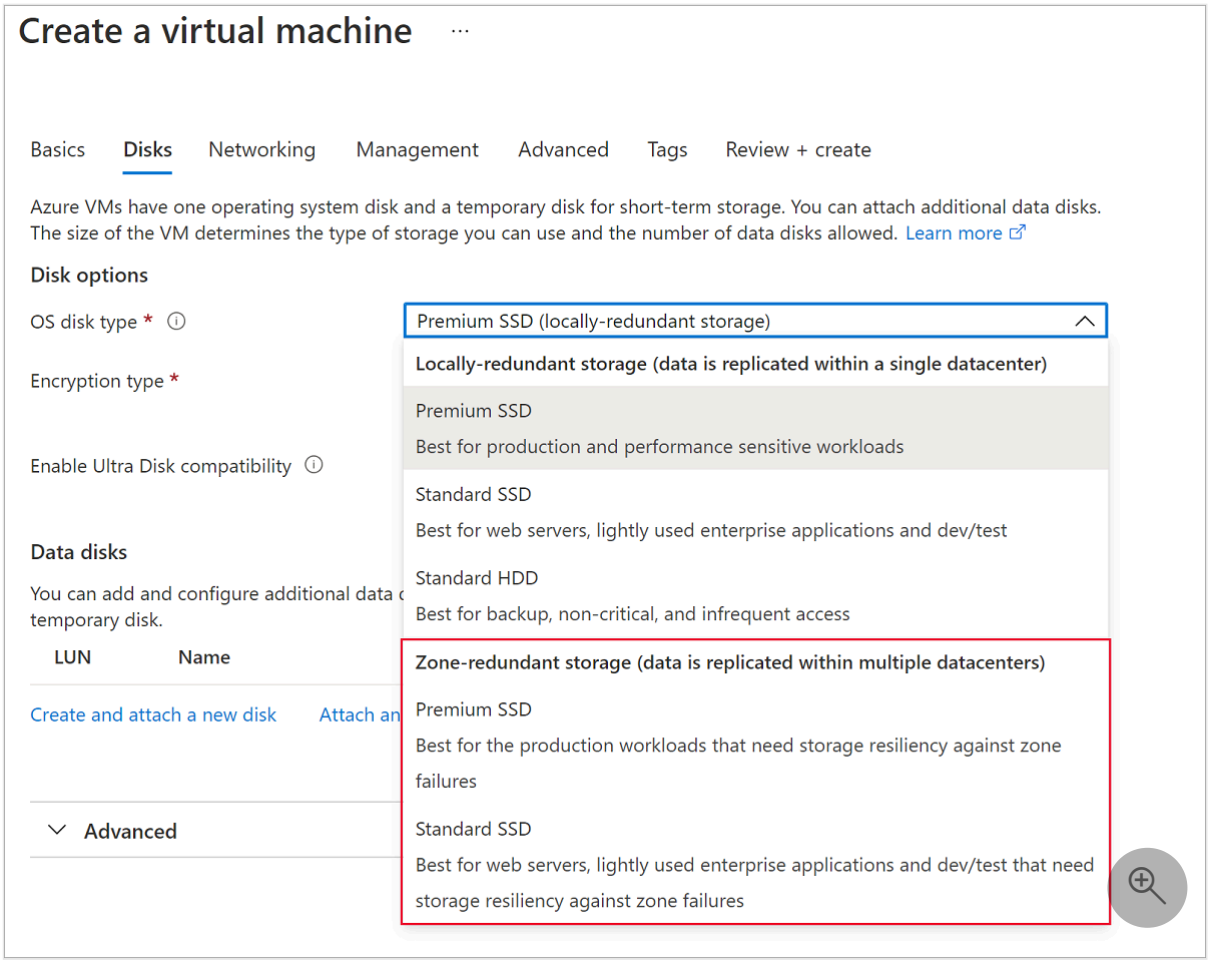
- South Africa North
- East Asia
- Southeast Asia
- Australia East
- US Gov Virginia
- Brazil South
- Canada Central
- China North 3
- North Europe
- West Europe
- France Central
- Germany West Central
- Central India
- Israel Central
- Italy North
- Japan East
- Korea Central
- Mexico Central

- New Zealand North
- Norway East
- Poland Central
- Qatar Central
- Spain Central
- Sweden Central
- Switzerland North
- UAE North
- UK South
- Central US
- East US
- East US 2
- South Central US
- West US 2
- West US 3

Azure portal

Create a VM with a ZRS OS disk

1. Sign in to the Azure portal.
2. Navigate to **Virtual machines** and follow the normal VM creation process.
3. Proceed to the **Disks** pane.
4. Select your disk and select one of the ZRS disks in the drop-down.

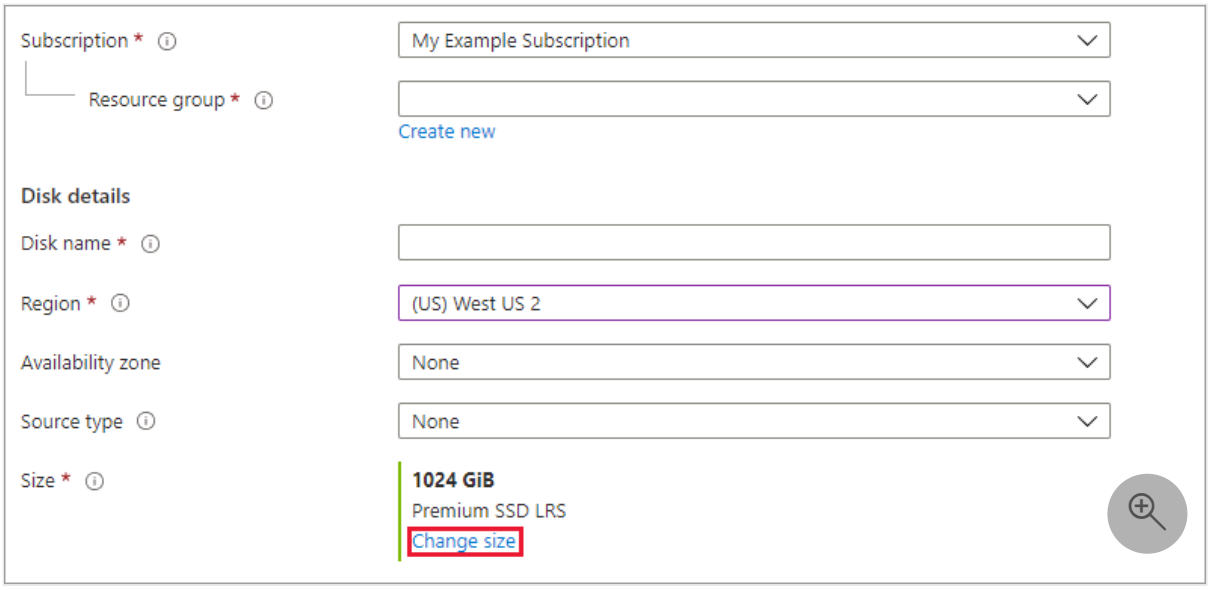


5. Proceed through the rest of the VM deployment, making any choices that you desire.

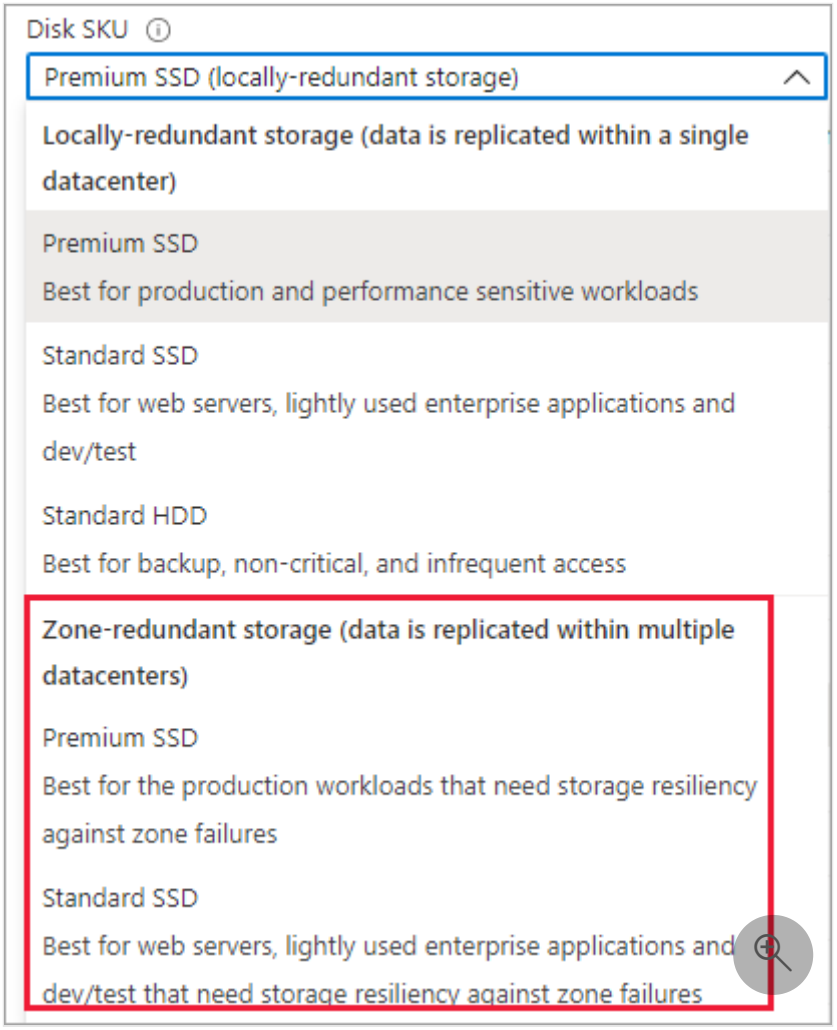
You've now deployed a VM with a ZRS OS disk.

Create a ZRS disk

1. In the Azure portal, search for and select **Disks**.
2. Select **+ Add** to create a new disk.
3. Select a supported region and **Availability zone** to **None**.
4. Select **Change size**.



5. Select one of the available ZRS disks and select **OK**.



6. Continue through the deployment process.

You have now created a managed disk that uses ZRS.

Next steps

- Check out more Azure Resource Manager templates to create a VM with ZRS disks .

Best practices for achieving high availability with Azure virtual machines and managed disks

Azure offers several configuration options for ensuring high availability of Azure virtual machines (VMs) and Azure managed disks. This article covers the default availability and durability of managed disks and provides recommendations to further increase your application's availability and resiliency.

At a glance

Expand table

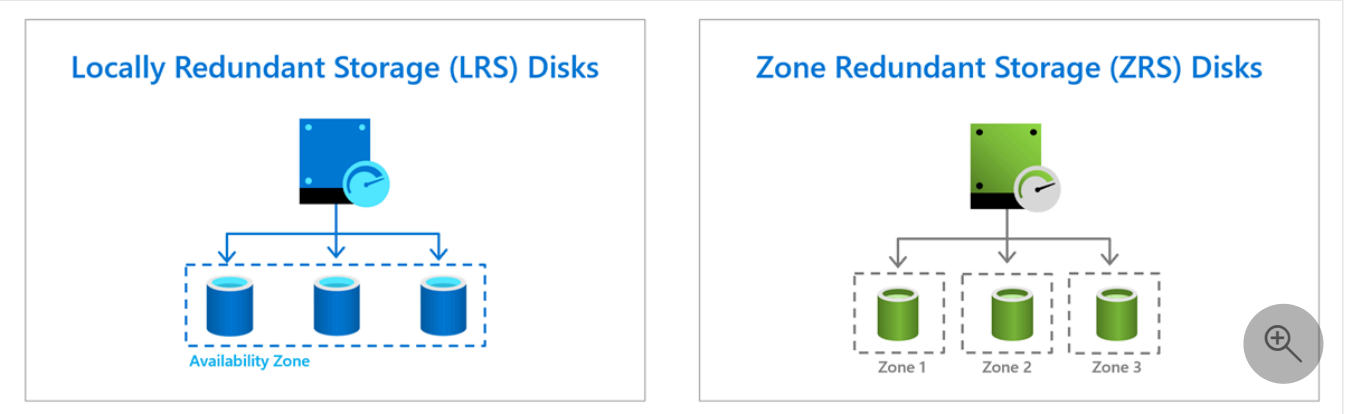
Configuration	Recommendation	Benefits
Applications running on a single VM	Use Ultra Disks, Premium SSD v2, and Premium SSD disks.	Single VMs using only Premium SSD disks as the OS disks, and either Ultra Disks, Premium SSD v2, or Premium SSD disks as data disks have the highest uptime service level agreement (SLA), and these disk types offer the best performance.
	Use zone-redundant storage (ZRS) disks.	Access to your data even if an entire zone experiences an outage.
Applications running on multiple VMs	Distribute VMs and disks across multiple availability zones using a zone-spanning Virtual Machine Scale Set with flexible orchestration mode or by deploying VMs and disks across three availability zones.	Multiple VMs have the highest uptime SLA when deployed across multiple zones.
	Deploy VMs and disks across multiple fault domains with either regional Virtual Machine Scale Sets with flexible orchestration mode or availability sets.	Multiple VMs have the second highest uptime SLA when deployed across fault domains.
	Use ZRS disks when sharing disks between VMs.	Prevents a shared disk from becoming a single point of failure.

Availability and durability of managed disks

Before going over recommendations for achieving higher availability, you should understand the default availability and durability of managed disks.

Managed disks are designed for 99.999% availability and provide at least 99.999999999% (11 9's) of durability. With managed disks, your data is replicated three times. If one of the three copies becomes unavailable, Azure automatically spawns a new copy of the data in the background. This ensures the persistence of your data and high fault tolerance.

Managed disks have two redundancy models, locally redundant storage (LRS) disks, and zone-redundant storage (ZRS) disks. The following diagram depicts how data is replicated with either model.



LRS disks provide at least 99.999999999% (11 9's) of durability over a given year and ZRS disks provide at least 99.9999999999% (12 9's) of durability over a given year. This architecture helps Azure consistently deliver enterprise-grade durability for infrastructure as a service (IaaS) disks, with an industry-leading zero percent annualized failure rate.

Recommendations for applications running on a single VM

Legacy applications, traditional web servers, line-of-business applications, development and testing environments, and small workloads are all examples of applications that may run on a single VM. These applications can't benefit from replication across multiple VMs, but the data on the disks is still replicated three times, and you can take the following steps to further increase availability.

Use Ultra Disks, Premium SSD v2, or Premium SSD

Single VMs using only Premium SSD disks as the OS disk, and either Ultra Disks, Premium SSD v2, or Premium SSD disks as data disks have the highest single VM uptime SLA, and these disk types offer the best performance.

Use zone-redundant storage disks

Zone-redundant storage (ZRS) disks synchronously replicate data across three availability zones, which are separated groups of data centers in a region that have independent power, cooling, and networking infrastructure. With ZRS disks, your data is accessible even in the event of a zonal outage. Also, ZRS data disks allow you to forcibly detach them from VMs experiencing issues. ZRS disks have limitations, see the limitations section of the redundancy options article for details.

Recommendations for applications running on multiple VMs

Quorum-based applications, clustered databases (SQL, MongoDB), enterprise-grade web applications, and gaming applications are all examples of applications running on multiple VMs. Applications running on multiple VMs can designate a primary VM and multiple secondary VMs and replicate data across these VMs. This setup enables failover to a secondary VM if the primary VM goes down.

Multiple VMs have the highest uptime service level agreement (SLA) when deployed across multiple availability zones, and they have the second highest uptime SLA when deployed across multiple storage and compute fault domains.

Distribute VMs and disks across availability zones

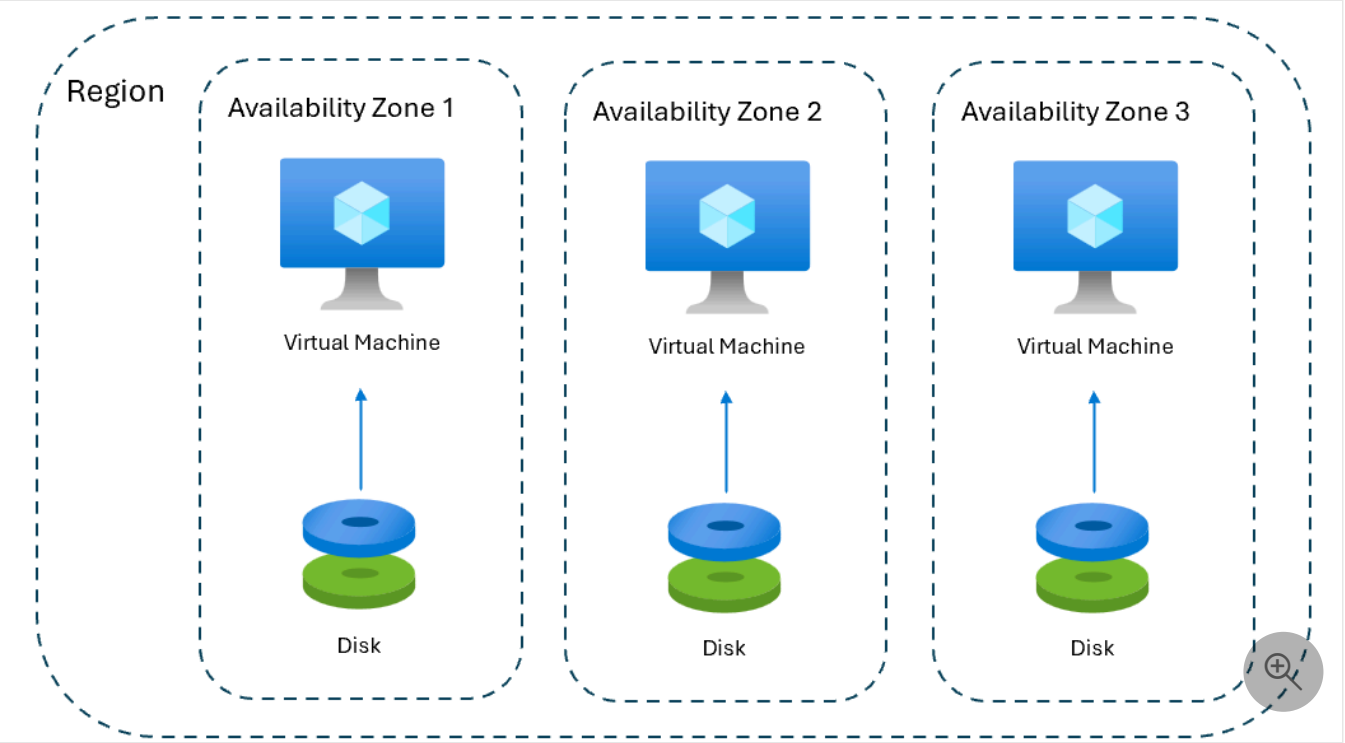
Availability zones are separated groups of data centers within a region that have independent power, cooling, and networking infrastructure. They're close enough to have low-latency connections to other availability zones but far enough to reduce the possibility that more than one is affected by local outages or weather. See What are availability zones? for details.

Multiple VMs have the highest SLA when distributed across three availability zones. For VMs and disks distributed across multiple availability zones, the disks and their parent VMs are respectively collocated in the same zone, which prevents multiple VMs from going down even if an entire zone experiences an outage. Availability zones aren't currently available in every region, see Azure regions with availability zone support.

VMs distributed across multiple availability zones may have higher network latency than VMs distributed in a single availability zone, which could be a concern for workloads that require ultra-low latency. If low latency is your top priority, consider the methods described in Deploy VMs and disks across multiple fault domains.

To deploy resources across availability zones, you can either use zone-spanning Virtual Machine Scale Sets or deploy resources across availability zones.

The following diagram depicts how VMs and disks are collocated in the same zones when deployed across availability zones directly or using zone-spanning Virtual Machine Scale Sets.



Use zone-spanning Virtual Machine Scale Sets with flexible orchestration

Virtual Machine Scale Sets let you create and manage a group of load balanced VMs. The number of VM instances can automatically adjust in response to demand or follow a schedule you define. A zone-spanning Virtual Machine Scale Set is a Virtual Machine Scale Set that has been deployed across multiple availability zones. See Zone spanning.

With zone-spanning Virtual Machine Scale Sets using the flexible orchestration mode, VMs, and their disks are replicated to one or more zones within the region they're deployed in to improve the resiliency and availability of your applications and data. This configuration spreads VMs across selected zones in a best effort approach by default but also provides the ability to specify strict zone balance in the deployment.

Deploy VMs and disks across three availability zones

Another method to distribute VMs and disks across availability zones is to deploy the VMs and disks across three availability zones. This deployment provides redundancy in VMs and disks across multiple data centers in a region, allowing you to fail over to another zone if there's a data center or zonal outage.

Deploy VMs and disks across multiple fault domains

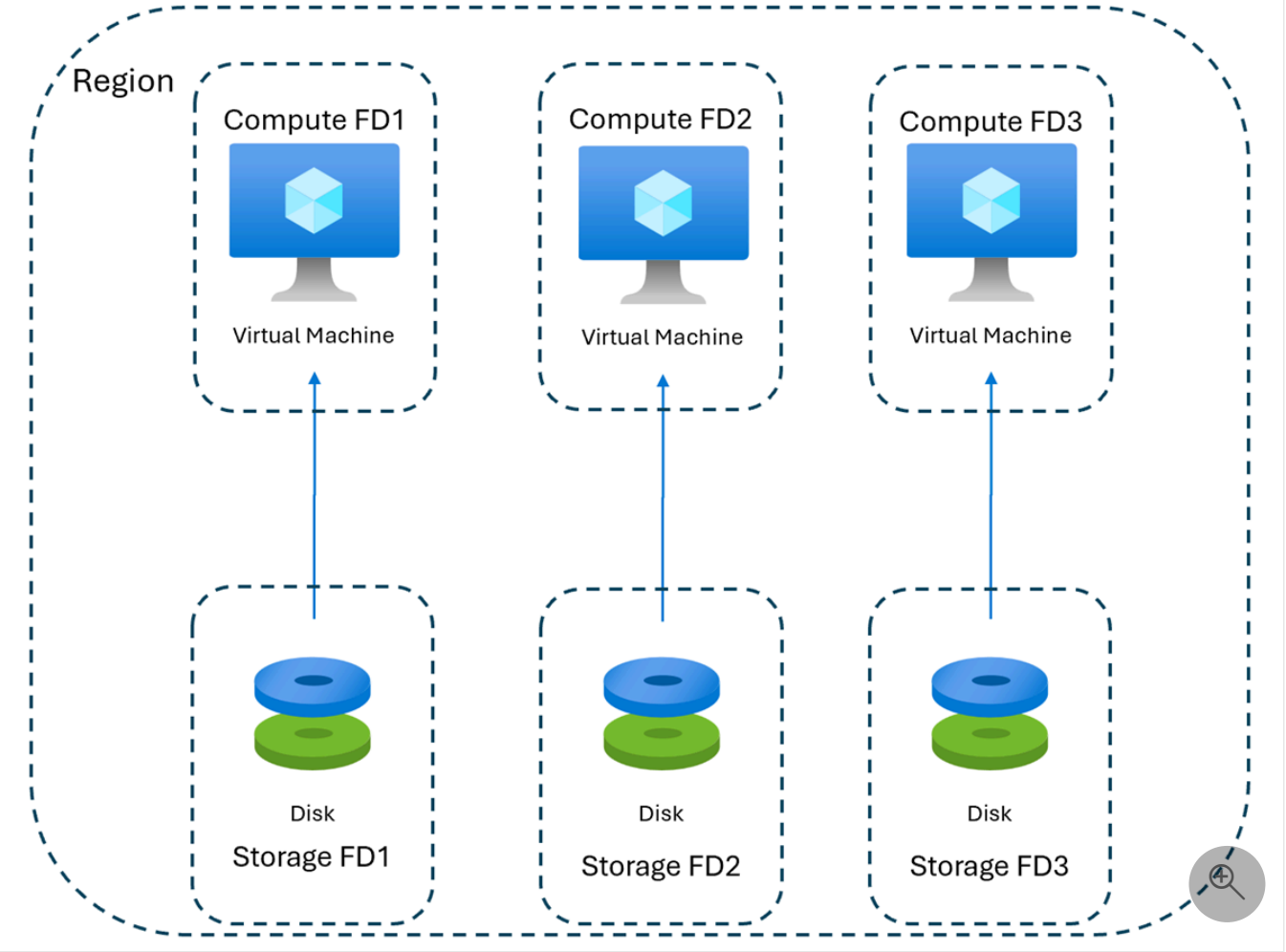
If you can't deploy your VMs and disks across availability zones or have ultra-low latency requirements, you can deploy them across fault domains instead. Fault domains define groups of VMs that share a common power source and a network switch. For details, see How do availability sets work?.

For VMs and disks deployed across fault domains via the following methods, the storage fault domains of the disks are aligned with the compute fault domains of their respective parent VMs, which prevents multiple VMs from going down if a single storage fault domain experiences an outage.

Multiple VMs have the second highest uptime SLA when deployed across fault domains. To learn more, see the Virtual Machines section of the SLA.

To deploy resources across multiple fault domains, you can either use regional Virtual Machine Scale Sets or availability sets.

The following diagram depicts the alignment of compute and storage fault domains when using either regional Virtual Machine Scale Sets or availability sets.



Use regional Virtual Machine Scale Sets with flexible orchestration

A regional Virtual Machine Scale Set is a Virtual Machine Scale Set that has no explicitly defined availability zones. With regional virtual machine scale sets, VM resources are replicated across fault domains within the region they're deployed in to improve the resiliency and availability of applications and data. This configuration spreads VMs across fault domains by default but also provides the ability to assign fault domains on VM creation. See this section for details.

Regional Virtual Machine Scale Sets don't protect against large-scale outages like a data center or region outage, and don't currently support Ultra Disks or Premium SSD v2 disks.

Use availability sets

Availability sets are logical groupings of VMs that place VMs in different fault domains to limit the chance of correlated failures bringing related VMs down at the same time. Availability sets also have better VM to VM latencies compared to availability zones.

Availability sets don't let you select the fault domains for your VMs, can't be used with availability zones, don't protect against data center or region-wide outages, and don't currently support Ultra Disks. Premium SSD v2 disks are supported with some limitations.

Use ZRS disks when sharing disks between VMs

You should use ZRS when sharing a disk between multiple VMs. If you use LRS, the shared disk becomes a single point of failure for your clustered application. This means that if your shared LRS disk experiences an outage, all the VMs to which this disk is attached will experience downtime. Using a ZRS disk mitigates this, since the disk's data is in three different availability zones. To learn more about shared disks, see Share an Azure managed disk.

Next steps

- Zone-redundant storage for managed disks
- What are availability zones?
- Create a Virtual Machine Scale Set that uses Availability Zones

---

Last updated on 01/26/2026

Share an Azure managed disk across VMs

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure shared disks is a feature for Azure managed disks that allows you to attach a managed disk to multiple virtual machines (VMs) simultaneously. This capability enables you to deploy new or migrate existing clustered applications to Azure without modifying your architecture. Shared disks ensure high availability for clustered applications, as other VMs in the cluster retain full access to the disk if one VM fails.

Shared disks require a cluster manager, like Windows Server Failover Cluster (WSFC), or Pacemaker, that handles cluster node communication and write locking. Shared managed disks don't natively offer a fully managed file system that can be accessed using SMB/NFS.

How it works

VMs in the cluster can read or write to their attached disk based on the reservation chosen by the clustered application using SCSI Persistent Reservations (SCSI PR). SCSI PR is an industry standard used by applications running on Storage Area Network (SAN) on-premises. Enabling SCSI PR on a managed disk allows you to migrate these applications to Azure as-is.

Shared managed disks offer shared block storage that can be accessed from multiple VMs, these are exposed as logical unit numbers (LUNs). LUNs are then presented to an initiator (VM) from a target (disk). These LUNs look like direct-attached-storage (DAS) or a local drive to the VM.

Limitations

General limitations

Shared disks have general limitations that apply to all shared disks, regardless of disk type. They also have more limitations that only apply to specific types of shared disks. The following list is the list of general limitations:

- Currently, only Ultra Disks, Premium SSD v2, Premium SSD, and Standard SSDs can be used as a shared disk
- Shared disks can be attached to individual Virtual Machine Scale Sets but can't be defined in the Virtual Machine Scale Set models or automatically deployed
- A shared disk can't be expanded without either deallocating all VMs the disk is attached to, or detaching the disk from all of these VMs
- Write accelerator isn't supported for shared disks

- Host caching isn't supported for shared disks

Each managed disk that has shared disks enabled are also subject to the following limitations, organized by disk type:

Ultra disks

Ultra disks have their own separate list of limitations, unrelated to shared disks. For ultra disk limitations, refer to Using Azure ultra disks.

When sharing ultra disks, they have the following additional limitations:

- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Can't be shared across availability zones.

Premium SSD v2

Premium SSD v2 managed disks have their own separate list of limitations, unrelated to shared disks. For these limitations, see Premium SSD v2 limitations.

When sharing Premium SSD v2 disks, they have the following additional limitation:

- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Can't be shared across availability zones.

Premium SSD

- Can only be enabled on data disks, not OS disks.
- Host caching isn't available for premium SSD disks with `maxShares>1`.
- Disk bursting isn't available for premium SSD disks with `maxShares>1`.
- When using Availability sets or Virtual Machine Scale Sets with Azure shared disks, storage fault domain alignment with virtual machine fault domain isn't enforced for the shared data disk.
- When using proximity placement groups (PPG), all virtual machines sharing a disk must be part of the same PPG.
- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Azure Site Recovery support isn't yet available.
- Azure Backup is available through Azure Disk Backup.
- Only server-side encryption is supported, Azure Disk Encryption isn't currently supported.

- Can only be shared across availability zones if using Zone-redundant storage for managed disks.

Standard SSDs

- Can only be enabled on data disks, not OS disks.
- Host caching isn't available for standard SSDs with `maxShares>1`.
- When using Availability sets and Virtual Machine Scale Sets with Azure shared disks, storage fault domain alignment with virtual machine fault domain isn't enforced for the shared data disk.
- When using proximity placement groups (PPG), all virtual machines sharing a disk must be part of the same PPG.
- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Azure Site Recovery support isn't yet available.
- Azure Backup is available through Azure Disk Backup.
- Only server-side encryption is supported, Azure Disk Encryption isn't currently supported.
- Can only be shared across availability zones if using Zone-redundant storage for managed disks.

Operating system requirements

Shared disks support several operating systems. See the Windows or Linux sections for the supported operating systems.

Billing implications

When you share a disk, your billing could be impacted in two different ways, depending on the type of disk.

For shared Premium SSD disks, in addition to cost of the disk's tier, there's an extra charge that increases with each VM the SSD is mounted to. See managed disks pricing for details.

Both shared Ultra Disks and shared Premium SSD v2 disks don't have an extra charge for each VM that they're mounted to. They're billed on the total IOPS and MB/s that the disk is configured for. Normally, Ultra Disks and Premium SSD v2 has two performance throttles that determine its total IOPS/MB/s. However, when configured as a shared disk, two more performance throttles are exposed, for a total of four. These two extra throttles allow for increased performance at an extra expense and each meter has a default value, which raises the performance and cost of the disk.

The four performance throttles a shared Ultra Disk and shared Premium SSD v2 disk have are diskIOPSReadWrite, diskMB/sReadWrite, diskIOPSReadOnly, and diskMB/sReadOnly. Each performance throttle can be configured to change the performance of your disk. The performance of shared Ultra Disks and Premium SSD v2 disks are calculated in the following ways: total provisioned IOPS (diskIOPSReadWrite + diskIOPSReadOnly) and for total provisioned throughput MB/s (diskMB/sReadWrite + diskMB/sReadOnly).

Once you've determined your total provisioned IOPS and total provisioned throughput, you can use them in the pricing calculator to determine the cost of an Ultra shared disk or a Premium SSD v2 shared disk.

Disk sizes

For now, only ultra disks, premium SSD v2, premium SSD, and standard SSDs can enable shared disks. Different disk sizes may have a different `maxShares` limit, which you can't exceed when setting the `maxShares` value.

For each disk, you can define a `maxShares` value that represents the maximum number of nodes that can simultaneously share the disk. For example, if you plan to set up a 2-node failover cluster, you would set `maxShares=2`. The maximum value is an upper bound. Nodes can join or leave the cluster (mount or unmount the disk) as long as the number of nodes is lower than the specified `maxShares` value.

> **Note**
>
> The `maxShares` value can only be set or edited when the disk is detached from all nodes.

Premium SSD ranges

The following table illustrates the allowed maximum values for `maxShares` by premium SSD sizes:

Expand table

Disk sizes	maxShares limit
P1,P2,P3,P4,P6,P10,P15,P20	3
P30, P40, P50	5
P60, P70, P80	10

The IOPS and bandwidth limits for a disk aren't affected by the `maxShares` value. For example, the max IOPS of a P15 disk is 1100 whether maxShares = 1 or maxShares > 1.

Standard SSD ranges

The following table illustrates the allowed maximum values for `maxShares` by standard SSD sizes:

Expand table

Disk sizes	maxShares limit
E1,E2,E3,E4,E6,E10,E15,E20	3
E30, E40, E50	5
E60, E70, E80	10

The IOPS and bandwidth limits for a disk aren't affected by the `maxShares` value. For example, the max IOPS of a E15 disk is 500 whether maxShares = 1 or maxShares > 1.

Ultra disk ranges

The minimum `maxShares` value is 1, while the maximum `maxShares` value is 15. There are no size restrictions on ultra disks, any size ultra disk can use any value for `maxShares`, up to and including the maximum value.

Premium SSD v2 ranges

The minimum `maxShares` value is 1, while the maximum `maxShares` value is 15. There are no size restrictions on Premium SSD v2, any size Premium SSD v2 disk can use any value for `maxShares`, up to and including the maximum value.

Sample workloads

Windows

Azure shared disks are supported on Windows Server 2008 and newer. Most Windows-based clustering builds on WSFC, which handles all core infrastructure for cluster node communication, allowing your applications to take advantage of parallel access patterns. WSFC

enables both CSV and non-CSV-based options depending on your version of Windows Server. For details, refer to Create a failover cluster.

Some popular applications running on WSFC include:

- Create an FCI with Azure shared disks (SQL Server on Azure VMs)
  - Migrate your failover cluster instance to SQL Server on Azure VMs with shared disks
- File servers. You can use the Scale-Out File Server features deployed on a Windows Server failover cluster, which uses shared disk in active-active mode. Cluster witness resources are stored on Azure shared disks and all file shares are simultaneously online on all nodes. For an example, see the Scale-out File Server (SoFS) template
- SAP application servers use clustered shared disks to store SAP Central Services (ASCS for ABAP and SCS for Java) and SAP global host files. SAP ASCS/SCS template
- File Server for General Use (IW workload) - shared disks enable high availability for file servers in general
- Remote Desktop Server User Profile Disk (RDS UPD)

Linux

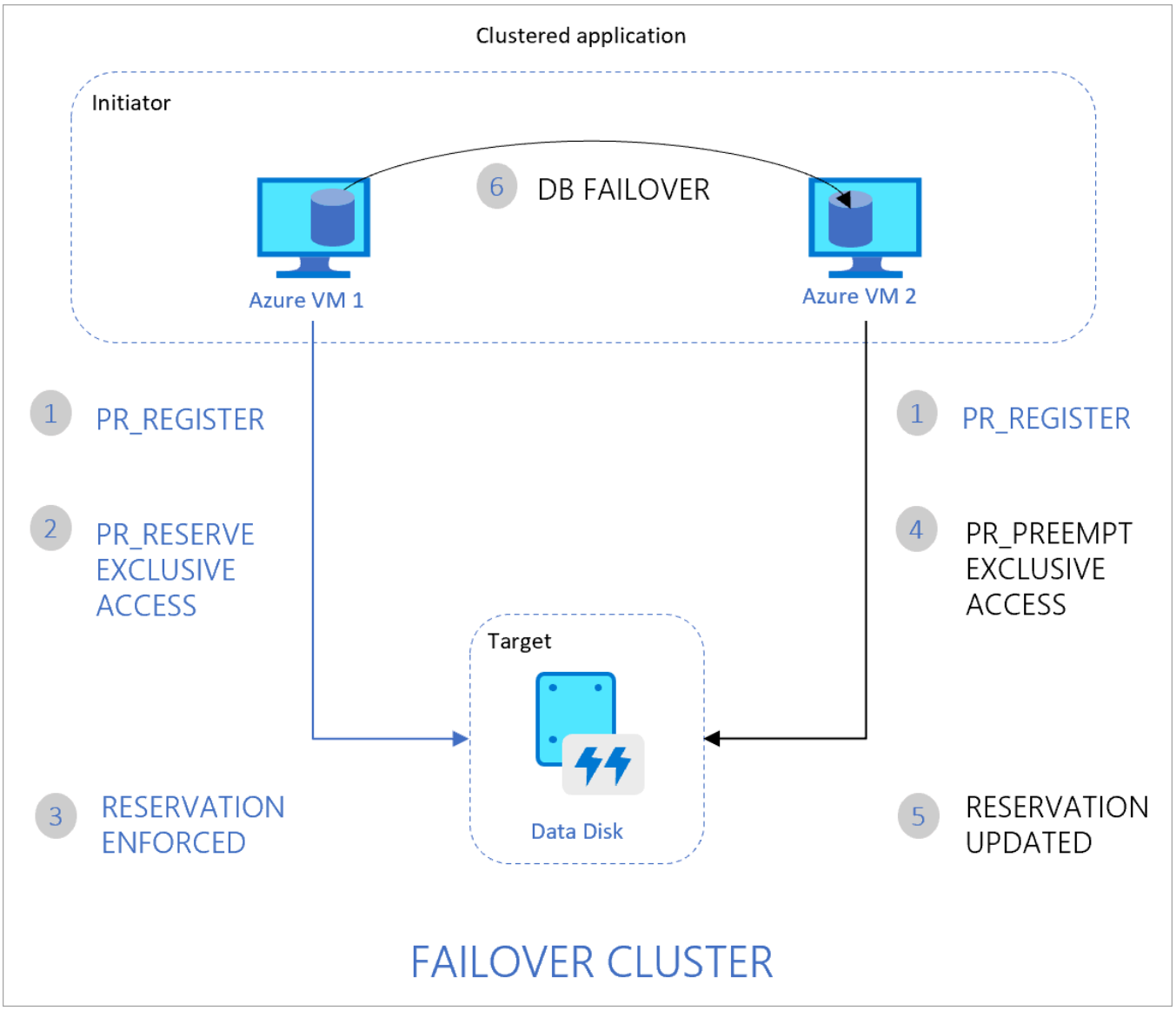
Azure shared disks are supported on:

- SUSE SLE HA 15 SP1 and above
- Ubuntu 18.04 and above
- Red Hat Enterprise Linux (RHEL) (support policy)
  - RHEL 7.9
  - RHEL 8.3 and above
- Oracle Enterprise Linux

Linux clusters can use cluster managers such as Pacemaker. Pacemaker builds on Corosync, enabling cluster communications for applications deployed in highly available environments. Some common clustered filesystems include ocfs2 and gfs2. You can use SCSI Persistent Reservation (SCSI PR) and/or STONITH Block Device (SBD) based clustering models for arbitrating access to the disk. When using SCSI PR, you can manipulate reservations and registrations using utilities such as fence_scsi and sg_persist.

Persistent reservation flow

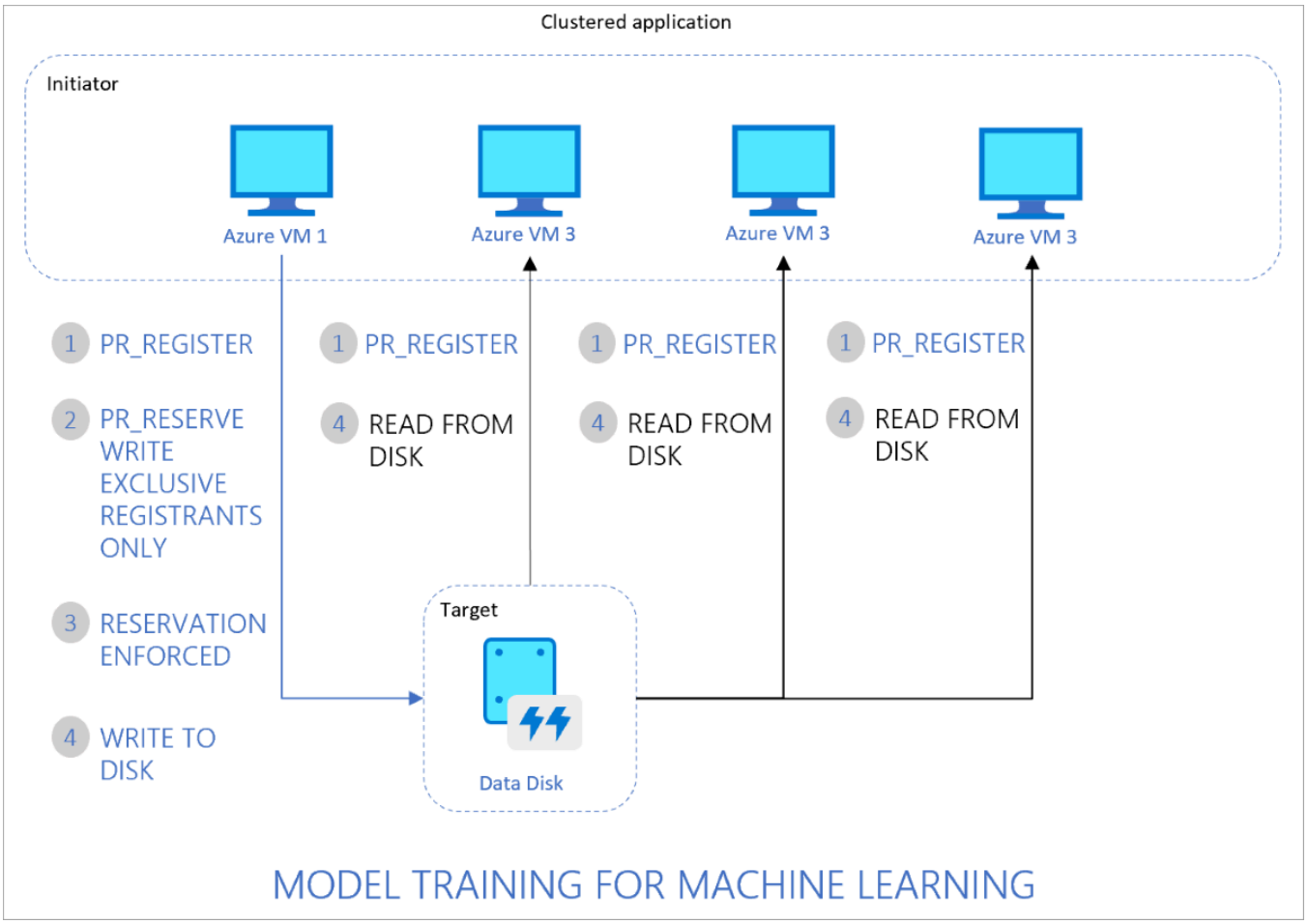
The following diagram illustrates a sample 2-node clustered database application that uses SCSI PR to enable failover from one node to the other.



The flow is as follows:

1. The clustered application running on both Azure VM1 and VM2 registers its intent to read or write to the disk.
2. The application instance on VM1 then takes exclusive reservation to write to the disk.
3. This reservation is enforced on your Azure disk and the database can now exclusively write to the disk. Any writes from the application instance on VM2 won't succeed.
4. If the application instance on VM1 goes down, the instance on VM2 can now initiate a database failover and take-over of the disk.
5. This reservation is now enforced on the Azure disk and the disk will no longer accept writes from VM1. It will only accept writes from VM2.
6. The clustered application can complete the database failover and serve requests from VM2.

The following diagram illustrates another common clustered workload consisting of multiple nodes reading data from the disk for running parallel processes, such as training of machine learning models.



The flow is as follows:

1. The clustered application running on all VMs registers the intent to read or write to the disk.
2. The application instance on VM1 takes an exclusive reservation to write to the disk while opening up reads to the disk from other VMs.
3. This reservation is enforced on your Azure disk.
4. All nodes in the cluster can now read from the disk. Only one node writes back results to the disk, on behalf of all nodes in the cluster.

Ultra Disk and Premium SSD v2 reservation flow

Both Ultra disks and Premium SSD v2 managed disks offer two extra throttles, giving each of them a total of four throttles. Due to this, the reservation flow can work as described in the earlier section, or it can throttle and distribute performance more granularly.

	Reservation Holder		Registered		Others	
Reservation Type	ReadOnly	ReadWrite	ReadOnly	ReadWrite	ReadOnly	ReadWrite
No Reservation	N/A	N/A	N/A	Yes	N/A	Yes
Write Exclusive	N/A	Yes	Yes	N/A	Yes	N/A
Exclusive Access	N/A	Yes	N/A	N/A	N/A	N/A
Write Exclusive – Registrants Only	N/A	Yes	N/A	Yes	Yes	N/A
Exclusive Access – Registrants Only	N/A	Yes	N/A	Yes	N/A	N/A
Write Exclusive – All Registrants	N/A	Yes	N/A	Yes	Yes	N/A
Exclusive Access – All Registrants	N/A	Yes	N/A	Yes	N/A	N/A

Performance throttles

Premium SSD performance throttles

With Premium SSD, the disk IOPS and throughput is fixed, for example, IOPS of a P30 is 5000. This value remains whether the disk is shared across 2 VMs or 5 VMs. The disk limits can be reached from a single VM or divided across two or more VMs.

Ultra Disk and Premium SSD v2 performance throttles

Both Ultra Disks and Premium SSD v2 managed disks have the unique capability of allowing you to set your performance by exposing modifiable attributes and allowing you to modify them. By default, there are only two modifiable attributes but, shared Ultra Disks and shared Premium SSD v2 managed disks have two more attributes. Ultra Disks and Premium SSD v2 split these attributes across each attached VM. For some examples on how this distribution of capacity, IOPS, and throughput works, see the Examples section.

Expand table

Attribute	Description
DiskIOPSReadWrite (Read/write disk IOPS)	The total number of IOPS allowed across all VMs mounting the shared disk with write access.
DiskMB/sReadWrite (Read/write disk throughput)	The total throughput (MB/s) allowed across all VMs mounting the shared disk with write access.
DiskIOPSReadOnly* (Read-only disk IOPS)	The total number of IOPS allowed across all VMs mounting the shared disk as `ReadOnly`.
DiskMB/sReadOnly* (Read-only disk throughput)	The total throughput (MB/s) allowed across all VMs mounting the shared disk as `ReadOnly`.

* Applies to shared Ultra Disks and shared Premium SSD v2 managed disks only

The following formulas explain how the performance attributes can be set, since they're user modifiable:

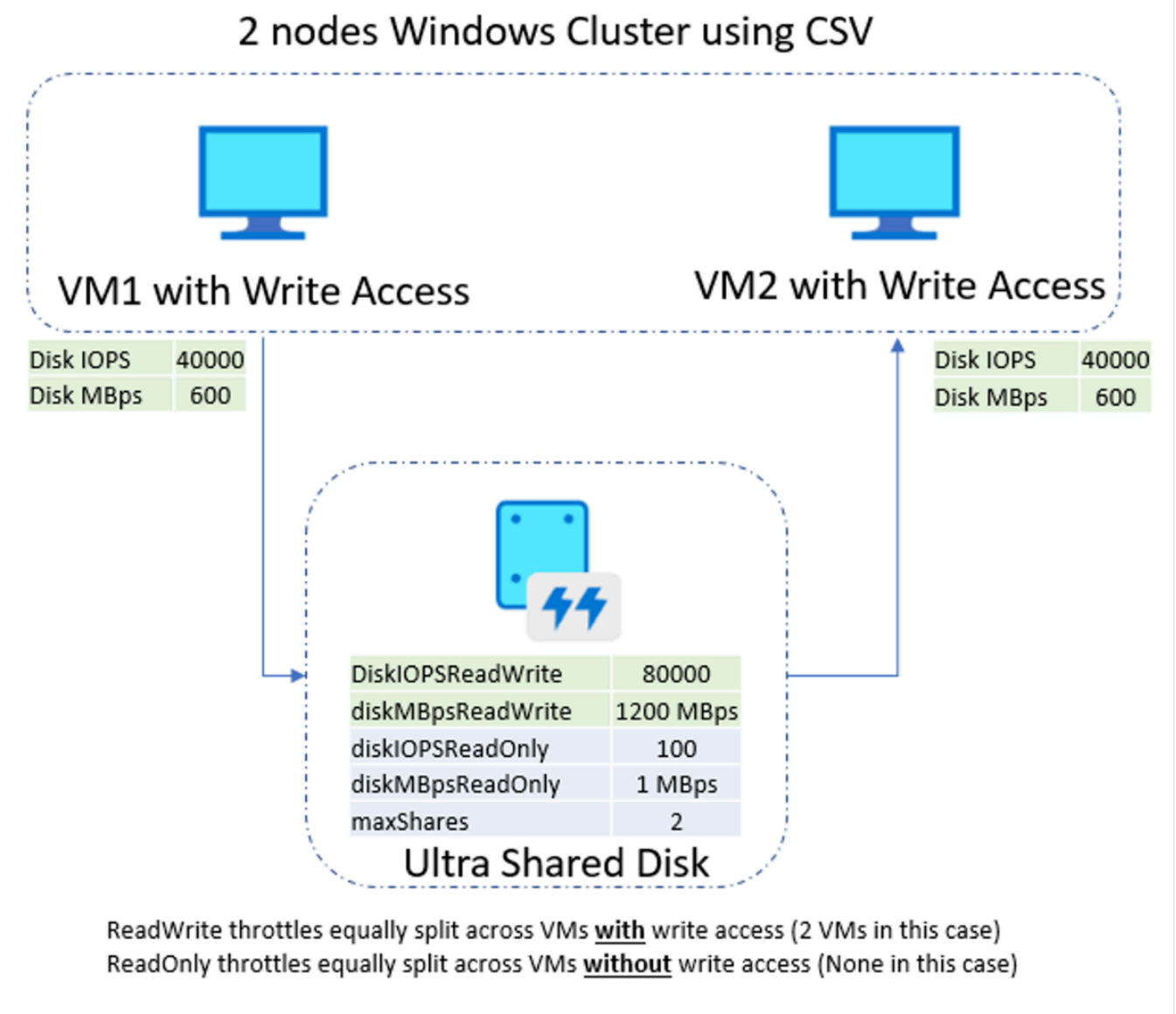
- DiskIOPSReadWrite (Read/write disk IOPS):
  - Has a baseline minimum IOPS of 100, for disks 100 GiB and smaller.
    - For disks larger than 100 GiB, the baseline minimum IOPS you can set increases by 1 per GiB. So the lowest you can set DiskIOPSReadWrite for a 101 GiB disk is 101 IOPS.
  - The maximum you can set this attribute is determined by the size of your disk, the formula is 300 * GiB, up to a maximum of 400,000.
- DiskMB/sReadWrite (Read/write disk throughput)
  - The minimum throughput (MB/s) of this attribute is determined by your IOPS, the formula is 4 KiB per second per IOPS. So if you had 101 IOPS, the minimum MB/s you can set is 1.
  - The maximum you can set this attribute is determined by the amount of IOPS you set, the formula is 256 KiB per second per IOPS, up to a maximum of 10,000 MB/s.
- DiskIOPSReadOnly (Read-only disk IOPS)
  - The minimum baseline IOPS for this attribute is 100. For DiskIOPSReadOnly, the baseline doesn't increase with disk size.
  - The maximum you can set this attribute is determined by the size of your disk, the formula is 300 * GiB, up to a maximum of 400,000.
- DiskMB/sReadOnly (Read-only disk throughput)
  - The minimum throughput (MB/s) for this attribute is 1. For DiskMB/sReadOnly, the baseline doesn't increase with IOPS.
  - The maximum you can set this attribute is determined by the amount of IOPS you set, the formula is 256 KiB per second per IOPS, up to a maximum of 10,000 MB/s.

Examples

The following examples depict a few scenarios that show how the throttling can work with shared Ultra Disks, specifically.

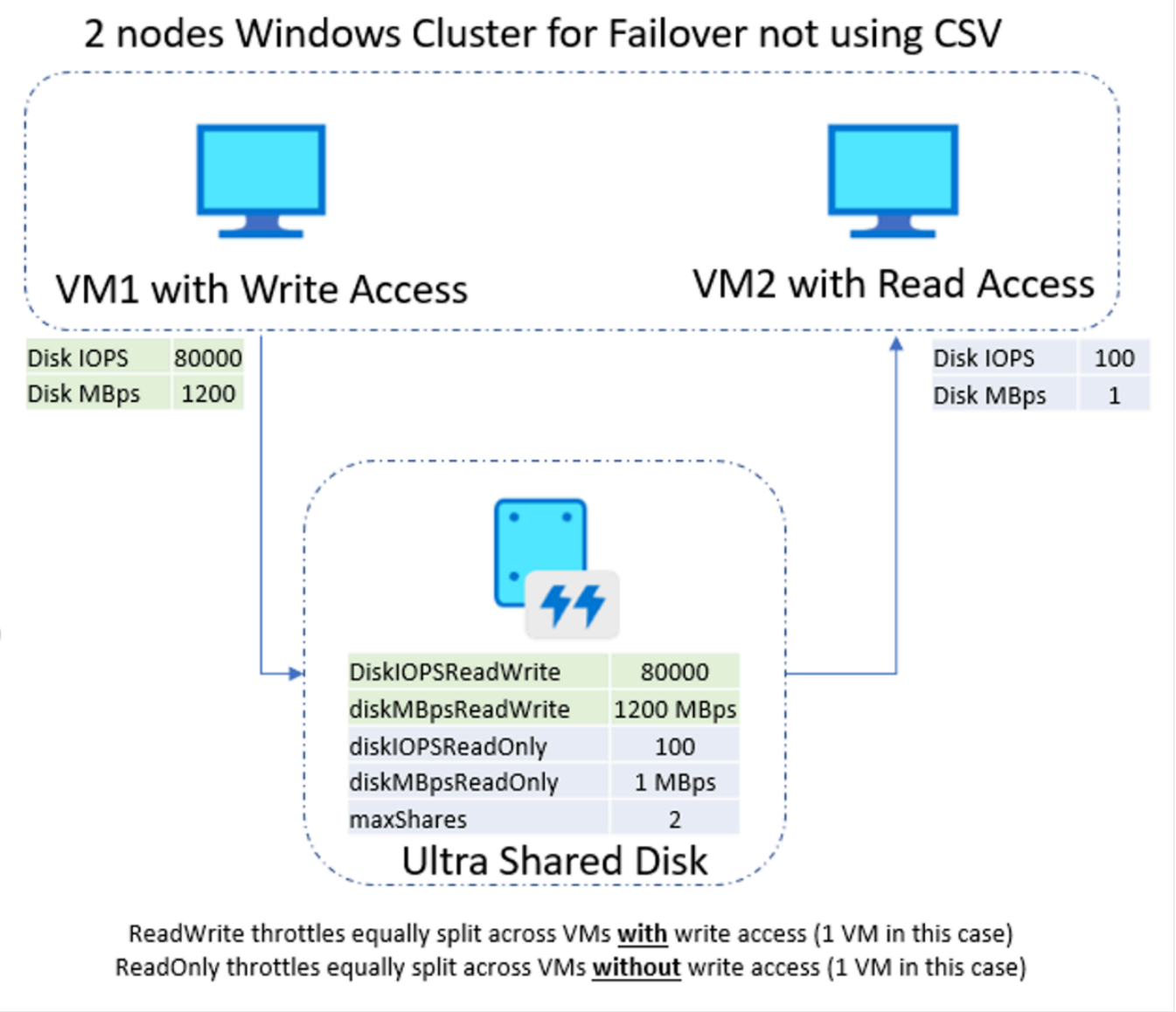
Two nodes cluster using cluster shared volumes

The following is an example of a 2-node WSFC using clustered shared volumes. With this configuration, both VMs have simultaneous write-access to the disk, which results in the `ReadWrite` throttle being split across the two VMs and the `ReadOnly` throttle not being used.



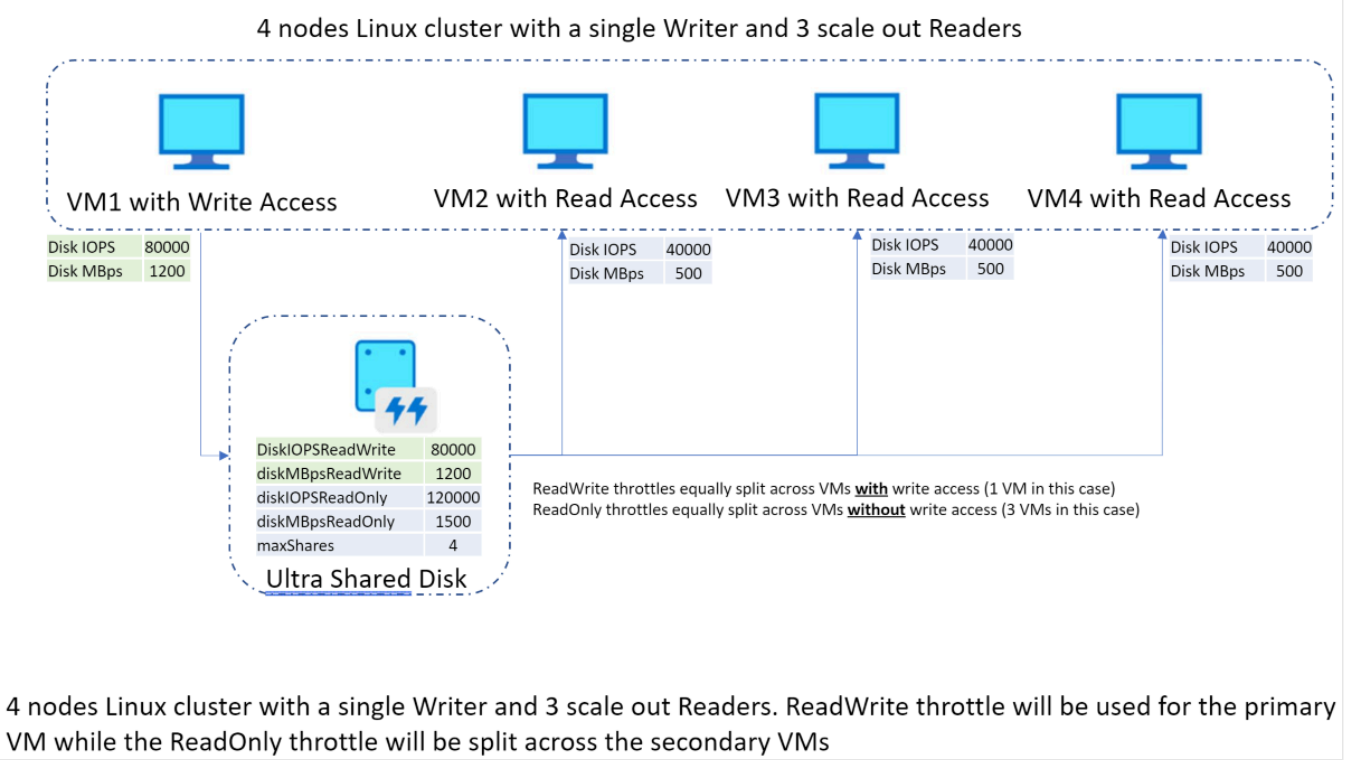
Two node cluster without cluster share volumes

The following is an example of a 2-node WSFC that isn't using clustered shared volumes. With this configuration, only one VM has write-access to the disk. This results in the `ReadWrite` throttle being used exclusively for the primary VM and the `ReadOnly` throttle only being used by the secondary.



Four node Linux cluster

The following is an example of a 4-node Linux cluster with a single writer and three scale-out readers. With this configuration, only one VM has write-access to the disk. This results in the `ReadWrite` throttle being used exclusively for the primary VM and the `ReadOnly` throttle being split by the secondary VMs.



4 nodes Linux cluster with a single Writer and 3 scale out Readers. ReadWrite throttle will be used for the primary VM while the ReadOnly throttle will be split across the secondary VMs

Shared Ultra Disk and Premium SSD v2 pricing

Both shared Ultra Disks and shared Premium SSD v2 managed disks are priced based on provisioned capacity, total provisioned IOPS (diskIOPSReadWrite + diskIOPSReadOnly) and total provisioned Throughput MB/s (diskMB/sReadWrite + diskMB/sReadOnly). There's no extra charge for each additional VM mount. For example, a shared Ultra Disk with the following configuration (diskSizeGB: 1024, DiskIOPSReadWrite: 10000, DiskMB/sReadWrite: 600, DiskIOPSReadOnly: 100, DiskMB/sReadOnly: 1) is charged with 1024 GiB, 10100 IOPS, and 601 MB/s regardless of whether it's mounted to two VMs or five VMs.

Next steps

If you're interested in enabling and using shared disks for your managed disks, proceed to our article Enable shared disk

If you have additional questions, see the shared disks section of the FAQ.

Last updated on 12/03/2025

Enable shared disks

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article explains how to enable the shared disks feature for Azure Managed Disks. With Azure shared disks, you can attach a managed disk to multiple virtual machines (VMs) simultaneously, enabling the deployment or migration of clustered applications to Azure.

If you're looking for conceptual information on managed disks that have shared disks enabled, see Azure shared disks.

Prerequisites

The scripts and commands in this article require either:

- Version 6.0.0 or newer of the Azure PowerShell module.

Or

- The latest version of the Azure CLI.

Limitations

General limitations

Shared disks have general limitations that apply to all shared disks, regardless of disk type. They also have more limitations that only apply to specific types of shared disks. The following list is the list of general limitations:

- Currently, only Ultra Disks, Premium SSD v2, Premium SSD, and Standard SSDs can be used as a shared disk
- Shared disks can be attached to individual Virtual Machine Scale Sets but can't be defined in the Virtual Machine Scale Set models or automatically deployed
- A shared disk can't be expanded without either deallocating all VMs the disk is attached to, or detaching the disk from all of these VMs
- Write accelerator isn't supported for shared disks
- Host caching isn't supported for shared disks

Each managed disk that has shared disks enabled are also subject to the following limitations, organized by disk type:

Ultra Disks

Ultra Disks have their own separate list of limitations, unrelated to shared disks. For Ultra Disk limitations, refer to Using Azure Ultra Disks.

When sharing Ultra Disks, they have the following additional limitations:

- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Can't be shared across availability zones.

Premium SSD v2

Premium SSD v2 managed disks have their own separate list of limitations, unrelated to shared disks. For these limitations, see Premium SSD v2 limitations.

When sharing Premium SSD v2 disks, they have the following additional limitation:

- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Can't be shared across availability zones.

Premium SSD

- Can only be enabled on data disks, not OS disks.
- Host caching isn't available for Premium SSDs with `maxShares>1`.
- Disk bursting isn't available for Premium SSDs with `maxShares>1`.
- When using Availability sets or Virtual Machine Scale Sets with Azure shared disks, storage fault domain alignment with virtual machine fault domain isn't enforced for the shared data disk.
- When using proximity placement groups (PPG), all virtual machines sharing a disk must be part of the same PPG.
- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Azure Site Recovery is supported in certain scenarios.
- Azure Backup is available through Azure Disk Backup.
- Only server-side encryption is supported, Azure Disk Encryption isn't currently supported.
- Can only be shared across availability zones if using Zone-redundant storage for managed disks.

Standard SSDs

- Can only be enabled on data disks, not OS disks.
- Host caching isn't available for Standard SSDs with `maxShares>1`.
- When using Availability sets and Virtual Machine Scale Sets with Azure shared disks, storage fault domain alignment with virtual machine fault domain isn't enforced for the shared data disk.
- When using proximity placement groups (PPG), all virtual machines sharing a disk must be part of the same PPG.
- Only basic disks can be used with some versions of Windows Server Failover Cluster, for details see Failover clustering hardware requirements and storage options.
- Azure Site Recovery is supported in certain scenarios.
- Azure Backup is available through Azure Disk Backup.
- Only server-side encryption is supported, Azure Disk Encryption isn't currently supported.
- Can only be shared across availability zones if using Zone-redundant storage for managed disks.

Supported operating systems

Shared disks support several operating systems. See the Windows and Linux sections of the conceptual article for the supported operating systems.

Disk sizes

For now, only Ultra Disks, Premium SSD v2, Premium SSD, and Standard SSDs can enable shared disks. Different disk sizes may have a different `maxShares` limit, which you can't exceed when setting the `maxShares` value.

For each disk, you can define a `maxShares` value that represents the maximum number of nodes that can simultaneously share the disk. For example, if you plan to set up a 2-node failover cluster, you would set `maxShares=2`. The maximum value is an upper bound. Nodes can join or leave the cluster (mount or unmount the disk) as long as the number of nodes is lower than the specified `maxShares` value.

> **Note**
>
> The `maxShares` value can only be set or edited when the disk is detached from all nodes.

Premium SSD ranges

The following table illustrates the allowed maximum values for `maxShares` by Premium SSD sizes:

Expand table

Disk sizes	maxShares limit
P1,P2,P3,P4,P6,P10,P15,P20	3
P30, P40, P50	5
P60, P70, P80	10

The IOPS and bandwidth limits for a disk aren't affected by the `maxShares` value. For example, the max IOPS of a P15 disk is 1100 whether maxShares = 1 or maxShares > 1.

Standard SSD ranges

The following table illustrates the allowed maximum values for `maxShares` by Standard SSD sizes:

Expand table

Disk sizes	maxShares limit
E1,E2,E3,E4,E6,E10,E15,E20	3
E30, E40, E50	5
E60, E70, E80	10

The IOPS and bandwidth limits for a disk aren't affected by the `maxShares` value. For example, the max IOPS of a E15 disk is 500 whether maxShares = 1 or maxShares > 1.

Ultra Disk ranges

The minimum `maxShares` value is 1, while the maximum `maxShares` value is 15. There are no size restrictions on Ultra Disks, any size Ultra Disk can use any value for `maxShares`, up to and including the maximum value.

Premium SSD v2 ranges

The minimum `maxShares` value is 1, while the maximum `maxShares` value is 15. There are no size restrictions on Premium SSD v2, any size Premium SSD v2 can use any value for `maxShares`, up to and including the maximum value.

Deploy shared disks

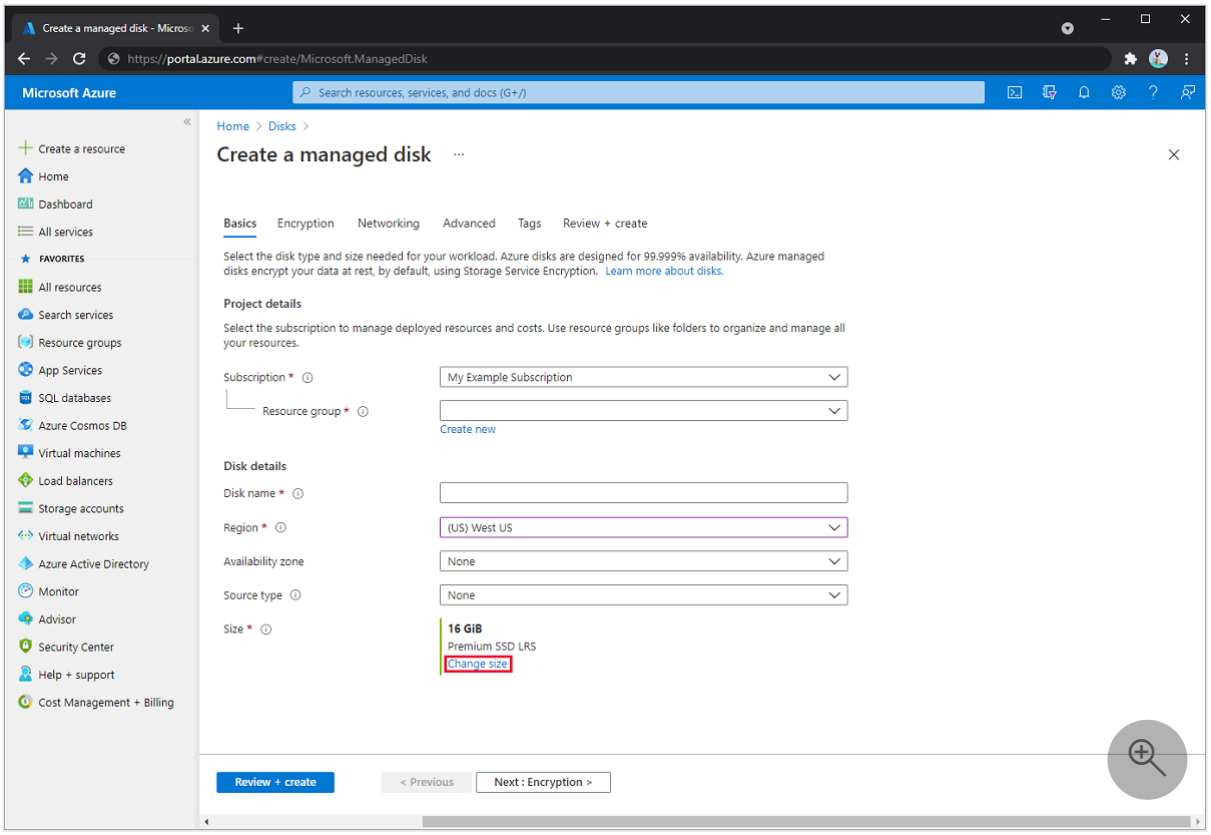
Deploy a Premium SSD as a shared disk

To deploy a managed disk with the shared disk feature enabled, use the new property `maxShares` and define a value greater than 1. This makes the disk shareable across multiple VMs.

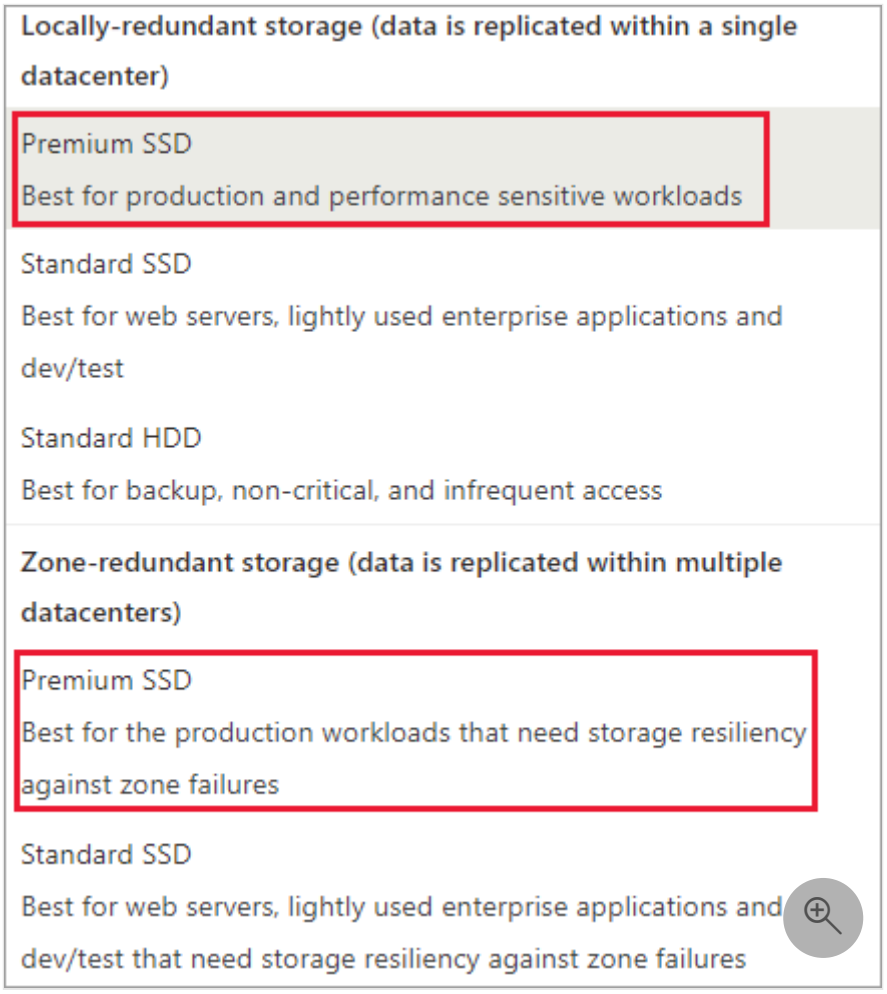
> **ⓘ Important**
>
> Host caching isn't supported for shared disks.
>
> The value of `maxShares` can only be set or changed when a disk is unmounted from all VMs. See the Disk sizes for the allowed values for `maxShares`.

Portal

1. Sign in to the Azure portal.
2. Search for and Select **Disks**.
3. Select **+ Create** to create a new managed disk.
4. Fill in the details and select an appropriate region, then select **Change size**.

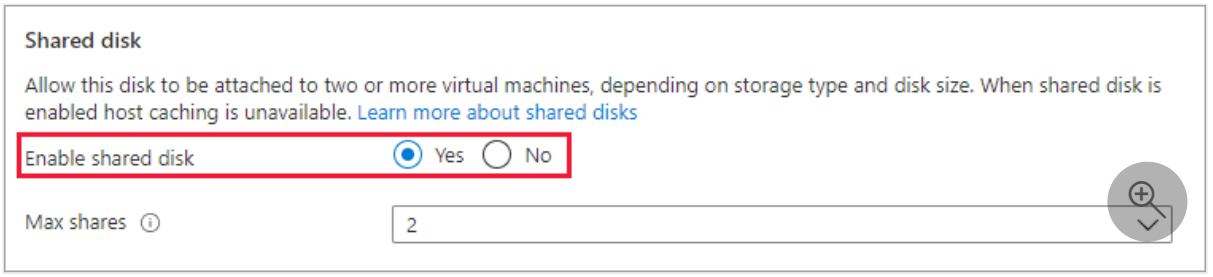


5. Select the Premium SSD size and SKU that you want and select **OK**.



6. Proceed through the deployment until you get to the **Advanced** pane.

7. Select **Yes** for **Enable shared disk** and select the amount of **Max shares** you want.



8. Select **Review + Create**.

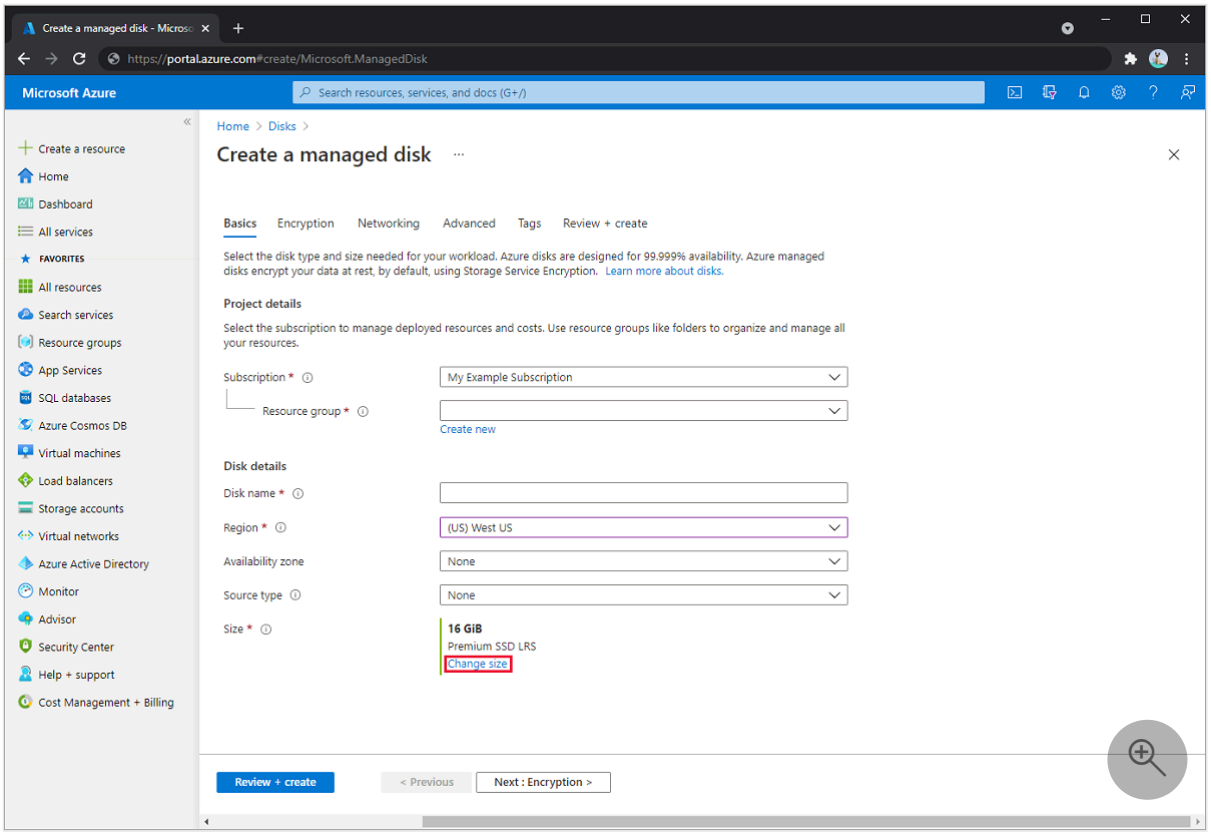
Deploy a Standard SSD as a shared disk

To deploy a managed disk with the shared disk feature enabled, use the new property `maxShares` and define a value greater than 1. This makes the disk shareable across multiple VMs.

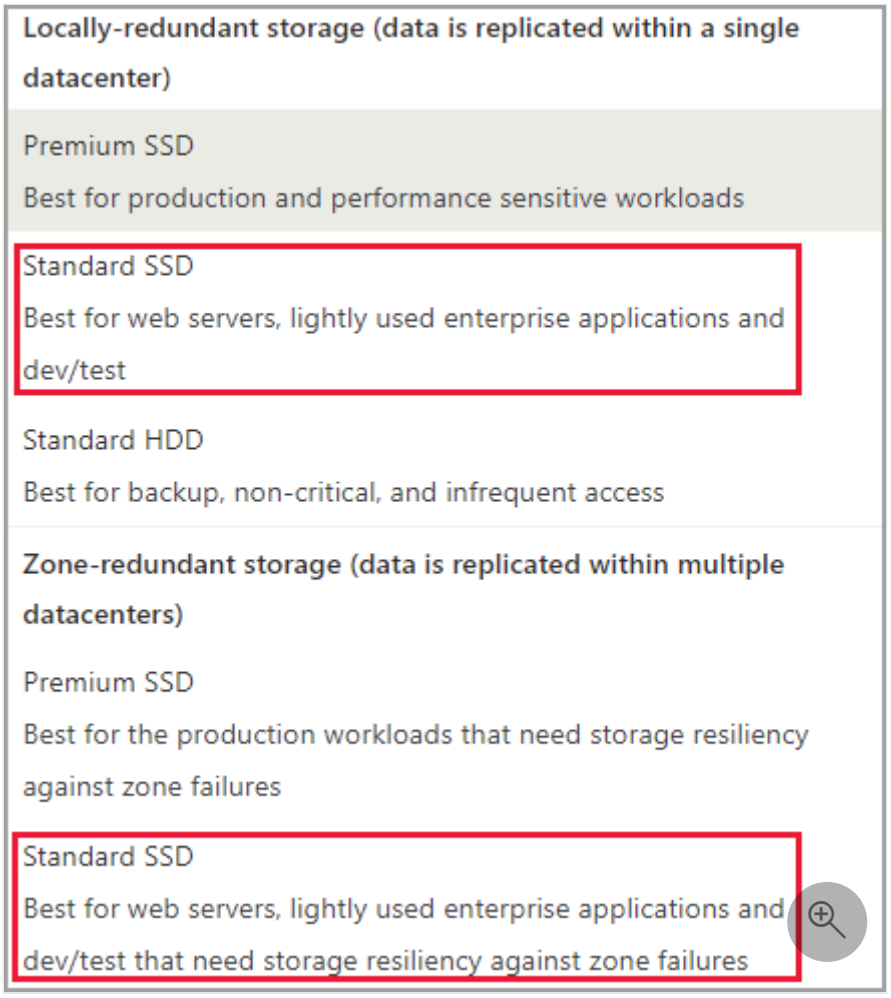
> **Important**
>
> Host caching isn't supported for shared disks.
>
> The value of `maxShares` can only be set or changed when a disk is unmounted from all VMs. See the Disk sizes for the allowed values for `maxShares`.

Portal

1. Sign in to the Azure portal.
2. Search for and Select **Disks**.
3. Select **+ Create** to create a new managed disk.
4. Fill in the details and select an appropriate region, then select **Change size**.

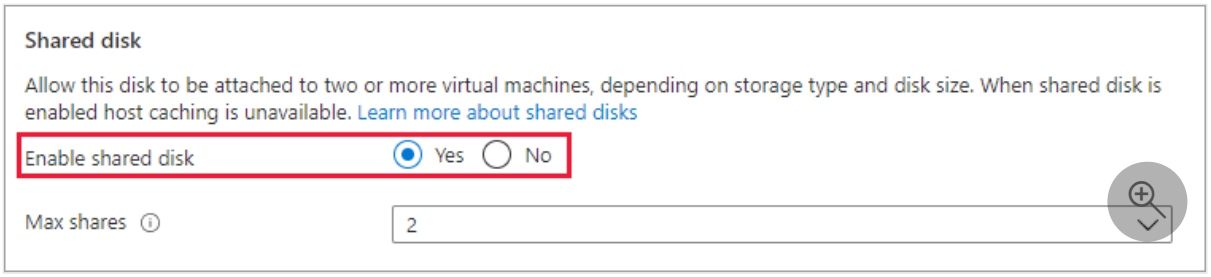


5. Select the Standard SSD size and SKU that you want and select **OK**.



6. Proceed through the deployment until you get to the **Advanced** pane.

7. Select **Yes** for **Enable shared disk** and select the amount of **Max shares** you want.



8. Select **Review + Create**.

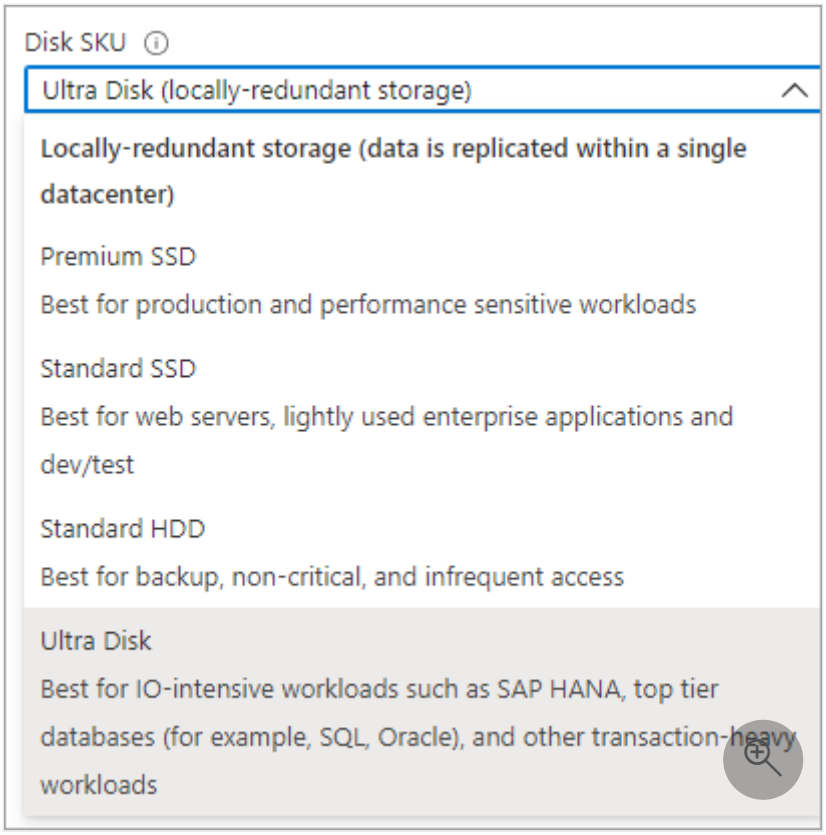
Deploy an Ultra Disk as a shared disk

To deploy a managed disk with the shared disk feature enabled, change the `maxShares` parameter to a value greater than 1. This makes the disk shareable across multiple VMs.

> **Important**
>
> The value of `maxShares` can only be set or changed when a disk is unmounted from all VMs. See the Disk sizes for the allowed values for `maxShares`.

Portal

1. Sign in to the Azure portal.
2. Search for and Select **Disks**.
3. Select **+ Create** to create a new managed disk.
4. Fill in the details, then select **Change size**.
5. Select Ultra Disk for the **Disk SKU**.

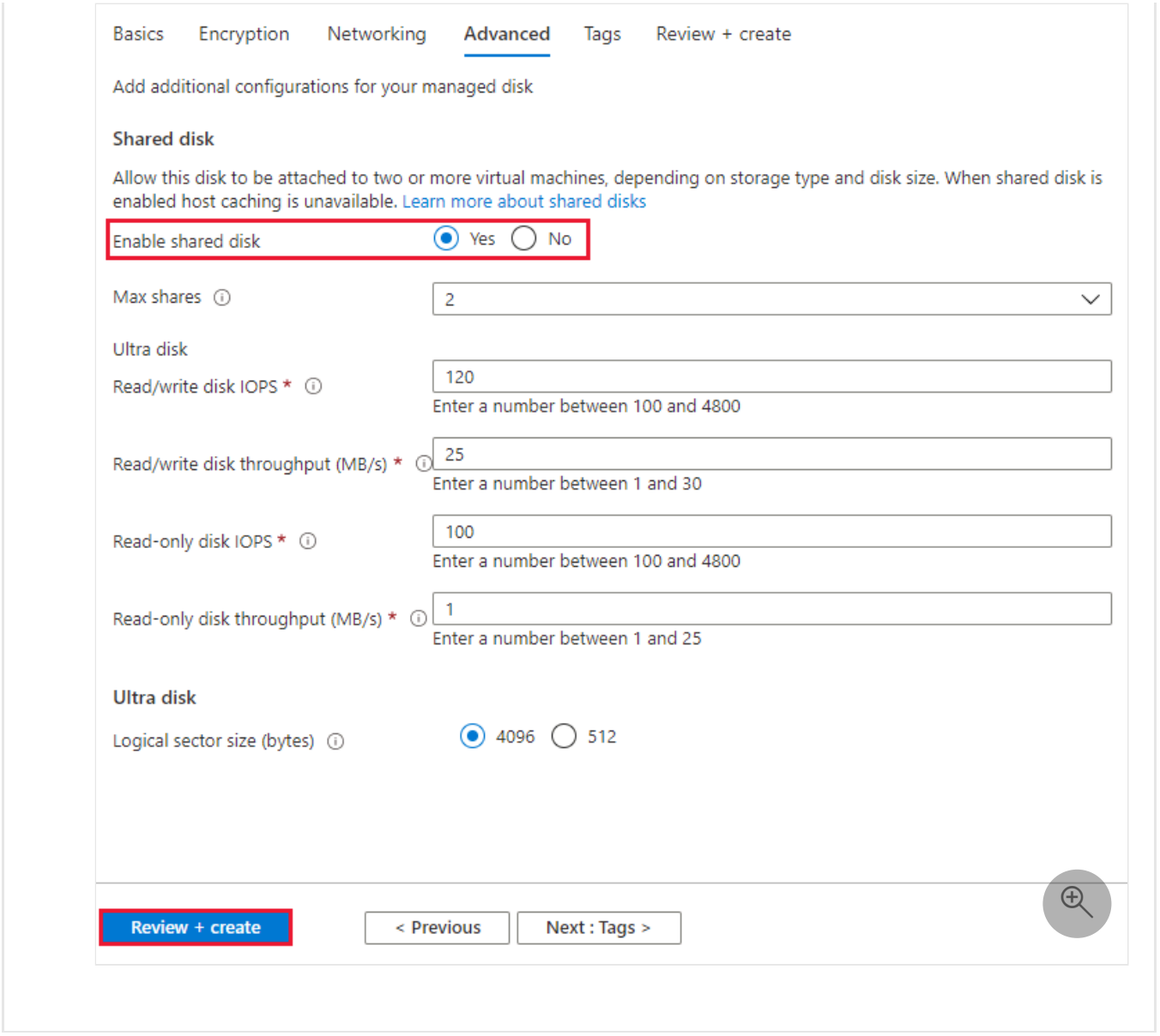


6. Select the disk size that you want and select **OK**.

7. Proceed through the deployment until you get to the **Advanced** pane.

8. Select **Yes** for **Enable shared disk** and select the amount of **Max shares** you want.

9. Select **Review + Create**.



Share an existing disk

To share an existing disk, or update how many VMs it can mount to, set the `maxShares` parameter with either the Azure PowerShell module or Azure CLI. You can also set `maxShares` to 1, if you want to disable sharing.

> **ⓘ Important**
>
> Host caching isn't supported for shared disks.
>
> The value of `maxShares` can only be set or changed when a disk is unmounted from all VMs. See the Disk sizes for the allowed values for `maxShares`. Before detaching a disk, record the LUN ID for when you reattach it.

PowerShell

Azure PowerShell

```
$datadiskconfig = Get-AzDisk -DiskName "mySharedDisk"
$datadiskconfig.maxShares = 3

Update-AzDisk -ResourceGroupName 'myResourceGroup' -DiskName 'mySharedDisk' -Disk
$datadiskconfig
```

CLI

Azure CLI

```
#Modifying a disk to enable or modify sharing configuration

az disk update --name mySharedDisk --max-shares 5 --resource-group myResourceGroup
```

Using Azure shared disks with your VMs

After you deploy a shared disk with `maxShares>1`, you can mount the disk to one or more of your VMs.

> **Note**
>
> Host caching isn't supported for shared disks.
>
> If you're deploying an Ultra Disk, make sure it matches the necessary requirements. See Using Azure Ultra Disks for details.

Azure PowerShell

```
$resourceGroup = "myResourceGroup"
$location = "WestCentralUS"

$vm = New-AzVm -ResourceGroupName $resourceGroup -Name "myVM" -Location $location -
VirtualNetworkName "myVnet" -SubnetName "mySubnet" -SecurityGroupName
"myNetworkSecurityGroup" -PublicIpAddressName "myPublicIpAddress"

$dataDisk = Get-AzDisk -ResourceGroupName $resourceGroup -DiskName "mySharedDisk"

$vm = Add-AzVMDataDisk -VM $vm -Name "mySharedDisk" -CreateOption Attach -
ManagedDiskId $dataDisk.Id -Lun 0

update-AzVm -VM $vm -ResourceGroupName $resourceGroup
```

Supported SCSI PR commands

Once you've mounted the shared disk to your VMs in your cluster, you can establish quorum and read/write to the disk using SCSI PR. The following PR commands are available when using Azure shared disks:

To interact with the disk, start with the persistent-reservation-action list:

```
PR_REGISTER_KEY

PR_REGISTER_AND_IGNORE

PR_GET_CONFIGURATION

PR_RESERVE

PR_PREEMPT_RESERVATION

PR_CLEAR_RESERVATION

PR_RELEASE_RESERVATION
```

When using PR_RESERVE, PR_PREEMPT_RESERVATION, or PR_RELEASE_RESERVATION, provide one of the following persistent-reservation-type:

```
PR_NONE

PR_WRITE_EXCLUSIVE

PR_EXCLUSIVE_ACCESS

PR_WRITE_EXCLUSIVE_REGISTRANTS_ONLY

PR_EXCLUSIVE_ACCESS_REGISTRANTS_ONLY

PR_WRITE_EXCLUSIVE_ALL_REGISTRANTS

PR_EXCLUSIVE_ACCESS_ALL_REGISTRANTS
```

You also need to provide a persistent-reservation-key when using PR_RESERVE, PR_REGISTER_AND_IGNORE, PR_REGISTER_KEY, PR_PREEMPT_RESERVATION, PR_CLEAR_RESERVATION, or PR_RELEASE-RESERVATION.

Next steps

If you have more questions, see the shared disks section of the FAQ.

Last updated on 02/26/2026

Overview of managed disk encryption options

09/24/2025

There are several types of encryption available for your managed disks, including Azure Disk Encryption (ADE), Server-Side Encryption (SSE), and encryption at host.

- **Azure Disk Storage Server-Side Encryption** (also referred to as encryption-at-rest or Azure Storage encryption) is always enabled and automatically encrypts data stored on Azure managed disks (OS and data disks) when persisting on the Storage Clusters. When configured with a Disk Encryption Set (DES), it supports customer-managed keys as well. It doesn't encrypt temp disks or disk caches. For full details, see Server-side encryption of Azure Disk Storage.

- **Encryption at host** is a Virtual Machine option that enhances Azure Disk Storage Server-Side Encryption to ensure that all temp disks and disk caches are encrypted at rest and flow encrypted to the Storage clusters. For full details, see Encryption at host - End-to-end encryption for your VM data.

- **Confidential disk encryption** binds disk encryption keys to the virtual machine's TPM and makes the protected disk content accessible only to the VM. The TPM and VM guest state is always encrypted in attested code using keys released by a secure protocol that bypasses the hypervisor and host operating system. Currently only available for the OS disk; temp disk support is in preview. Encryption at host may be used for other disks on a Confidential VM in addition to Confidential Disk Encryption. For full details, see DCasv5 and ECasv5 series confidential VMs.

- **Azure Disk Encryption** helps protect and safeguard your data to meet your organizational security and compliance commitments. ADE encrypts the OS and data disks of Azure virtual machines (VMs) inside your VMs by using the DM-Crypt feature of Linux or the BitLocker feature of Windows. ADE is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets, with the option to encrypt with a key encryption key (KEK). For full details, see Azure Disk Encryption for Linux VMs or Azure Disk Encryption for Windows VMs.

  > **Important**
  >
  > Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use **encryption at host** for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to

encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Encryption is part of a layered approach to security and should be used with other recommendations to secure Virtual Machines and their disks. For full details, see Security recommendations for virtual machines in Azure and Restrict import/export access to managed disks.

Comparison

Here's a comparison of Disk Storage SSE, ADE, encryption at host, and Confidential disk encryption.

Expand table

	Azure Disk Storage Server-Side Encryption	Encryption at Host	Azure Disk Encryption	Confidential disk encryption (For the OS disk only)
Encryption at rest (OS and data disks)	✅	✅	✅	✅
Temp disk encryption	❌	✅ Only supported with platform managed key	✅	✅ In Preview
Encryption of caches	❌	✅	✅	✅
Data flows encrypted between Compute and Storage	❌	✅	✅	✅
Customer control of keys	✅ When configured with DES	✅ When configured with DES	✅ When configured with KEK	✅ When configured with DES
HSM Support	Azure Key Vault Premium and Managed HSM	Azure Key Vault Premium and Managed HSM	Azure Key Vault Premium	Azure Key Vault Premium and Managed HSM
Does not use your VM's CPU	✅	✅	❌	❌

	Azure Disk Storage Server-Side Encryption	Encryption at Host	Azure Disk Encryption	Confidential disk encryption (For the OS disk only)
Works for custom images	✅	✅	❌ Does not work for custom Linux images	✅
Enhanced Key Protection	❌	❌	❌	✅
Microsoft Defender for Cloud disk encryption status*	Unhealthy	Healthy	Healthy	Not applicable

> **ⓘ Important**
>
> For Confidential disk encryption, Microsoft Defender for Cloud does not currently have a recommendation that is applicable.

* Microsoft Defender for Cloud has the following disk encryption recommendations:

- Virtual machines and virtual machine scale sets should have encryption at host enabled (Only detects Encryption at Host)
- Virtual machines should encrypt temp disks, caches, and data flows between Compute and Storage resources (Only detects Azure Disk Encryption)
- Windows virtual machines should enable Azure Disk Encryption or EncryptionAtHost (Detects both Azure Disk Encryption and EncryptionAtHost)
- Linux virtual machines should enable Azure Disk Encryption or EncryptionAtHost (Detects both Azure Disk Encryption and EncryptionAtHost)

Next steps

- Azure Disk Encryption for Linux VMs
- Azure Disk Encryption for Windows VMs
- Server-side encryption of Azure Disk Storage
- Encryption at host
- Migrate from Azure Disk Encryption to server-side encryption
- DCasv5 and ECasv5 series confidential VMs
- Azure Security Fundamentals - Azure encryption overview

Migrate from Azure Disk Encryption to encryption at host

> **ⓘ Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

This article provides step-by-step guidance for migrating your virtual machines from Azure Disk Encryption (ADE) to encryption at host. The migration process requires creating new disks and VMs, as in-place conversion is not supported.

Migration overview

Azure Disk Encryption (ADE) encrypts data within the VM using BitLocker (Windows) or dm-crypt (Linux), while encryption at host encrypts data at the VM host level without consuming VM CPU resources. Encryption at host enhances Azure's default server-side encryption (SSE) by providing end-to-end encryption for all VM data, including temp disks, caches, and data flows between compute and storage.

For more information, see Overview of managed disk encryption options and Enable end-to-end encryption using encryption at host.

Migration limitations and considerations

Before starting the migration process, be aware of these important limitations and considerations that affect your migration strategy:

- **No in-place migration**: You cannot directly convert ADE-encrypted disks to encryption at host. Migration requires creating new disks and VMs.
- **Linux OS disk limitation**: Disabling ADE on Linux OS disks is not supported. For Linux VMs with ADE-encrypted OS disks, you must create a new VM with a new OS disk.

- **Windows ADE encryption patterns**: On Windows VMs, Azure Disk Encryption can only encrypt the OS disk alone OR all disks (OS + data disks). It's not possible to encrypt only data disks on Windows VMs.
- **UDE flag persistence**: Disks encrypted with Azure Disk Encryption have a Unified Data Encryption (UDE) flag that persists even after decryption. Both snapshots and disk copies using the Copy option retain this UDE flag. The migration requires creating new managed disks using the Upload method and copying the VHD blob data, which creates a new disk object without any metadata from the source disk.
- **Downtime required**: The migration process requires VM downtime for disk operations and VM recreation.
- **Domain-joined VMs**: If your VMs are part of an Active Directory domain, more steps are required:
  - The original VM must be removed from the domain before deletion
  - After creating the new VM, it must be rejoined to the domain
  - For Linux VMs, domain joining can be accomplished using Azure AD extensions

  For more information, see What is Microsoft Entra Domain Services?

Prerequisites

Before starting the migration:

1. **Backup your data**: Create backups of all critical data before beginning the migration process.
2. **Test the process**: If possible, test the migration process on a nonproduction VM first.
3. **Prepare encryption resources**: Ensure your VM size supports encryption at host. Most current VM sizes support this feature. For more information about VM size requirements, see Enable end-to-end encryption using encryption at host.
4. **Document configuration**: Record your current VM configuration, including network settings, extensions, and attached resources.

Migration steps

The following migration steps work for most scenarios, with specific differences noted for each operating system.

> **Important**

Linux VMs with encrypted OS disks cannot be decrypted in-place. For these VMs, you must create a new VM with a new OS disk and migrate your data. See the **Migrating Linux VMs with encrypted OS disks** section after reviewing the general process below.

Disable Azure Disk Encryption

First step is to disable the existing Azure Disk Encryption when possible:

- **Windows**: Follow the instructions in Disable encryption and remove the encryption extension on Windows
- **Linux**: If **only data disks** are encrypted, follow Disable encryption and remove the encryption extension on Linux. If **the OS disk is encrypted**, see the Migrating Linux VMs with encrypted OS disks.

After running the ADE disable command, the VM encryption status in the Azure portal changes to "SSE + PMK" immediately. However, the actual decryption process at the OS level takes time and depends on the amount of data encrypted. You must verify that OS-level decryption is completed before proceeding to the next step.

For Windows VMs:

- Open Command Prompt as Administrator and run: `manage-bde -status`
- Verify that all volumes show "Fully Decrypted" status
- The decryption percentage should show 100% for all encrypted volumes

For Linux VMs (data disks only):

- Run: `sudo cryptsetup status /dev/mapper/<device-name>`
- Verify that encrypted devices are no longer active
- Check with: `lsblk` to confirm no encrypted mappings remain

Wait for complete decryption before continuing with disk migration to ensure data integrity.

Create new managed disks

Create new disks that don't carry over the ADE encryption metadata. This process works for both Windows and Linux VMs, with some specific considerations for Linux OS disks.

CLI

Important

You need to add an offset of 512 bytes when copying a managed disk from Azure. This is because Azure omits the footer when reporting the disk size. The copy will fail if you don't do this. The following script already does this for you.

If you are creating an OS disk, add `--hyper-v-generation <yourGeneration>` to `az disk create`.

Azure CLI

```
# Set variables
sourceDiskName="MySourceDisk"
sourceRG="MyResourceGroup"
targetDiskName="MyTargetDisk"
targetRG="MyResourceGroup"
targetLocation="eastus"
# For OS disks, specify either "Windows" or "Linux"
# For data disks, omit the targetOS variable and --os-type parameter
targetOS="Windows"

# Get source disk size in bytes
sourceDiskSizeBytes=$(az disk show -g $sourceRG -n $sourceDiskName --query
'[diskSizeBytes]' -o tsv)

# Create a new empty target disk with upload capability
az disk create -g $targetRG -n $targetDiskName -l $targetLocation --os-type
$targetOS --for-upload --upload-size-bytes $(($sourceDiskSizeBytes+512)) --sku
standard_lrs

# Generate SAS URIs for both disks
targetSASURI=$(az disk grant-access -n $targetDiskName -g $targetRG --access-
level Write --duration-in-seconds 86400 --query [accessSas] -o tsv)

sourceSASURI=$(az disk grant-access -n $sourceDiskName -g $sourceRG --access-
level Read --duration-in-seconds 86400 --query [accessSas] -o tsv)

# Copy the disk data using AzCopy
azcopy copy $sourceSASURI $targetSASURI --blob-type PageBlob

# Revoke SAS access when complete
az disk revoke-access -n $sourceDiskName -g $sourceRG

az disk revoke-access -n $targetDiskName -g $targetRG
```

This method creates new disks without the Azure Disk Encryption metadata (UDE flag), which is essential for a clean migration.

Create a new VM with encryption

Create a new VM using the newly created disks with your chosen encryption method.

You can choose from several encryption options, depending on your security requirements. This article provides steps for creating a new VM with encryption at host, which is the most common migration path. Other encryption options are covered in Overview of managed disk encryption options.

Create a new VM with encryption at host

Encryption at host provides the closest equivalent to Azure Disk Encryption's coverage, and is covered in this section.

CLI

For OS disks:

Azure CLI

```
# For Windows OS disks
az vm create
  --resource-group "MyResourceGroup"
  --name "MyVM-New"
  --os-type "Windows"
  --attach-os-disk "MyTargetDisk"
  --encryption-at-host true

# For Linux OS disks
# az vm create
#   --resource-group "MyResourceGroup"
#   --name "MyVM-New"
#   --os-type "Linux"
#   --attach-os-disk "MyTargetDisk"
#   --encryption-at-host true
```

For data disks:

Azure CLI

```
# Enable encryption at host on the VM
az vm update
  --resource-group "MyResourceGroup"
  --name "MyVM-New"
  --encryption-at-host true

# Attach the newly created data disk
az vm disk attach
  --resource-group "MyResourceGroup"
```

```
    --vm-name "MyVM-New"
    --name "MyTargetDisk"
```

Verify and configure the new disks

After creating the new VM with encryption at host, you need to verify and configure the disks properly for your operating system.

CLI

For Windows VMs:

- Verify disk letters are assigned correctly
- Check that applications can access the disks correctly
- Update any applications or scripts that reference specific disk IDs

For Linux VMs:

- Update `/etc/fstab` with the new disk UUIDs
- Mount the data disks to the correct mount points

Bash

```
# Get UUIDs of all disks
sudo blkid

# Mount all disks defined in fstab
sudo mount -a
```

Both Windows and Linux may require additional configuration steps specific to your applications or workloads.

Verify encryption and cleanup

Verify that encryption at host is properly configured on both Windows and Linux VMs.

CLI

Azure CLI

```
# Check encryption at host status
az vm show --resource-group "MyResourceGroup" --name "MyVM-New" --query
```

```
"securityProfile.encryptionAtHost"
```

After confirming that encryption at host is working properly:

1. Test VM functionality to ensure applications work correctly
2. Verify that data is accessible and intact
3. Delete the original resources when you're satisfied with the migration:

Azure CLI

```
# Delete the original VM
az vm delete --resource-group "MyResourceGroup" --name "MyVM-Original" --yes

# Delete the original disk
az disk delete --resource-group "MyResourceGroup" --name "MySourceDisk" --yes
```

Migrating Linux VMs with encrypted OS disks

Since you cannot disable encryption on Linux OS disks, the process is different from Windows.

CLI

1. Create a new VM with encryption at host enabled

   Azure CLI

   ```
   az vm create \
     --resource-group "MyResourceGroup" \
     --name "MyVM-New" \
     --image "Ubuntu2204" \
     --encryption-at-host true \
     --admin-username "azureuser" \
     --generate-ssh-keys
   ```

2. For data migration options:

   - For application data: Use SCP, rsync, or other file transfer methods to copy data
   - For configuration: Replicate important configuration files and settings
   - For complex applications: Use backup/restore procedures appropriate for your applications

   Azure CLI

   ```
   # Example of using SCP to copy files from source to new VM
   az vm run-command invoke -g MyResourceGroup -n MyVM-Original --command-id
   ```

```
RunShellScript \
  --scripts "scp -r /path/to/data azureuser@new-vm-ip:/path/to/destination"
```

After creating the new VM:

1. Configure the new VM to match the original environment
    - Set up the same network configurations
    - Install the same applications and services
    - Apply the same security settings
2. Test thoroughly before decommissioning the original VM

This approach works for both Windows and Linux VMs, but is especially important for Linux VMs with encrypted OS disks that cannot be decrypted in-place.

For guidance on data migration, see Upload a VHD to Azure and Copy files to a Linux VM using SCP.

Domain-joined VM considerations

If your VMs are members of an Active Directory domain, additional steps are required during the migration process:

Premigration domain steps

1. **Document domain membership**: Record the current domain, organizational unit (OU), and any special group memberships
2. **Note computer account**: The computer account in Active Directory needs to be managed
3. **Backup domain-specific configurations**: Save any domain-specific settings, group policies, or certificates

Domain removal process

1. **Remove from domain**: Before deleting the original VM, remove it from the domain using one of these methods:
    - Use `Remove-Computer` PowerShell cmdlet on Windows
    - Use the System Properties dialog to change to workgroup
    - Manually delete the computer account from Active Directory Users and Computers
2. **Clean up Active Directory**: Remove any orphaned computer accounts or DNS entries

Post-migration domain rejoining

1. **Join new VM to domain**: After creating the new VM with encryption at host:
    - For Windows: Use `Add-Computer` PowerShell cmdlet or System Properties
    - For Linux: Use Azure AD domain join extension or manual configuration
2. **Restore domain settings**: Reapply any domain-specific configurations, group policies, or certificates
3. **Verify domain functionality**: Test domain authentication, group policy application, and network resource access

Linux domain joining

For Linux VMs, you can use the Azure AD Domain Services VM extension:

Azure CLI

```
az vm extension set \
    --resource-group "MyResourceGroup" \
    --vm-name "MyLinuxVM-New" \
    --name "AADSSHLoginForLinux" \
    --publisher "Microsoft.Azure.ActiveDirectory"
```

For more information, see What is Microsoft Entra Domain Services?

Important domain considerations

- The new VM has a different computer SID, which may affect some applications
- Kerberos tickets and cached credentials must be refreshed
- Some domain-integrated applications may require reconfiguration
- Plan for potential temporary loss of domain services during migration

Post-migration verification

After completing the migration, verify that encryption at host is working correctly:

1. **Check encryption at host status**: Verify that encryption at host is enabled:

    Azure CLI

    ```
    az vm show --resource-group "MyResourceGroup" --name "MyVM-New" --query
    "securityProfile.encryptionAtHost"
    ```

2. **Test VM functionality**: Ensure your applications and services are working correctly.

3. **Verify disk encryption**: Confirm that disks are properly encrypted:

   Azure PowerShell

   ```
   Get-AzDisk -ResourceGroupName "MyResourceGroup" -DiskName "MyVM-OS-New" |
   Select-Object Name, DiskState
   ```

4. **Monitor performance**: Compare performance before and after migration to confirm the expected improvements.

For more information about encryption verification, see Enable end-to-end encryption using encryption at host.

Cleanup

After successful migration and verification:

1. **Delete old VM**: Remove the original ADE-encrypted VM

2. **Delete old disks**: Remove the original encrypted disks

3. **Update Key Vault access policies**: Other disk encryption solutions use standard Key Vault authorization mechanisms. If you no longer need the Key Vault for Azure Disk Encryption, update its access policies to disable the special disk encryption setting:

   CLI

   Azure CLI

   ```
   az keyvault update --name "YourKeyVaultName" --resource-group
   "YourResourceGroup" --enabled-for-disk-encryption false
   ```

---

4. **Clean up resources**: Remove any temporary resources created during migration
5. **Update documentation**: Update your infrastructure documentation to reflect the migration to encryption at host

Common issues and solutions

VM size doesn't support encryption at host

Solution: Check the list of supported VM sizes and resize your VM if necessary

VM fails to start after migration

Solution: Check that all disks are properly attached and that the OS disk is set as the boot disk

Encryption at host not enabled

Solution: Verify that the VM was created with the `--encryption-at-host true` parameter and that your subscription supports this feature

Performance issues persist

Solution: Verify that encryption at host is properly enabled and that the VM size supports the expected performance.

Next steps

- Overview of managed disk encryption options
- Enable end-to-end encryption using encryption at host - Azure portal
- Enable end-to-end encryption using encryption at host - Azure PowerShell
- Enable end-to-end encryption using encryption at host - Azure CLI
- Server-side encryption of Azure Disk Storage
- Azure Disk Encryption for Windows VMs
- Azure Disk Encryption for Linux VMs
- Azure Disk Encryption FAQ
- Upload a VHD to Azure or copy a managed disk to another region
- Azure Disk Encryption for Linux VMs
- Azure Disk Encryption FAQ
- Upload a VHD to Azure or copy a managed disk to another region
- Azure Disk Encryption FAQ
- Upload a VHD to Azure or copy a managed disk to another region
- Azure Disk Encryption for Linux VMs
- Azure Disk Encryption FAQ
- Upload a VHD to Azure or copy a managed disk to another region

---

Last updated on 11/17/2025

Server-side encryption of Azure Disk Storage

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Most Azure managed disks are encrypted with Azure Storage encryption, which uses server-side encryption (SSE) to protect your data and to help you meet your organizational security and compliance commitments. Azure Storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud. Disks with encryption at host enabled, however, aren't encrypted through Azure Storage. For disks with encryption at host enabled, the server hosting your VM provides the encryption for your data, and that encrypted data flows into Azure Storage.

Data in Azure managed disks is encrypted transparently using 256-bit AES encryption, one of the strongest block ciphers available, and is FIPS 140-2 compliant. For more information about the cryptographic modules underlying Azure managed disks, see Cryptography API: Next Generation

Azure Storage encryption doesn't impact the performance of managed disks and there's no extra cost. For more information about Azure Storage encryption, see Azure Storage encryption.

> **Important**
>
> Temporary disks are not managed disks and are not encrypted by SSE, unless you enable encryption at host.
>
> Azure VMs that are version 5 and above (such as Dsv5 or Dsv6) automatically encrypt their temporary disks and (if in use) their ephemeral OS disks with encryption at rest.

About encryption key management

You can rely on platform-managed keys for the encryption of your managed disk, or you can manage encryption using your own keys. If you choose to manage encryption with your own keys, you can specify a *customer-managed key* to use for encrypting and decrypting all data in managed disks.

The following sections describe each of the options for key management in greater detail.

Platform-managed keys

By default, managed disks use platform-managed encryption keys. All managed disks, snapshots, images, and data written to existing managed disks are automatically encrypted-at-rest with platform-managed keys. Platform-managed keys are managed by Microsoft.

Customer-managed keys

You can choose to manage encryption at the level of each managed disk, with your own keys. When you specify a customer-managed key, that key is used to protect and control access to the key that encrypts your data. Customer-managed keys offer greater flexibility to manage access controls.

You must use one of the following Azure key stores to store your customer-managed keys:

- Azure Key Vault (Premium tier recommended for HSM-backed key protection)
- Azure Key Vault Managed Hardware Security Module (HSM)

You can either import your RSA keys to your Key Vault or generate new RSA keys in Azure Key Vault. Azure managed disks handles the encryption and decryption in a fully transparent fashion using envelope encryption. It encrypts data using an AES 256 based data encryption key (DEK), which is, in turn, protected using your keys. The Storage service generates data encryption keys and encrypts them with customer-managed keys using RSA encryption. The envelope encryption allows you to rotate (change) your keys periodically as per your compliance policies without impacting your VMs. When you rotate your keys, the Storage service re-wraps the data encryption keys with the new customer-managed key version. The underlying disk data itself is not re-encrypted. Both old and new key versions must remain enabled until re-wrapping is complete.

Managed disks and the Key Vault or managed HSM must be in the same Azure region, but they can be in different subscriptions. They must also be in the same Microsoft Entra tenant, unless you're using Encrypt managed disks with cross-tenant customer-managed keys.

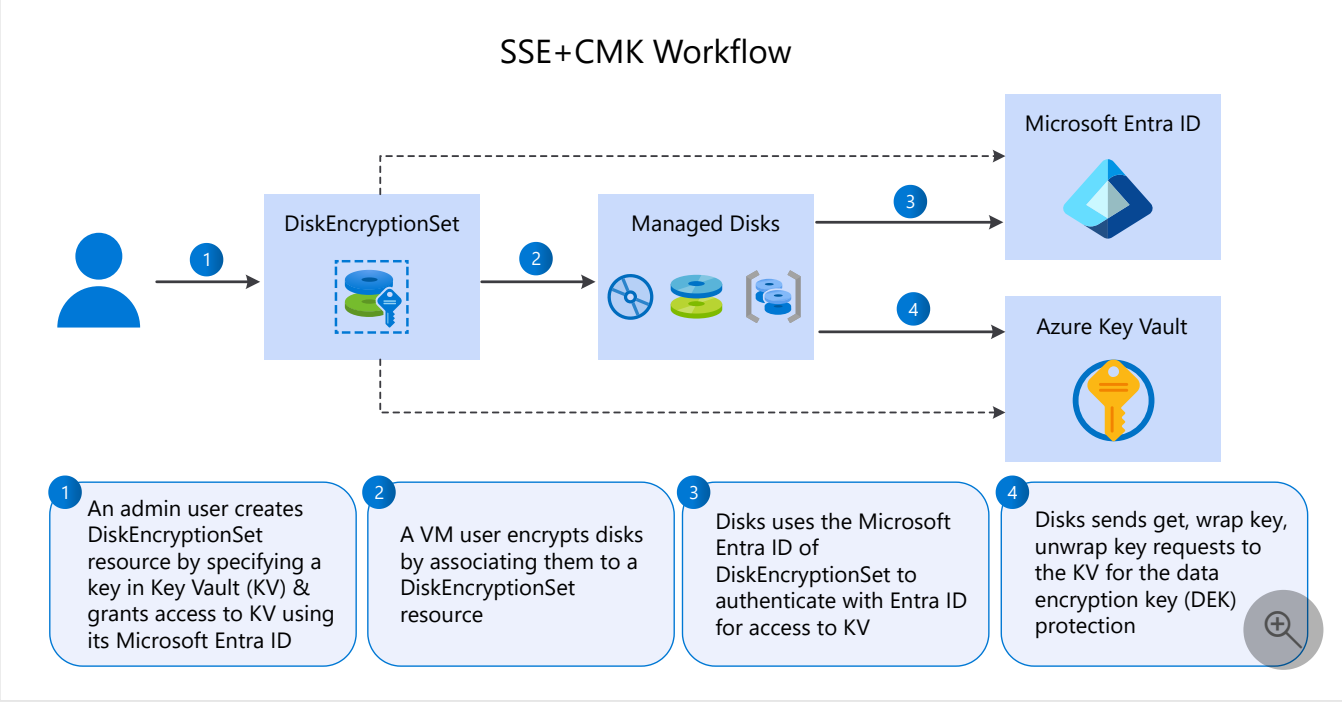
Full control of your keys

You must grant access to managed disks in your Key Vault or managed HSM to use your keys for encrypting and decrypting the DEK. This allows you full control of your data and keys. You can disable your keys or revoke access to managed disks at any time. You can also audit the

encryption key usage with Azure Key Vault monitoring to ensure that only managed disks or other trusted Azure services are accessing your keys.

> **Important**
>
> When a key is either disabled, deleted, or expired, any VMs with either OS or data disks using that key will automatically shut down. After the automated shut down, VMs won't boot until the key is enabled again, or you assign a new key.
>
> Generally, disk I/O (read or write operations) start to fail one hour after a key is either disabled, deleted, or expired.

The following diagram shows how managed disks use Microsoft Entra ID and Azure Key Vault to make requests using the customer-managed key:



The following list explains the diagram in more detail:

1. An Azure Key Vault administrator creates key vault resources.
2. The key vault admin either imports their RSA keys to Key Vault or generate new RSA keys in Key Vault.
3. That administrator creates an instance of Disk Encryption Set resource, specifying an Azure Key Vault ID and a key URL. Disk Encryption Set is a new resource introduced for simplifying the key management for managed disks.
4. When a disk encryption set is created, a system-assigned managed identity is created in Microsoft Entra ID and associated with the disk encryption set.

5. The Azure key vault administrator then grants the managed identity permission to perform operations in the key vault.
6. A VM user creates disks by associating them with the disk encryption set. The VM user can also enable server-side encryption with customer-managed keys for existing resources by associating them with the disk encryption set.
7. Managed disks use the managed identity to send requests to the Azure Key Vault.
8. For reading or writing data, managed disks sends requests to Azure Key Vault to encrypt (wrap) and decrypt (unwrap) the data encryption key in order to perform encryption and decryption of the data.

To revoke access to customer-managed keys, see Azure Key Vault PowerShell and Azure Key Vault CLI. Revoking access effectively blocks access to all data in the storage account, as the encryption key is inaccessible by Azure Storage.

Automatic key rotation of customer-managed keys

Generally, if you're using customer-managed keys, you should enable automatic key rotation to the latest key version. Automatic key rotation helps ensure your keys are secure. A disk references a key via its disk encryption set. When you enable automatic rotation for a disk encryption set, the system will automatically update all managed disks, snapshots, and images referencing the disk encryption set to use the new version of the key within one hour. To learn how to enable customer-managed keys with automatic key rotation, see Set up an Azure Key Vault and DiskEncryptionSet with automatic key rotation.

> **Note**
>
> Virtual Machines aren't rebooted during automatic key rotation.

If you can't enable automatic key rotation, you can use other methods to alert you before keys expire. This way, you can make sure to rotate your keys before expiration and keep business continuity. You can use either an Azure Policy or Azure Event Grid to send a notification when a key expires soon.

Restrictions

For now, customer-managed keys have the following restrictions:

- If this feature is enabled for a disk with incremental snapshots, it can't be disabled on that disk or its snapshots. To work around this, copy all the data to an entirely different

managed disk that isn't using customer-managed keys. You can do that with either the Azure CLI or the Azure PowerShell module.

- A disk and all of its associated incremental snapshots must have the same disk encryption set.
- Only software and HSM RSA keys of sizes 2,048-bit, 3,072-bit and 4,096-bit are supported, no other keys or sizes.
  - HSM keys require the **premium** tier of Azure Key vaults.
- For Ultra Disks and Premium SSD v2 disks only:
  - (Preview) User-assigned managed identities are available for Ultra Disks and Premium SSD v2 disks encrypted with customer-managed keys.
- Most resources related to your customer-managed keys (disk encryption sets, VMs, disks, and snapshots) must be in the same subscription and region.
  - Azure Key Vaults may be used from a different subscription but must be in the same region as your disk encryption set. Azure Key Vaults in different Microsoft Entra tenants are supported for managed disks. For details, see Encrypt managed disks with cross-tenant customer-managed keys.
- Disks encrypted with customer-managed keys can only move to another resource group if the VM they're attached to is deallocated.
- Disks, snapshots, and images encrypted with customer-managed keys can't be moved between subscriptions.
- Managed disks currently or previously encrypted using Azure Disk Encryption can't be encrypted using customer-managed keys.
- Can only create up to 5000 disk encryption sets per region per subscription.
- For information about using customer-managed keys with shared image galleries, see Preview: Use customer-managed keys for encrypting images.

Supported regions

Customer-managed keys are available in all regions that managed disks are available.

> **Important**
>
> Customer-managed keys rely on managed identities for Azure resources, a feature of Microsoft Entra ID. When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Microsoft Entra directory to another, the managed identity associated with managed disks isn't transferred to the new

tenant, so customer-managed keys may no longer work. For more information, see Transferring a subscription between Microsoft Entra directories.

To enable customer-managed keys for managed disks, see our articles covering how to enable it with either the Azure PowerShell module, the Azure CLI or the Azure portal.

See Create a managed disk from a snapshot with CLI for a code sample.

Encryption at host - End-to-end encryption for your VM data

When you enable encryption at host, that encryption starts on the VM host itself, the Azure server that your VM is allocated to. The data for your temporary disk and OS/data disk caches are stored on that VM host. After enabling encryption at host, all this data is encrypted at rest and flows encrypted to the Storage service, where it's persisted. Essentially, encryption at host encrypts your data from end-to-end. Encryption at host doesn't use your VM's CPU and doesn't impact your VM's performance.

Temporary disks and ephemeral OS disks are encrypted at rest with platform-managed keys when you enable end-to-end encryption. The OS and data disk caches are encrypted at rest with either customer-managed or platform-managed keys, depending on the selected disk encryption type. For example, if a disk is encrypted with customer-managed keys, then the cache for the disk is encrypted with customer-managed keys, and if a disk is encrypted with platform-managed keys then the cache for the disk is encrypted with platform-managed keys.

Restrictions

- Can't be enabled on virtual machines (VMs) or virtual machine scale sets that currently or ever had Azure Disk Encryption enabled.
- Azure Disk Encryption can't be enabled on disks that have encryption at host enabled.
- The encryption can be enabled on existing virtual machine scale sets. However, only new VMs created after enabling the encryption are automatically encrypted.
- Existing VMs must be deallocated and reallocated in order to be encrypted.

The following restrictions only apply to Ultra Disks and Premium SSD v2:

- Disks using a 512e sector size need to have been created after 5/13/2023.
  - If your disk was created before this date, snapshot your disk, and create a new disk using the snapshot.

Supported VM sizes

The complete list of supported VM sizes can be pulled programmatically. To learn how to retrieve them programmatically, refer to the finding supported VM sizes section of either the Azure PowerShell module or Azure CLI articles.

To enable end-to-end encryption using encryption at host, see our articles covering how to enable it with either the Azure PowerShell module, the Azure CLI, or the Azure portal.

Double encryption at rest

High security sensitive customers who are concerned of the risk associated with any particular encryption algorithm, implementation, or key being compromised can now opt for extra layer of encryption using a different encryption algorithm/mode at the infrastructure layer using platform managed encryption keys. This new layer can be applied to persisted OS and data disks, snapshots, and images, all of which will be encrypted at rest with double encryption.

Restrictions

Double encryption at rest isn't currently supported with either Ultra Disks or Premium SSD v2 disks.

To enable double encryption at rest for managed disks, see Enable double encryption at rest for managed disks.

Encryption at host versus Azure disk encryption

Azure Disk Encryption leverages either the DM-Crypt feature of Linux or the BitLocker feature of Windows to encrypt managed disks with customer-managed keys within the guest VM. Server-side encryption with encryption at host improves on ADE. With encryption at host, data for your temporary disk and OS/data disk caches are stored on that VM host. After enabling encryption at host, all this data is encrypted at rest and flows encrypted to the Storage service, where it's persisted. Essentially, encryption at host encrypts your data from end-to-end. Encryption at host doesn't use your VM's CPU and doesn't impact your VM's performance.

> **Important**
>
> Customer-managed keys rely on managed identities for Azure resources, a feature of Microsoft Entra ID. When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the

subscription, resource group, or managed disk from one Microsoft Entra directory to another, the managed identity associated with managed disks is not transferred to the new tenant, so customer-managed keys may no longer work. For more information, see Transferring a subscription between Microsoft Entra directories.

Next steps

- Enable end-to-end encryption using encryption at host with either the Azure PowerShell module, the Azure CLI, or the Azure portal.
- To enable double encryption at rest for managed disks, see Enable double encryption at rest for managed disks.
- Enable customer-managed keys for managed disks with either the Azure PowerShell module, the Azure CLI or the Azure portal.
- Migrate from Azure Disk Encryption to server-side encryption
- Explore the Azure Resource Manager templates for creating encrypted disks with customer-managed keys
- What is Azure Key Vault?

Last updated on 04/02/2026

Azure PowerShell - Enable customer-managed keys with server-side encryption - managed disks

Article • 08/23/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Disk Storage allows you to manage your own keys when using server-side encryption (SSE) for managed disks, if you choose. For conceptual information on SSE with customer-managed keys, and other managed disk encryption types, see the Customer-managed keys section of our disk encryption article.

Restrictions

For now, customer-managed keys have the following restrictions:

- If this feature is enabled for a disk with incremental snapshots, it can't be disabled on that disk or its snapshots. To work around this, copy all the data to an entirely different managed disk that isn't using customer-managed keys. You can do that with either the Azure CLI or the Azure PowerShell module.
- Only software and HSM RSA keys of sizes 2,048-bit, 3,072-bit and 4,096-bit are supported, no other keys or sizes.
  - HSM keys require the **premium** tier of Azure Key vaults.
- For Ultra Disks and Premium SSD v2 disks only:
  - Snapshots created from disks that are encrypted with server-side encryption and customer-managed keys must be encrypted with the same customer-managed keys.
  - User-assigned managed identities aren't supported for Ultra Disks and Premium SSD v2 disks encrypted with customer-managed keys.
  - Not currently supported in Azure Government or Azure China.
- Most resources related to your customer-managed keys (disk encryption sets, VMs, disks, and snapshots) must be in the same subscription and region.
  - Azure Key Vaults may be used from a different subscription but must be in the same region as your disk encryption set. As a preview, you can use Azure Key Vaults from different Microsoft Entra tenants.
- Disks encrypted with customer-managed keys can only move to another resource group if the VM they are attached to is deallocated.
- Disks, snapshots, and images encrypted with customer-managed keys can't be moved between subscriptions.

- Managed disks currently or previously encrypted using Azure Disk Encryption can't be encrypted using customer-managed keys.
- Can only create up to 5000 disk encryption sets per region per subscription.
- For information about using customer-managed keys with shared image galleries, see Preview: Use customer-managed keys for encrypting images.

Set up an Azure Key Vault and DiskEncryptionSet optionally with automatic key rotation

To use customer-managed keys with SSE, you must set up an Azure Key Vault and a DiskEncryptionSet resource.

1. Make sure that you have installed latest Azure PowerShell version, and you are signed in to an Azure account in with Connect-AzAccount

2. Create an instance of Azure Key Vault and encryption key.

   When creating the Key Vault instance, you must enable purge protection. Purge protection ensures that a deleted key cannot be permanently deleted until the retention period lapses. These settings protect you from losing data due to accidental deletion. These settings are mandatory when using a Key Vault for encrypting managed disks.

   PowerShell

   ```
   $ResourceGroupName="yourResourceGroupName"
   $LocationName="westcentralus"
   $keyVaultName="yourKeyVaultName"
   $keyName="yourKeyName"
   $keyDestination="Software"
   $diskEncryptionSetName="yourDiskEncryptionSetName"

   $keyVault = New-AzKeyVault -Name $keyVaultName `
   -ResourceGroupName $ResourceGroupName `
   -Location $LocationName `
   -EnablePurgeProtection

   $key = Add-AzKeyVaultKey -VaultName $keyVaultName `
         -Name $keyName `
         -Destination $keyDestination
   ```

3. Create an instance of a DiskEncryptionSet. You can set RotationToLatestKeyVersionEnabled equal to $true to enable automatic rotation of

the key. When you enable automatic rotation, the system will automatically update all managed disks, snapshots, and images referencing the disk encryption set to use the new version of the key within one hour.

PowerShell

```
$desConfig=New-AzDiskEncryptionSetConfig -Location $LocationName `
    -SourceVaultId $keyVault.ResourceId `
    -KeyUrl $key.Key.Kid `
    -IdentityType SystemAssigned `
    -RotationToLatestKeyVersionEnabled $false

$des=New-AzDiskEncryptionSet -Name $diskEncryptionSetName `
       -ResourceGroupName $ResourceGroupName `
       -InputObject $desConfig
```

4. Grant the DiskEncryptionSet resource access to the key vault.

> **Note**
>
> It may take few minutes for Azure to create the identity of your DiskEncryptionSet in your Microsoft Entra ID. If you get an error like "Cannot find the Active Directory object" when running the following command, wait a few minutes and try again.

PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ObjectId
$des.Identity.PrincipalId -PermissionsToKeys wrapkey,unwrapkey,get
```

Use a key vault in a different subscription

Alternatively, you can manage your Azure Key Vaults centrally from a single subscription, and use the keys stored in the Key Vault to encrypt managed disks and snapshots in other subscriptions in your organization. This allows your security team to enforce and easily manage a robust security policy to a single subscription.

> **Important**
>
> For this configuration, both your Key Vault and your disk encryption set must be in the same region and be using the same tenant.

The following script is an example of how you would configure a disk encryption set to use a key from a Key Vault in a different subscription, but same region:

Azure PowerShell

```
$sourceSubscriptionId="<sourceSubID>"
$sourceKeyVaultName="<sourceKVName>"
$sourceKeyName="<sourceKeyName>"

$targetSubscriptionId="<targetSubID>"
$targetResourceGroupName="<targetRGName>"
$targetDiskEncryptionSetName="<targetDiskEncSetName>"
$location="<targetRegion>"

Set-AzContext -Subscription $sourceSubscriptionId

$key = Get-AzKeyVaultKey -VaultName $sourceKeyVaultName -Name $sourceKeyName

Set-AzContext -Subscription $targetSubscriptionId

$desConfig=New-AzDiskEncryptionSetConfig -Location $location `
-KeyUrl $key.Key.Kid `
-IdentityType SystemAssigned `
-RotationToLatestKeyVersionEnabled $false

$des=New-AzDiskEncryptionSet -Name $targetDiskEncryptionSetName `
-ResourceGroupName $targetResourceGroupName `
-InputObject $desConfig
```

Examples

Now that you've created and configured these resources, you can use them to secure your managed disks. The following are example scripts, each with a respective scenario, that you can use to secure your managed disks.

Create a VM using a Marketplace image, encrypting the OS and data disks with customer-managed keys

Copy the script, replace all of the example values with your own parameters, and then run it.

PowerShell

```
$VMLocalAdminUser = "yourVMLocalAdminUserName"
$VMLocalAdminSecurePassword = ConvertTo-SecureString <password> -AsPlainText
-Force
$LocationName = "yourRegion"
```

```
$ResourceGroupName = "yourResourceGroupName"
$ComputerName = "yourComputerName"
$VMName = "yourVMName"
$VMSize = "yourVMSize"
$diskEncryptionSetName="yourdiskEncryptionSetName"

$NetworkName = "yourNetworkName"
$NICName = "yourNICName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -
AddressPrefix $SubnetAddressPrefix
$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName
$ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix
-Subnet $SingleSubnet
$NIC = New-AzNetworkInterface -Name $NICName -ResourceGroupName
$ResourceGroupName -Location $LocationName -SubnetId $Vnet.Subnets[0].Id

$Credential = New-Object System.Management.Automation.PSCredential
($VMLocalAdminUser, $VMLocalAdminSecurePassword);

$VirtualMachine = New-AzVMConfig -VMName $VMName -VMSize $VMSize
$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -
ComputerName $ComputerName -Credential $Credential -ProvisionVMAgent -
EnableAutoUpdate
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName
'MicrosoftWindowsServer' -Offer 'WindowsServer' -Skus '2012-R2-Datacenter' -
Version latest

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName
$ResourceGroupName -Name $diskEncryptionSetName

$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name $($VMName
+"_OSDisk") -DiskEncryptionSetId $diskEncryptionSet.Id -CreateOption
FromImage

$VirtualMachine = Add-AzVMDataDisk -VM $VirtualMachine -Name $($VMName
+"DataDisk1") -DiskSizeInGB 128 -StorageAccountType Premium_LRS -
CreateOption Empty -Lun 0 -DiskEncryptionSetId $diskEncryptionSet.Id

New-AzVM -ResourceGroupName $ResourceGroupName -Location $LocationName -VM
$VirtualMachine -Verbose
```

Create an empty disk encrypted using server-side encryption with customer-managed keys and attach it to a VM

Copy the script, replace all of the example values with your own parameters, and then run it.

PowerShell

```
$vmName = "yourVMName"
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$diskName = "yourDiskName"
$diskSKU = "Premium_LRS"
$diskSizeinGiB = 30
$diskLUN = 1
$diskEncryptionSetName="yourDiskEncryptionSetName"


$vm = Get-AzVM -Name $vmName -ResourceGroupName $ResourceGroupName

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName
$ResourceGroupName -Name $diskEncryptionSetName

$vm = Add-AzVMDataDisk -VM $vm -Name $diskName -CreateOption Empty -
DiskSizeInGB $diskSizeinGiB -StorageAccountType $diskSKU -Lun $diskLUN -
DiskEncryptionSetId $diskEncryptionSet.Id

Update-AzVM -ResourceGroupName $ResourceGroupName -VM $vm
```

Encrypt existing managed disks

Your existing disks must not be attached to a running VM in order for you to encrypt them using the following script:

PowerShell

```
$rgName = "yourResourceGroupName"
$diskName = "yourDiskName"
$diskEncryptionSetName = "yourDiskEncryptionSetName"

$diskEncryptionSet = Get-AzDiskEncryptionSet -ResourceGroupName $rgName -
Name $diskEncryptionSetName

New-AzDiskUpdateConfig -EncryptionType "EncryptionAtRestWithCustomerKey" -
DiskEncryptionSetId $diskEncryptionSet.Id | Update-AzDisk -ResourceGroupName
$rgName -DiskName $diskName
```

Encrypt an existing virtual machine scale set (uniform orchestration mode) with SSE and customer-managed

keys

This script will work for scale sets in uniform orchestration mode only. For scale sets in flexible orchestration mode, follow the Encrypt existing managed disks for each VM.

Copy the script, replace all the example values with your own parameters, and then run it:

PowerShell

```
#set variables
$vmssname = "name of the vmss that is already created"
$diskencryptionsetname = "name of the diskencryptionset already created"
$vmssrgname = "vmss resourcegroup name"
$diskencryptionsetrgname = "diskencryptionset resourcegroup name"

#get vmss object and create diskencryptionset object attach to vmss os disk
$ssevmss = get-azvmss -ResourceGroupName $vmssrgname -VMScaleSetName
$vmssname
$ssevmss.VirtualMachineProfile.StorageProfile.OsDisk.ManagedDisk.DiskEncrypt
ionSet = New-Object -TypeName
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSetParameters

#get diskencryption object and retrieve the resource id
$des = Get-AzDiskEncryptionSet -ResourceGroupName $diskencryptionsetrgname -
Name $diskencryptionsetname
write-host "the diskencryptionset resource id is:" $des.Id

#associate DES resource id to os disk and update vmss
$ssevmss.VirtualMachineProfile.StorageProfile.OsDisk.ManagedDisk.DiskEncrypt
ionSet.id = $des.Id
$ssevmss | update-azvmss
```

Create a virtual machine scale set using a Marketplace image, encrypting the OS and data disks with customer-managed keys

Copy the script, replace all of the example values with your own parameters, and then run it.

> **ⓘ Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to

[Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub](#)

PowerShell

```
$VMLocalAdminUser = "yourLocalAdminUser"
$VMLocalAdminSecurePassword = ConvertTo-SecureString Password@123 -
AsPlainText -Force
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$ComputerNamePrefix = "yourComputerNamePrefix"
$VMScaleSetName = "yourVMSSName"
$VMSize = "Standard_DS3_v2"
$diskEncryptionSetName="yourDiskEncryptionSetName"

$NetworkName = "yourVNETName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -
AddressPrefix $SubnetAddressPrefix

$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName
$ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix
-Subnet $SingleSubnet

$ipConfig = New-AzVmssIpConfig -Name "myIPConfig" -SubnetId
$Vnet.Subnets[0].Id

$VMSS = New-AzVmssConfig -Location $LocationName -SkuCapacity 2 -SkuName
$VMSize -UpgradePolicyMode 'Automatic' -OrchestrationMode 'Uniform'

$VMSS = Add-AzVmssNetworkInterfaceConfiguration -Name "myVMSSNetworkConfig"
-VirtualMachineScaleSet $VMSS -Primary $true -IpConfiguration $ipConfig

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName
$ResourceGroupName -Name $diskEncryptionSetName

# Enable encryption at rest with customer managed keys for OS disk by
setting DiskEncryptionSetId property

$VMSS = Set-AzVmssStorageProfile $VMSS -OsDiskCreateOption "FromImage" -
DiskEncryptionSetId $diskEncryptionSet.Id -ImageReferenceOffer
'WindowsServer' -ImageReferenceSku '2012-R2-Datacenter' -
ImageReferenceVersion latest -ImageReferencePublisher
'MicrosoftWindowsServer'

$VMSS = Set-AzVmssOsProfile $VMSS -ComputerNamePrefix $ComputerNamePrefix -
AdminUsername $VMLocalAdminUser -AdminPassword $VMLocalAdminSecurePassword

# Add a data disk encrypted at rest with customer managed keys by setting
DiskEncryptionSetId property
```

```
$VMSS = Add-AzVmssDataDisk -VirtualMachineScaleSet $VMSS -CreateOption Empty
-Lun 1 -DiskSizeGB 128 -StorageAccountType Premium_LRS -DiskEncryptionSetId
$diskEncryptionSet.Id

$Credential = New-Object System.Management.Automation.PSCredential
($VMLocalAdminUser, $VMLocalAdminSecurePassword);

New-AzVmss -VirtualMachineScaleSet $VMSS -ResourceGroupName
$ResourceGroupName -VMScaleSetName $VMScaleSetName
```

Change the key of a DiskEncryptionSet to rotate the key for all the resources referencing the DiskEncryptionSet

Copy the script, replace all of the example values with your own parameters, and then run it.

PowerShell

```
$ResourceGroupName="yourResourceGroupName"
$keyVaultName="yourKeyVaultName"
$keyName="yourKeyName"
$diskEncryptionSetName="yourDiskEncryptionSetName"

$keyVault = Get-AzKeyVault -VaultName $keyVaultName -ResourceGroupName
$ResourceGroupName

$keyVaultKey = Get-AzKeyVaultKey -VaultName $keyVaultName -Name $keyName

Update-AzDiskEncryptionSet -Name $diskEncryptionSetName -ResourceGroupName
$ResourceGroupName -SourceVaultId $keyVault.ResourceId -KeyUrl
$keyVaultKey.Id
```

Find the status of server-side encryption of a disk

PowerShell

```
$ResourceGroupName="yourResourceGroupName"
$DiskName="yourDiskName"

$disk=Get-AzDisk -ResourceGroupName $ResourceGroupName -DiskName $DiskName
$disk.Encryption.Type
```

Important

Customer-managed keys rely on managed identities for Azure resources, a feature of Microsoft Entra ID. When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Microsoft Entra directory to another, the managed identity associated with the managed disks is not transferred to the new tenant, so customer-managed keys may no longer work. For more information, see **Transferring a subscription between Microsoft Entra directories**.

Next steps

- Explore the Azure Resource Manager templates for creating encrypted disks with customer-managed keys
- Replicate machines with customer-managed keys enabled disks
- Set up disaster recovery of VMware VMs to Azure with PowerShell
- Set up disaster recovery to Azure for Hyper-V VMs using PowerShell and Azure Resource Manager

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use the Azure CLI to enable server-side encryption with customer-managed keys for managed disks

Article • 08/23/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Disk Storage allows you to manage your own keys when using server-side encryption (SSE) for managed disks, if you choose. For conceptual information on SSE with customer managed keys, as well as other managed disk encryption types, see the Customer-managed keys section of our disk encryption article.

Restrictions

For now, customer-managed keys have the following restrictions:

- If this feature is enabled for a disk with incremental snapshots, it can't be disabled on that disk or its snapshots. To work around this, copy all the data to an entirely different managed disk that isn't using customer-managed keys. You can do that with either the Azure CLI or the Azure PowerShell module.
- Only software and HSM RSA keys of sizes 2,048-bit, 3,072-bit and 4,096-bit are supported, no other keys or sizes.
  - HSM keys require the **premium** tier of Azure Key vaults.
- For Ultra Disks and Premium SSD v2 disks only:
  - Snapshots created from disks that are encrypted with server-side encryption and customer-managed keys must be encrypted with the same customer-managed keys.
  - User-assigned managed identities aren't supported for Ultra Disks and Premium SSD v2 disks encrypted with customer-managed keys.
  - Not currently supported in Azure Government or Azure China.
- Most resources related to your customer-managed keys (disk encryption sets, VMs, disks, and snapshots) must be in the same subscription and region.
  - Azure Key Vaults may be used from a different subscription but must be in the same region as your disk encryption set. As a preview, you can use Azure Key Vaults from different Microsoft Entra tenants.
- Disks encrypted with customer-managed keys can only move to another resource group if the VM they are attached to is deallocated.

- Disks, snapshots, and images encrypted with customer-managed keys can't be moved between subscriptions.
- Managed disks currently or previously encrypted using Azure Disk Encryption can't be encrypted using customer-managed keys.
- Can only create up to 5000 disk encryption sets per region per subscription.
- For information about using customer-managed keys with shared image galleries, see Preview: Use customer-managed keys for encrypting images.

Create resources

Once the feature is enabled, you'll need to set up a DiskEncryptionSet and either an Azure Key Vault or an Azure Key Vault Managed HSM.

Azure Key Vault

- Install the latest Azure CLI and log to an Azure account in with az login.
- Create an Azure Key Vault and encryption key.

When creating the Key Vault, you must enable purge protection. Purge protection ensures that a deleted key cannot be permanently deleted until the retention period lapses. These settings protect you from losing data due to accidental deletion. These settings are mandatory when using a Key Vault for encrypting managed disks.

> **ⓘ Important**
>
> Don't camel case the region, if you do so, you may experience problems when assigning additional disks to the resource in the Azure portal.

Azure CLI

```
subscriptionId=yourSubscriptionID
rgName=yourResourceGroupName
location=westcentralus
keyVaultName=yourKeyVaultName
keyName=yourKeyName
diskEncryptionSetName=yourDiskEncryptionSetName
diskName=yourDiskName

az account set --subscription $subscriptionId

az group create --resource-group $rgName --location $location

az keyvault create -n $keyVaultName \
-g $rgName \
```

```
-l $location \
--enable-purge-protection true

az keyvault key create --vault-name $keyVaultName \
-n $keyName \
--protection software
```

- Create a DiskEncryptionSet. You can set enable-auto-key-rotation equal to true to enable automatic rotation of the key. When you enable automatic rotation, the system will automatically update all managed disks, snapshots, and images referencing the disk encryption set to use the new version of the key within one hour.

Azure CLI

```
keyVaultKeyUrl=$(az keyvault key show --vault-name $keyVaultName --name
$keyName --query [key.kid] -o tsv)

az disk-encryption-set create -n $diskEncryptionSetName \
-l $location \
-g $rgName \
--key-url $keyVaultKeyUrl \
--enable-auto-key-rotation false
```

- Grant the DiskEncryptionSet resource access to the key vault.

> **Note**
>
> It may take few minutes for Azure to create the identity of your DiskEncryptionSet in your Microsoft Entra ID. If you get an error like "Cannot find the Active Directory object" when running the following command, wait a few minutes and try again.

Azure CLI

```
desIdentity=$(az disk-encryption-set show -n $diskEncryptionSetName -g
$rgName --query [identity.principalId] -o tsv)

az keyvault set-policy -n $keyVaultName \
-g $rgName \
--object-id $desIdentity \
--key-permissions wrapkey unwrapkey get
```

Azure Key Vault Managed HSM

Alternatively, you can use a Managed HSM to handle your keys.

To do this, you must complete the following prerequisites:

- Install the latest Azure CLI and log in to an Azure account in with az login.
- Create and configure a managed HSM.
- Assign permissions to a user, so they can manage your Managed HSM.

Configuration

Once you've created a Managed HSM and added permissions, enable purge protection and create an encryption key.

Azure CLI

```
subscriptionId=yourSubscriptionID
rgName=yourResourceGroupName
location=westcentralus
keyVaultName=yourKeyVaultName
keyName=yourKeyName
diskEncryptionSetName=yourDiskEncryptionSetName
diskName=yourDiskName

az account set --subscription $subscriptionId

az keyvault update-hsm --subscription $subscriptionId -g $rgName --hsm-name
$keyVaultName --enable-purge-protection true

az keyvault key create --hsm-name  $keyVaultName --name $keyName --ops
wrapKey unwrapKey --kty RSA-HSM --size 2048
```

Then, create a DiskEncryptionSet.

Azure CLI

```
keyVaultKeyUrl=$(az keyvault key show --vault-name $keyVaultName --name
$keyName --query [key.kid] -o tsv)

az disk-encryption-set create -n $diskEncryptionSetName \
-l $location \
-g $rgName \
--key-url $keyVaultKeyUrl \
--enable-auto-key-rotation false
```

Finally, grant the DiskEncryptionSet access to the Managed HSM.

Azure CLI

```
desIdentity=$(az disk-encryption-set show -n $diskEncryptionSetName -g
$rgName --query [identity.principalId] -o tsv)
```

```
az keyvault role assignment create --hsm-name $keyVaultName --role "Managed
HSM Crypto Service Encryption User" --assignee $desIdentity --scope /keys
```

Now that you've created and configured these resources, you can use them to secure your managed disks. The following links contain example scripts, each with a respective scenario, that you can use to secure your managed disks.

Examples

Create a VM using a Marketplace image, encrypting the OS and data disks with customer-managed keys

Azure CLI

```
rgName=yourResourceGroupName
vmName=yourVMName
location=westcentralus
vmSize=Standard_DS3_V2
image=LinuxImageURN
diskEncryptionSetName=yourDiskencryptionSetName

diskEncryptionSetId=$(az disk-encryption-set show -n $diskEncryptionSetName
-g $rgName --query [id] -o tsv)

az vm create -g $rgName -n $vmName -l $location --image $image --size
$vmSize --generate-ssh-keys --os-disk-encryption-set $diskEncryptionSetId --
data-disk-sizes-gb 128 128 --data-disk-encryption-sets $diskEncryptionSetId
$diskEncryptionSetId
```

Encrypt existing managed disks

Your existing disks must not be attached to a running VM in order for you to encrypt them using the following script:

Azure CLI

```
rgName=yourResourceGroupName
diskName=yourDiskName
diskEncryptionSetName=yourDiskEncryptionSetName

az disk update -n $diskName -g $rgName --encryption-type
EncryptionAtRestWithCustomerKey --disk-encryption-set $diskEncryptionSetId
```

Create a virtual machine scale set using a Marketplace image, encrypting the OS and data disks with customer-managed keys

Azure CLI

```
rgName=yourResourceGroupName
vmssName=yourVMSSName
location=westcentralus
vmSize=Standard_DS3_V2
image=LinuxImageURN
diskEncryptionSetName=yourDiskencryptionSetName

diskEncryptionSetId=$(az disk-encryption-set show -n $diskEncryptionSetName
-g $rgName --query [id] -o tsv)
az vmss create -g $rgName -n $vmssName --image $image --upgrade-policy
automatic --admin-username azureuser --generate-ssh-keys --os-disk-
encryption-set $diskEncryptionSetId --data-disk-sizes-gb 64 128 --data-disk-
encryption-sets $diskEncryptionSetId $diskEncryptionSetId
```

Create an empty disk encrypted using server-side encryption with customer-managed keys and attach it to a VM

Azure CLI

```
vmName=yourVMName
rgName=yourResourceGroupName
diskName=yourDiskName
diskSkuName=Premium_LRS
diskSizeinGiB=30
location=westcentralus
diskLUN=2
diskEncryptionSetName=yourDiskEncryptionSetName


diskEncryptionSetId=$(az disk-encryption-set show -n $diskEncryptionSetName
-g $rgName --query [id] -o tsv)

az disk create -n $diskName -g $rgName -l $location --encryption-type
EncryptionAtRestWithCustomerKey --disk-encryption-set $diskEncryptionSetId -
-size-gb $diskSizeinGiB --sku $diskSkuName

diskId=$(az disk show -n $diskName -g $rgName --query [id] -o tsv)

az vm disk attach --vm-name $vmName --lun $diskLUN --ids $diskId
```

Change the key of a DiskEncryptionSet to rotate the key for all the resources referencing the DiskEncryptionSet

Azure CLI

```
rgName=yourResourceGroupName
keyVaultName=yourKeyVaultName
keyName=yourKeyName
diskEncryptionSetName=yourDiskEncryptionSetName


keyVaultId=$(az keyvault show --name $keyVaultName--query [id] -o tsv)

keyVaultKeyUrl=$(az keyvault key show --vault-name $keyVaultName --name
$keyName --query [key.kid] -o tsv)

az disk-encryption-set update -n keyrotationdes -g keyrotationtesting --key-
url $keyVaultKeyUrl --source-vault $keyVaultId
```

Find the status of server-side encryption of a disk

Azure CLI

```
az disk show -g yourResourceGroupName -n yourDiskName --query
[encryption.type] -o tsv
```

> **Important**
>
> Customer-managed keys rely on managed identities for Azure resources, a feature of Microsoft Entra ID. When you configure customer-managed keys, a managed identity is automatically assigned to your resources under the covers. If you subsequently move the subscription, resource group, or managed disk from one Microsoft Entra directory to another, the managed identity associated with the managed disks is not transferred to the new tenant, so customer-managed keys may no longer work. For more information, see Transferring a subscription between Microsoft Entra directories.

Next steps

- Explore the Azure Resource Manager templates for creating encrypted disks with customer-managed keys

- Replicate machines with customer-managed keys enabled disks
- Set up disaster recovery of VMware VMs to Azure with PowerShell
- Set up disaster recovery to Azure for Hyper-V VMs using PowerShell and Azure Resource Manager
- See Create a managed disk from a snapshot with CLI for a code sample.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Encrypt managed disks with cross-tenant customer-managed keys

06/12/2025

This article covers building a solution where you encrypt managed disks with customer-managed keys using Azure Key Vaults stored in a different Microsoft Entra tenant. This configuration can be ideal for several scenarios, one example being Azure support for service providers that want to offer bring-your-own encryption keys to their customers where resources from the service provider's tenant are encrypted with keys from their customer's tenant.

A disk encryption set with federated identity in a cross-tenant CMK workflow spans service provider/ISV tenant resources (disk encryption set, managed identities, and app registrations) and customer tenant resources (enterprise apps, user role assignments, and key vault). In this case, the source Azure resource is the service provider's disk encryption set.

If you have questions about cross-tenant customer-managed keys with managed disks, email crosstenantcmkvteam@service.microsoft.com.

Limitations

- Managed Disks and the customer's Key Vault must be in the same Azure region, but they can be in different subscriptions.

Preview - Ultra Disk and Premium SSD v2

This configuration is also available for Ultra Disks and Premium SSD v2 disks, as a preview configuration, in select regions.

Sign up for the preview using either the PowerShell:

Azure PowerShell

```
Register-AzProviderFeature -FeatureName UserAssignedIdentityForDirectDriveDisks -
ProviderNamespace Microsoft.Compute
Register-AzProviderFeature -FeatureName CrossTenantCMKForDirectDriveDisks -
ProviderNamespace Microsoft.Compute
```

Or the CLI:

Azure CLI

```
az feature register --name UserAssignedIdentityForDirectDriveDisks  --namespace
Microsoft.Compute
az feature register --name CrossTenantCMKForDirectDriveDisks  --namespace
Microsoft.Compute
```

Preview regional availability

The preview for Ultra Disks and Premium SSD v2 disks is currently only available in the following regions:

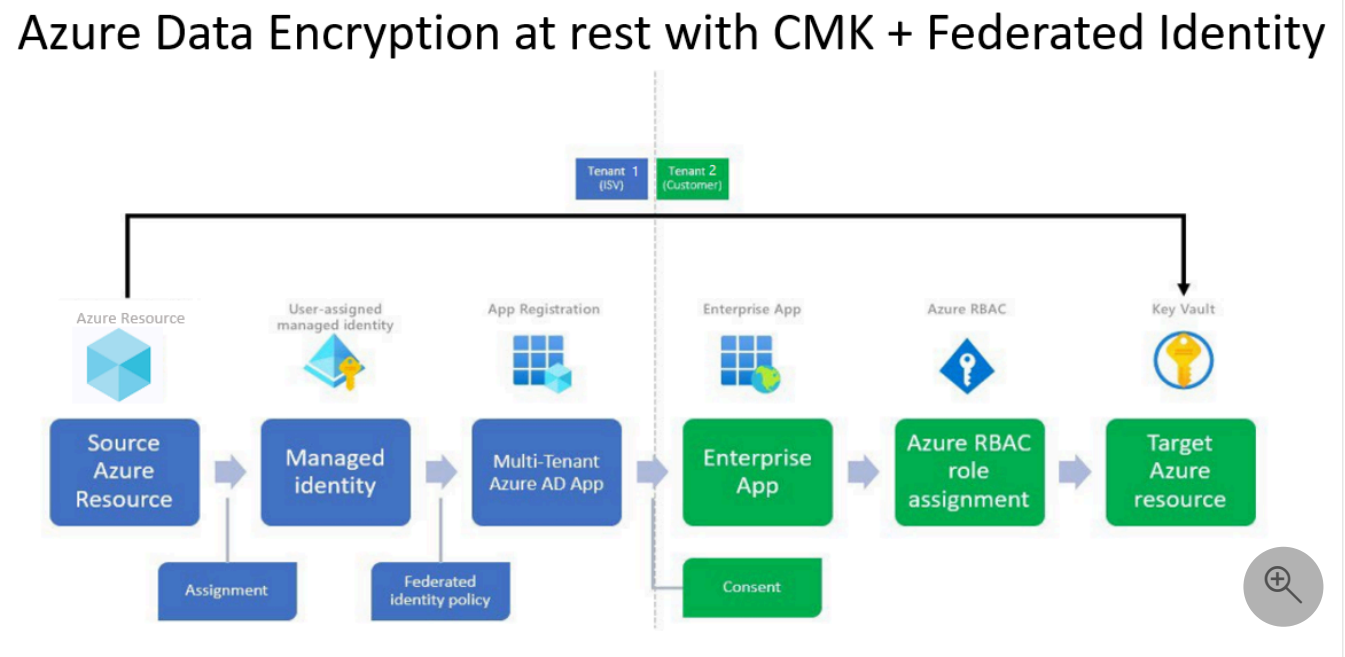
- Australia East
- Southeast Asia
- Canada Central
- North Europe
- France Central
- Germany West Central
- Korea Central
- Sweden Central
- UK South
- West US
- Central US

About cross-tenant customer-managed keys

Many service providers building Software as a Service (SaaS) offerings on Azure want to offer their customers the option to manage their own encryption keys. Customer-managed keys allow a service provider to encrypt the customer's data using an encryption key that is managed by the service provider's customer and that isn't accessible to the service provider. In Azure, the service provider's customer can use Azure Key Vault to manage their encryption keys in their own Microsoft Entra tenant and subscription.

Azure platform services and resources that are owned by the service provider and that reside in the service provider's tenant require access to the key from the customer's tenant to perform the encryption/decryption operations.

The image below shows a data encryption at rest with federated identity in a cross-tenant CMK workflow spanning a service provider and its customer.



In the example above, there are two Microsoft Entra tenants: an independent service provider's tenant (*Tenant 1*), and a customer's tenant (*Tenant 2*). *Tenant 1* hosts Azure platform services and *Tenant 2* hosts the customer's key vault.

A multitenant application registration is created by the service provider in *Tenant 1*. A federated identity credential is created on this application using a user-assigned managed identity. Then, the name and application ID of the app is shared with the customer.

A user with the appropriate permissions installs the service provider's application in the customer tenant, *Tenant 2*. A user then grants the service principal associated with the installed application access to the customer's key vault. The customer also stores the encryption key, or customer-managed key, in the key vault. The customer shares the key location (the URL of the key) with the service provider.

The service provider now has:

- An application ID for a multitenant application installed in the customer's tenant, which has been granted access to the customer-managed key.
- A managed identity configured as the credential on the multitenant application.
- The location of the key in the customer's key vault.

With these three parameters, the service provider provisions Azure resources in *Tenant 1* that can be encrypted with the customer-managed key in *Tenant 2*.

Let's divide the above end-to-end solution into three phases:

1. The service provider configures identities.
2. The customer grants the service provider's multitenant app access to an encryption key in Azure Key Vault.

3. The service provider encrypts data in an Azure resource using the CMK.

Operations in Phase 1 would be a one-time setup for most service provider applications. Operations in Phases 2 and 3 would repeat for each customer.

Phase 1 - The service provider configures a Microsoft Entra application

Expand table

Step	Description	Minimum role in Azure RBAC	Minimum role in Microsoft Entra RBAC
1.	Create a new multitenant Microsoft Entra application registration or start with an existing application registration. Note the application ID (client ID) of the application registration using Azure portal, Microsoft Graph API, Azure PowerShell, or Azure CLI	None	Application Developer
2.	Create a user-assigned managed identity (to be used as a Federated Identity Credential). Azure portal / Azure CLI / Azure PowerShell/ Azure Resource Manager Templates	Managed identity contributor	None
3.	Configure user-assigned managed identity as a *federated identity credential* on the application, so that it can impersonate the identity of the application. Graph API reference / Azure portal/ Azure CLI/ Azure PowerShell	None	Owner of the application
4.	Share the application name and application ID with the customer, so that they can install and authorize the application.	None	None

Considerations for service providers

- Azure Resource Manager (ARM) templates aren't recommended for creating Microsoft Entra applications.
- The same multitenant application can be used to access keys in any number of tenants, like *Tenant 2*, *Tenant 3*, *Tenant 4*, and so on. In each tenant, an independent instance of the application is created that has the same application ID but a different object ID. Each instance of this application is thus authorized independently. Consider how the application object used for this feature is used to partition your application across all customers.

    - Application can have a maximum of 20 federated identity credentials, which requires a service provider to share federated identities among its customers. For more information about federated identities design considerations and restrictions, see Configure an app to trust an external identity provider
- In rare scenarios, a service provider might use a single Application object per its customer, but that requires significant maintenance costs to manage applications at scale across all customers.
- In the service provider tenant, it isn't possible to automate the Publisher Verification.

Phase 2 - The customer authorizes access to the key vault

Expand table

Step	Description	Least privileged Azure RBAC roles	Least privileged Microsoft Entra roles
1.	• *Recommended*: Send the user to sign in to your app. If the user can sign in, then a service principal for your app exists in their tenant. • Use Microsoft Graph, Microsoft Graph PowerShell, Azure PowerShell, or Azure CLI to create the service principal. • Construct an admin-consent URL and grant tenant-wide consent to create the service principal using the application ID.	None	Users with permissions to install applications
2.	Create an Azure Key Vault and a key used as the customer-managed key.	A user must be assigned the Key Vault Contributor role to create the key vault A user must be assigned the Key Vault Crypto Officer role to add a key to the key vault	None
3.	Grant the consented application identity access to the Azure key vault by assigning the role Key Vault Crypto Service Encryption User	To assign the **Key Vault Crypto Service Encryption User** role to the application, you must have been assigned the User Access Administrator role.	None
4.	Copy the key vault URL and key name into the customer-managed	None	None

Step	Description	Least privileged Azure RBAC roles	Least privileged Microsoft Entra roles
	keys configuration of the SaaS offering.		

> **Note**
>
> To authorize access to the Managed HSM for encryption using CMK, see example for Storage Account here. For more information about managing keys with Managed HSM, see Manage a Managed HSM using the Azure CLI

Considerations for customers of service providers

- In the customer tenant, *Tenant 2*, an admin can set policies to block non-admin users from installing applications. These policies can prevent non-admin users from creating service principals. If such a policy is configured, then users with permissions to create service principals need to be involved.
- Access to Azure Key Vault can be authorized using Azure RBAC or access policies. When granting access to a key vault, make sure to use the active mechanism for your key vault.
- A Microsoft Entra application registration has an application ID (client ID). When the application is installed in your tenant, a service principal is created. The service principal shares the same application ID as the app registration, but generates its own object ID. When you authorize the application to have access to resources, you might need to use the service principal `Name` or `ObjectID` property.

Phase 3 - The service provider encrypts data in an Azure resource using the customer-managed key

After phases 1 and 2 are complete, the service provider can configure encryption on the Azure resource with the key and key vault in the customer's tenant and the Azure resource in the ISV's tenant. The service provider can configure cross-tenant customer-managed keys with the client tools supported by that Azure resource, with an ARM template, or with the REST API.

Configure cross-tenant customer-managed keys

This section describes how to configure a cross-tenant customer-managed key (CMK) and encrypt customer data. You learn how to encrypt customer data in a resource in *Tenant1* using

a CMK stored in a key vault in *Tenant2*. You can use the Azure portal, Azure PowerShell, or Azure CLI.

Portal

Sign in to the Azure portal and follow these steps.

The service provider configures identities

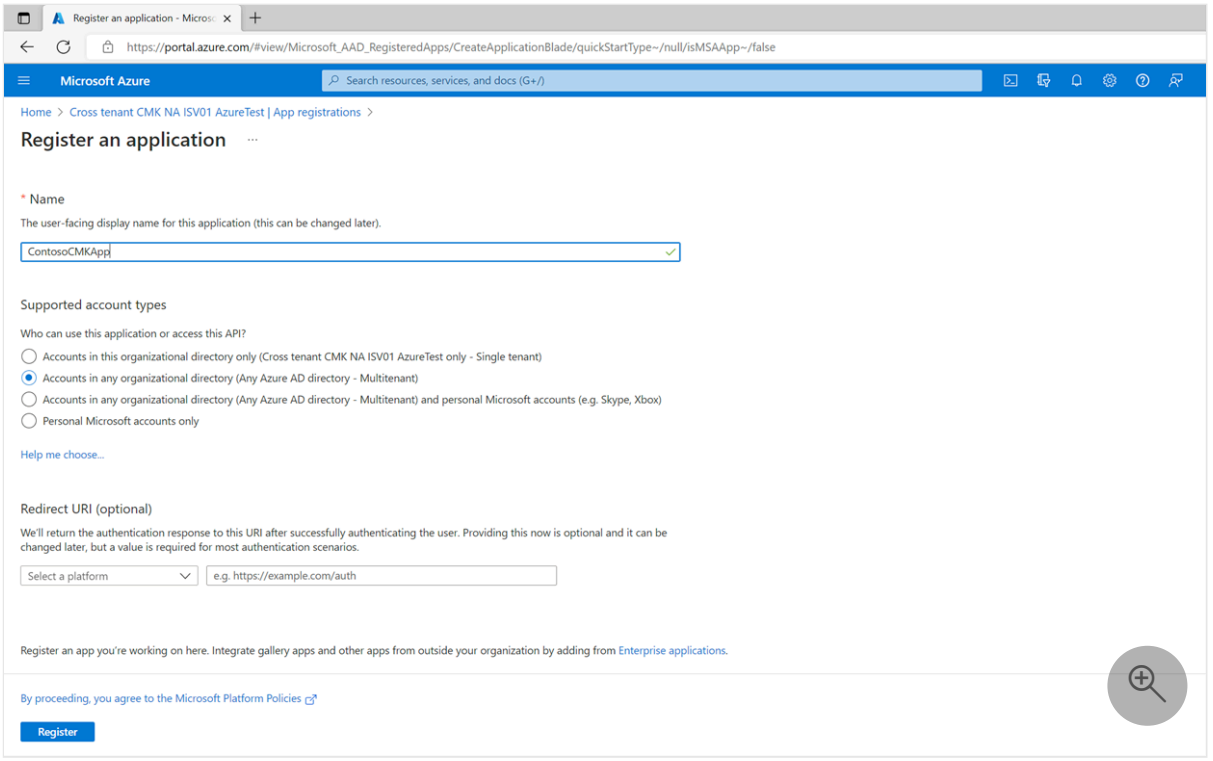
The following steps are performed by the service provider in the service provider's tenant *Tenant1*.

The service provider creates a new multi-tenant app registration

You can either create a new multi-tenant Microsoft Entra application registration or start with an existing multi-tenant application registration. If starting with an existing application registration, note the application ID (client ID) of the application.

To create a new registration:

1. Search for **Microsoft Entra ID** in the search box. Locate and select the **Microsoft Entra ID** extension.
2. Select **Manage** > **App registrations** from the left pane.
3. Select **+ New registration**.
4. Provide the name for the application registration and select *Account in any organizational directory (Any Microsoft Entra directory – Multitenant)*.
5. Select **Register**.
6. Note the **ApplicationId/ClientId** of the application.

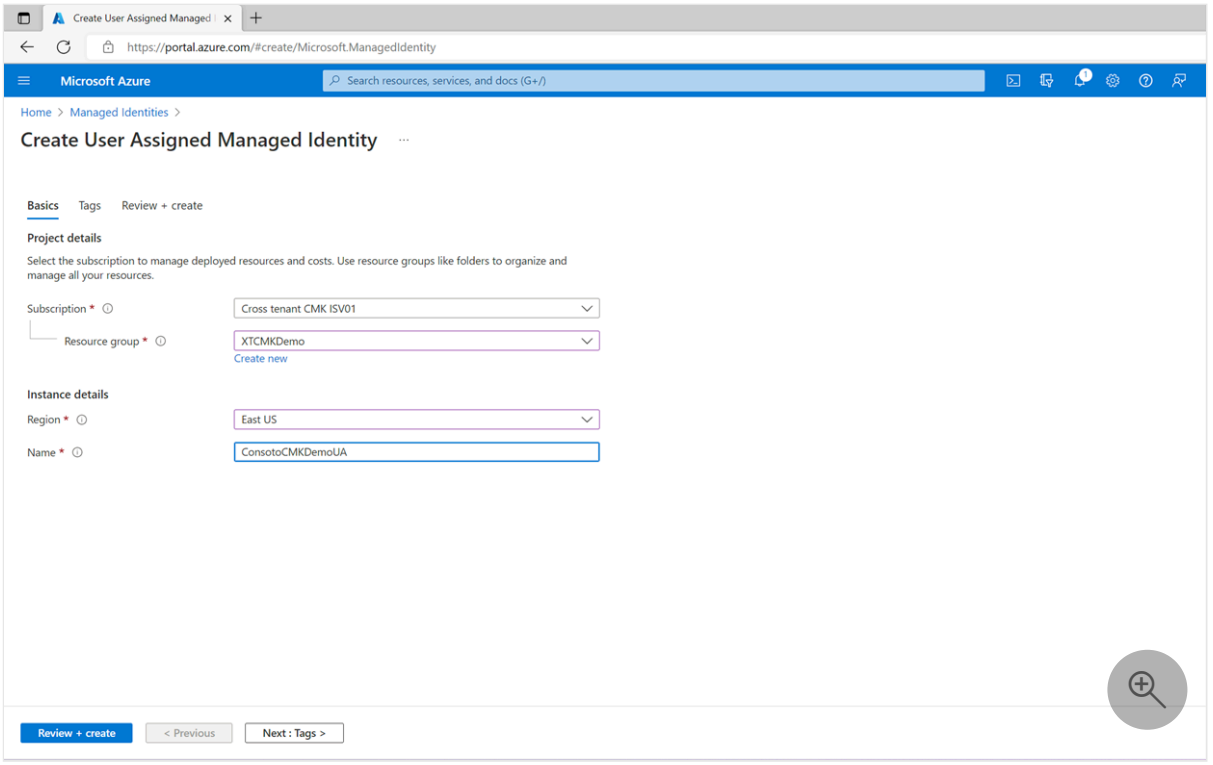


The service provider creates a user-assigned managed identity

Create a user-assigned managed identity to be used as a federated identity credential.

1. Search for **Managed Identities** in the search box. Locate and select the **Managed Identities** extension.
2. Select **+ Create**.
3. Provide the resource group, region, and name for the managed identity.
4. Select **Review + create**.
5. On successful deployment, note the **Azure ResourceId** of the user-assigned managed identity, which is available under **Properties**. For example:

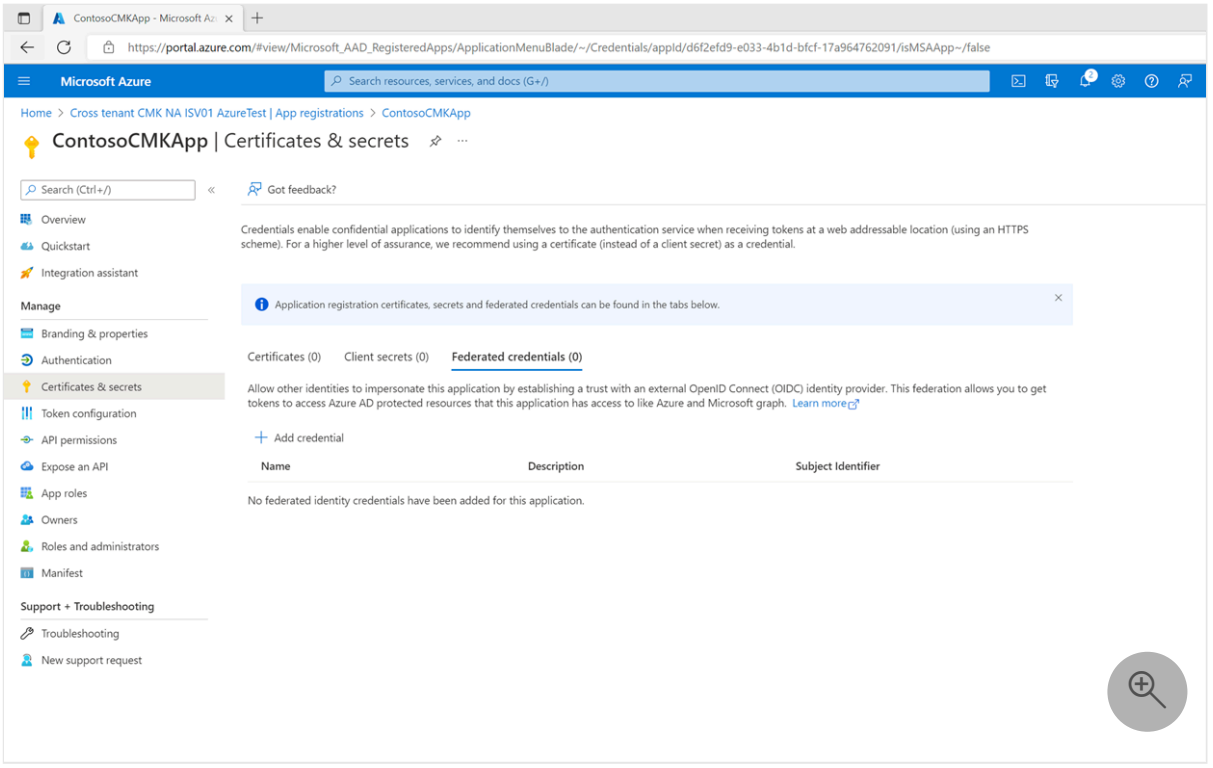
   `/subscriptions/tttttttt-0000-tttt-0000-tttt0000tttt/resourcegroups/XTCMKDemo/providers/Microsoft.ManagedIdentity/userAssignedIdentities/ConsotoCMKDemoUA`



The service provider configures the user-assigned managed identity as a federated credential on the application

Configure a user-assigned managed identity as a federated identity credential on the application, so that it can impersonate the identity of the application.

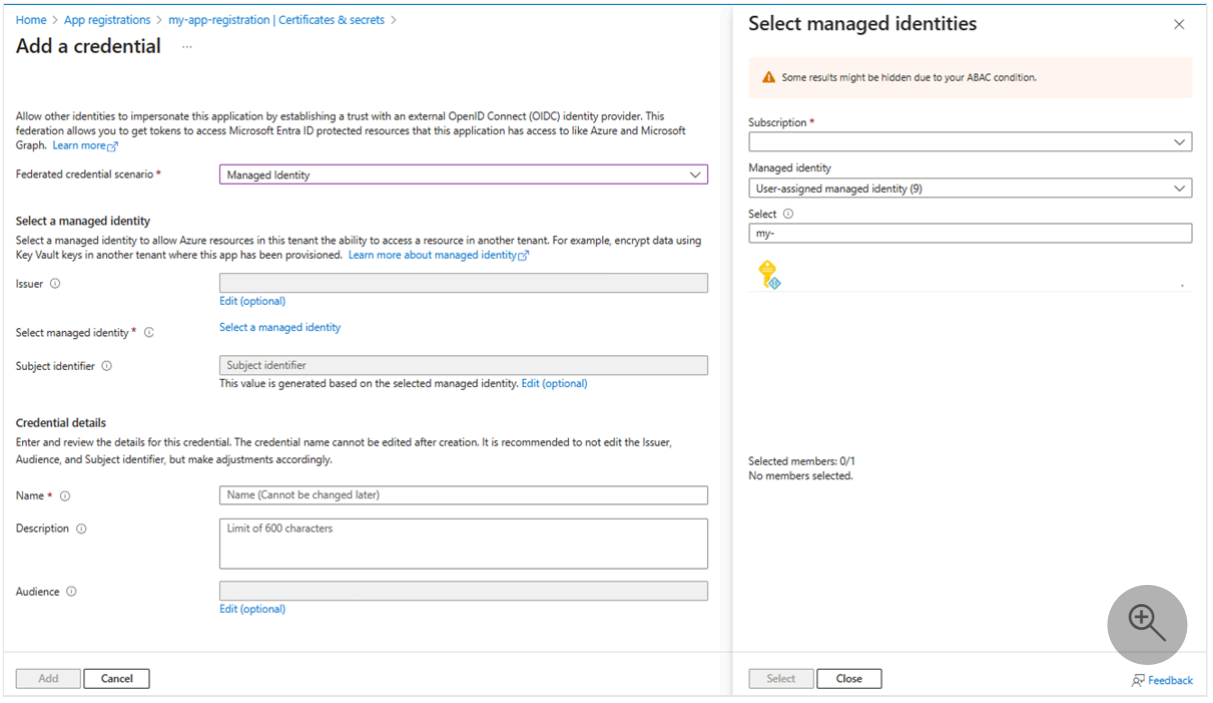
1. Navigate to **Microsoft Entra ID > App registrations > your application**.
2. Select **Certificates & secrets**.
3. Select **Federated credentials**.



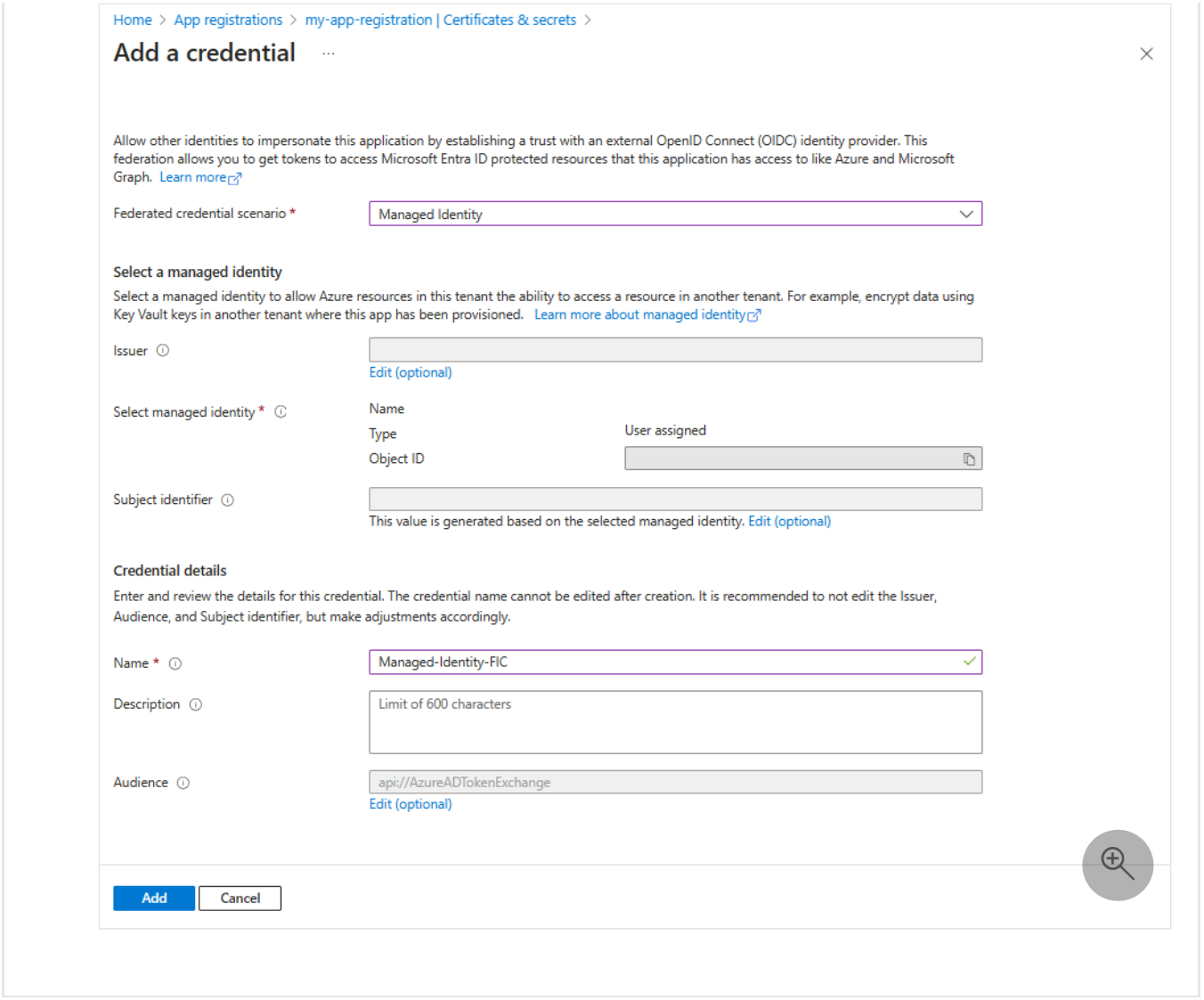
4. Select **+ Add credential**.

5. Under **Federated credential scenario**, select **Managed Identity**.

6. Click **Select a managed identity**. From the pane, select the subscription. Under **Managed identity**, select **User-assigned managed identity**. In the **Select** box, search for the managed identity you created earlier, then click **Select** at the bottom of the pane.



7. Under **Credential details**, provide a name and optional description for the credential and select **Add**.



The service provider shares the application ID with the customer

Find the application ID (client ID) of the multi-tenant application and share it with the customer.

The customer grants the service provider's app access to the key in the key vault

The following steps are performed by the customer in the customer's tenant *Tenant2*. The customer can use the Azure portal, Azure PowerShell, or Azure CLI.

The user executing the steps must be an administrator with a privileged role such as Application Administrator, Cloud Application Administrator, or Global Administrator.

Portal

Sign in to the Azure portal and follow these steps.

The customer installs the service provider application in the customer tenant

To install the service provider's registered application in the customer's tenant, you create a service principal with the application ID from the registered app. You can create the service principal in either of the following ways:

- Use Microsoft Graph, Microsoft Graph PowerShell, Azure PowerShell, or Azure CLI to manually create the service principal.
- Construct an admin-consent URL and grant tenant-wide consent to create the service principal. You'll need to provide them with your AppId.

The customer creates a key vault

To create the key vault, the user's account must be assigned the **Key Vault Contributor** role or another role that permits creation of a key vault.

1. From the Azure portal menu, or from the Home page, select **+ Create a resource**. In the Search box, enter **Key vaults**. From the results list, select **Key vaults**. On the **Key vaults** page, select **Create**.

2. On the **Basics** tab, choose a subscription. Under **Resource group**, select **Create new** and enter a resource group name.

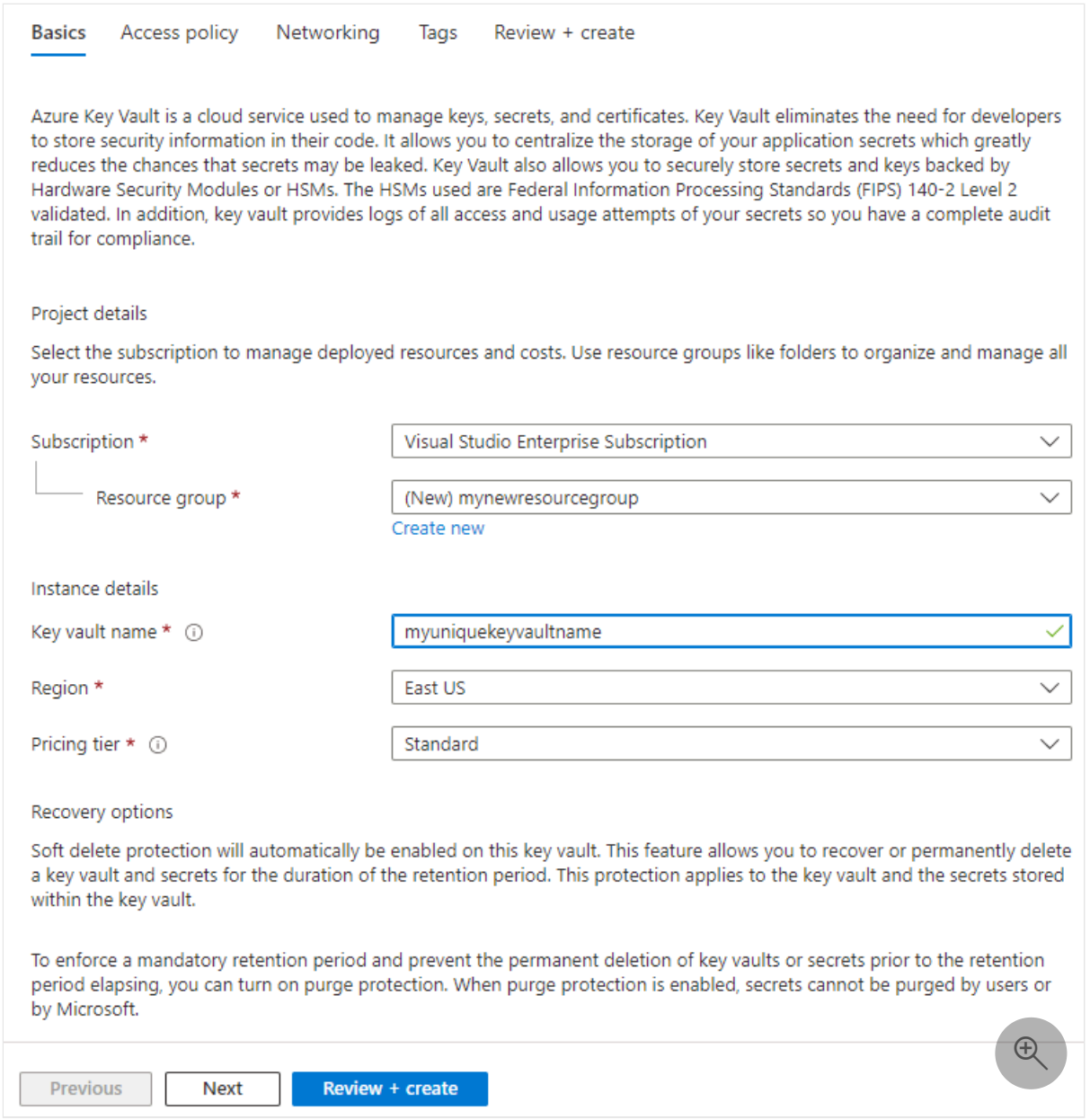
3. Enter a unique name for the key vault.

4. Select a region and pricing tier.

5. Enable purge protection for the new key vault.

6. On the **Access policy** tab, select **Azure role-based access control** for **Permission model**.

7. Select **Review + create** and then **Create**.



Take note of the key vault name and URI Applications that access your key vault must use this URI.

For more information, see Quickstart - Create an Azure Key Vault with the Azure portal.

The customer assigns Key Vault Crypto Officer role to a user account

This step ensures that you can create encryption keys.

1. Navigate to your key vault and select **Access Control (IAM)** from the left pane.
2. Under **Grant access to this resource**, select **Add role assignment**.
3. Search for and select **Key Vault Crypto Officer**.
4. Under **Members**, select **User, group, or service principal**.
5. Select **Members** and search for your user account.
6. Select **Review + Assign**.

The customer creates an encryption key

To create the encryption key, the user's account must be assigned the **Key Vault Crypto Officer** role or another role that permits creation of a key.

1. On the Key Vault properties page, select **Keys**.
2. Select **Generate/Import**.
3. On the **Create a key** screen, specify a name for the key. Leave the other values to their defaults.
4. Select **Create**.
5. Copy the key URI.

The customer grants the service provider application access to the key vault

Assign the Azure RBAC role **Key Vault Crypto Service Encryption User** to the service provider's registered application so that it can access the key vault.

1. Navigate to your key vault and select **Access Control (IAM)** from the left pane.
2. Under **Grant access to this resource**, select **Add role assignment**.
3. Search for and select **Key Vault Crypto Service Encryption User**.
4. Under **Members**, select **User, group, or service principal**.
5. Select **Members** and search for the application name of the application you installed from the service provider.
6. Select **Review + Assign**.

Now you can configure customer-managed keys with the key vault URI and key.

Create a disk encryption set

Now that you've created your Azure Key Vault and performed the required Microsoft Entra configurations, deploy a disk encryption set configured to work across tenants and associate it with a key in the key vault. You can do this using the Azure portal, Azure PowerShell, or Azure CLI. You can also use an ARM template or REST API.

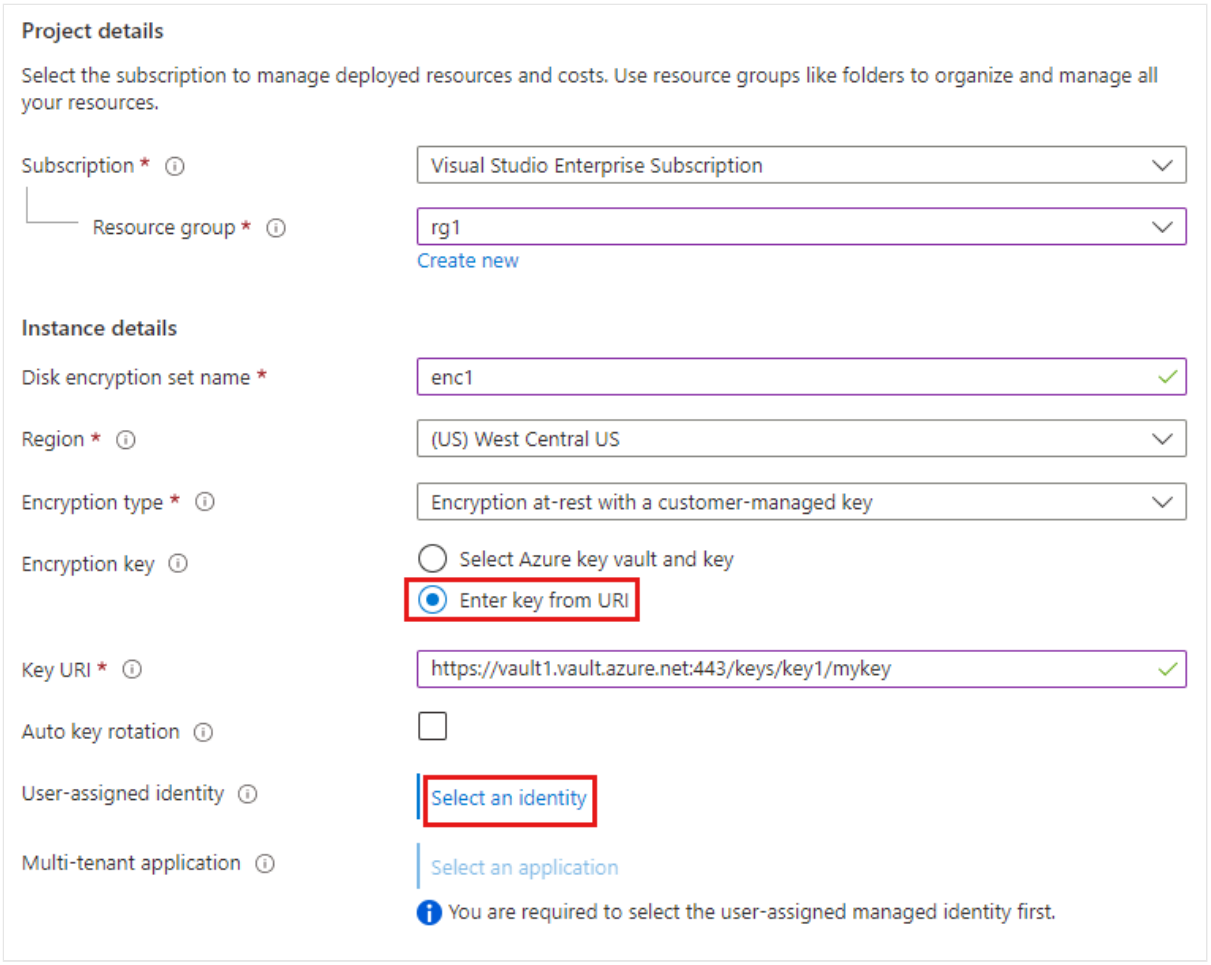
Portal

To use the Azure portal, sign in to the portal and follow these steps.

1. Select **+ Create a resource**, search for **Disk encryption set**, and select **Create** > **Disk encryption set**.

2. Under **Project details**, select the subscription and resource group in which to create the disk encryption set.

3. Under **Instance details**, provide a name for the disk encryption set.



4. Select the **Region** in which to create the disk encryption set.

5. For **Encryption type**, select **Encryption at-rest with a customer-managed key**.

6. Under **Encryption key**, select the **Enter key from URI** radio button, and then enter the Key URI of the key created in the customer's tenant.

7. Under **User-assigned identity**, select **Select an identity**.

8. Select the user-assigned managed identity that you created previously in the ISV's tenant, and then select **Add**.

9. Under **Multi-tenant application**, select **Select an application**.

10. Select the multi-tenant registered application that you created previously in the ISV's tenant, and click **Select**.

11. Select **Review + create**.

Use an ARM template

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "desname": {
      "defaultValue": "<Enter ISV disk encryption set name>",
      "type": "String"
    },
    "region": {
      "defaultValue": "WestCentralUS",
      "type": "String"
    },
    "userassignedmicmk": {
      "defaultValue": "/subscriptions/<Enter ISV Subscription
Id>/resourceGroups/<Enter ISV resource group
name>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<Enter ISV User
Assigned Identity Name>",
      "type": "String"
    },
    "cmkfederatedclientId": {
      "defaultValue": "<Enter ISV Multi-Tenant App Id>",
      "type": "String"
    },
    "keyVaultURL": {
      "defaultValue": "<Enter Client Key URL>",
      "type": "String"
    },
    "encryptionType": {
      "defaultValue": "EncryptionAtRestWithCustomerKey",
      "type": "String"
    }
  },
  "variables": {},
  "resources": [
    {
      "type": "Microsoft.Compute/diskEncryptionSets",
      "apiVersion": "2021-12-01",
      "name": "[parameters('desname')]",
      "location": "[parameters('region')]",
      "identity": {
        "type": "UserAssigned",
        "userAssignedIdentities": {
          "[parameters('userassignedmicmk')]": {}
        }
      },
      "properties": {
        "activeKey": {
          "keyUrl": "[parameters('keyVaultURL')]"
        },
```

```
        "federatedClientId": "[parameters('cmkfederatedclientId')]",
        "encryptionType": "[parameters('encryptionType')]"
      }
    }
  ]
}
```

Use REST API

Use bearer token as authorization header and application/JSON as content type in BODY. (Network tab, filter to management.azure while performing any ARM request on portal.)

rest

```
PUT https://management.azure.com/subscriptions/<Enter ISV Subscription
Id>/resourceGroups/<Enter ISV Resource Group
Name>/providers/Microsoft.Compute/diskEncryptionSets/<Enter ISV Disk Encryption
Set Name>?api-version=2021-12-01
Authorization: Bearer ...
Content-Type: application/json

{
  "name": "<Enter ISV disk encryption set name>",
  "id": "/subscriptions/<Enter ISV Subscription Id>/resourceGroups/<Enter ISV
resource group name>/providers/Microsoft.Compute/diskEncryptionSets/<Enter ISV
disk encryption set name>/",
  "type": "Microsoft.Compute/diskEncryptionSets",
  "location": "westcentralus",
  "identity": {
    "type": "UserAssigned",
    "userAssignedIdentities": {
"/subscriptions/<Enter ISV Subscription Id>/resourceGroups/<Enter ISV resource
group name>/providers/Microsoft.ManagedIdentity/userAssignedIdentities/<Enter ISV
User Assigned Identity Name>
": {}
    }
  },
  "properties": {
    "activeKey": {
      "keyUrl": "<Enter Client Key URL>"
    },
    "encryptionType": "EncryptionAtRestWithCustomerKey",
    "federatedClientId": "<Enter ISV Multi-Tenant App Id>"
  }
}
```

Next steps

See also:

- Encrypt disks using customer-managed keys in Azure DevTest Labs
- Use the Azure portal to enable server-side encryption with customer-managed keys for managed disks

Use the Azure portal to enable end-to-end encryption using encryption at host

Applies to: ✔ Linux VMs ✔ Windows VMs

When you enable encryption at host, data stored on the VM host is encrypted at rest and flows encrypted to the Storage service. For conceptual information on encryption at host, and other managed disk encryption types, see: Encryption at host - End-to-end encryption for your VM data.

Temporary disks and ephemeral OS disks are encrypted at rest with platform-managed keys when you enable end-to-end encryption. The OS and data disk caches are encrypted at rest with either customer-managed or platform-managed keys, depending on the selected disk encryption type. For example, if a disk is encrypted with customer-managed keys, then the cache for the disk is encrypted with customer-managed keys, and if a disk is encrypted with platform-managed keys then the cache for the disk is encrypted with platform-managed keys.

Restrictions

- Can't be enabled on virtual machines (VMs) or virtual machine scale sets that currently or ever had Azure Disk Encryption enabled.
- Azure Disk Encryption can't be enabled on disks that have encryption at host enabled.
- The encryption can be enabled on existing virtual machine scale sets. However, only new VMs created after enabling the encryption are automatically encrypted.
- Existing VMs must be deallocated and reallocated in order to be encrypted.

The following restrictions only apply to Ultra Disks and Premium SSD v2:

- Disks using a 512e sector size need to have been created after 5/13/2023.
  - If your disk was created before this date, snapshot your disk, and create a new disk using the snapshot.

Supported VM sizes

Legacy VM Sizes aren't supported. You can find the list of supported VM sizes by either using the Azure PowerShell module or Azure CLI.

Prerequisites

You must enable the feature for your subscription before you can use encryption at host for either your VM or Virtual Machine Scale Set. Use the following steps to enable the feature for your subscription:

1. **Azure portal**: Select the Cloud Shell icon on the Azure portal:



2. Execute the following command to set context to current subscription

Azure PowerShell

PowerShell

```
Set-AzContext -SubscriptionId "<yourSubIDHere>"
```

3. Execute the following command to register the feature for your subscription

Azure PowerShell

PowerShell

```
Register-AzProviderFeature -FeatureName "EncryptionAtHost" -
ProviderNamespace "Microsoft.Compute"
```

4. Confirm that the registration state is **Registered** (registration might take a few minutes) using the following command before trying out the feature.

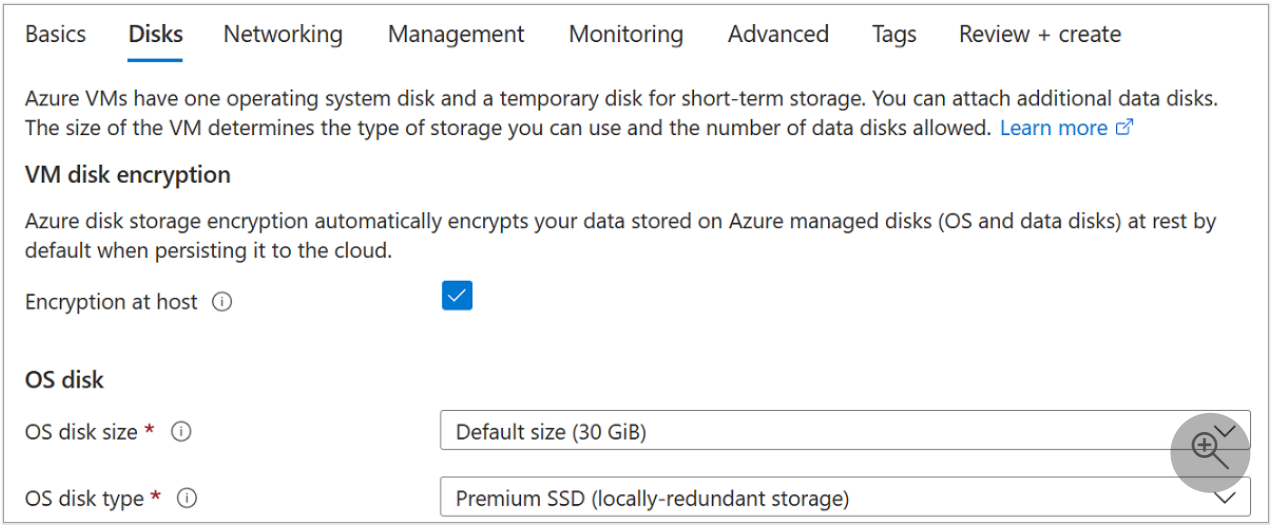
Azure PowerShell

PowerShell

```
Get-AzProviderFeature -FeatureName "EncryptionAtHost" -ProviderNamespace
"Microsoft.Compute"
```

Deploy a VM with platform-managed keys

1. Sign in to the Azure portal.
2. Search for **Virtual Machines** and select **+ Create** to create a VM.
3. Select an appropriate region and a supported VM size.
4. Fill in the other values on the **Basic** pane as you like, and then proceed to the **Disks** pane.
5. On the **Disks** pane, select **Encryption at host**.
6. Make the remaining selections as you like.



7. For the rest of the VM deployment process, make selections that fit your environment, and complete the deployment.

You've now deployed a VM with encryption at host enabled, and the cache for the disk is encrypted using platform-managed keys.

Deploy a VM with customer-managed keys

Alternatively, you can use customer-managed keys to encrypt your disk caches.

Create an Azure Key Vault and disk encryption set

Once the feature is enabled, you need to set up an Azure Key Vault and a disk encryption set, if you haven't already.

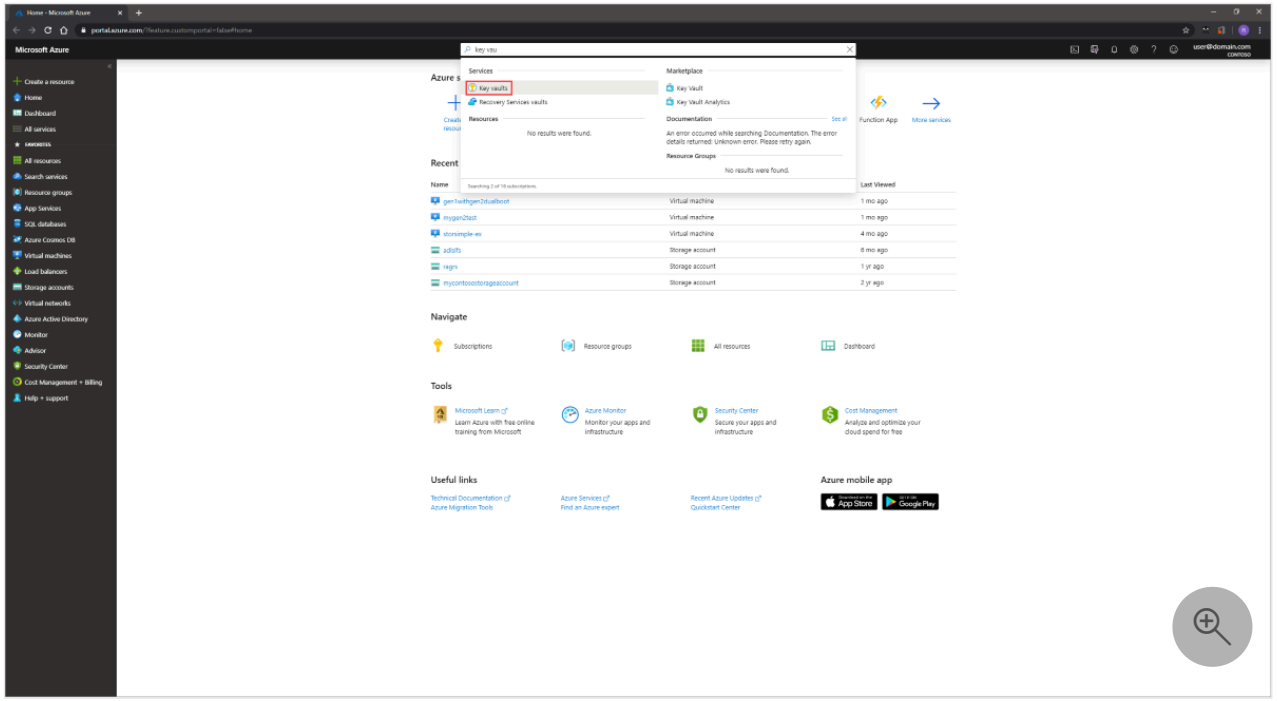
Setting up customer-managed keys for your disks requires you to create resources in a particular order, if you're doing it for the first time. First, you'll need to create and set up an

Azure Key Vault.

Set up your Azure Key Vault

1. Sign in to the Azure portal.

2. Search for and select **Key Vaults**.



> **Important**
>
> Your disk encryption set, VM, disks, and snapshots must all be in the same region and subscription for deployment to succeed.
>
> - Azure Key Vaults may be used from a different subscription, but they must be in the same region as your disk encryption set.
> - Azure Key Vaults in different Microsoft Entra tenants are supported for managed disks. For prerequisites, see Encrypt managed disks with cross-tenant customer-managed keys.

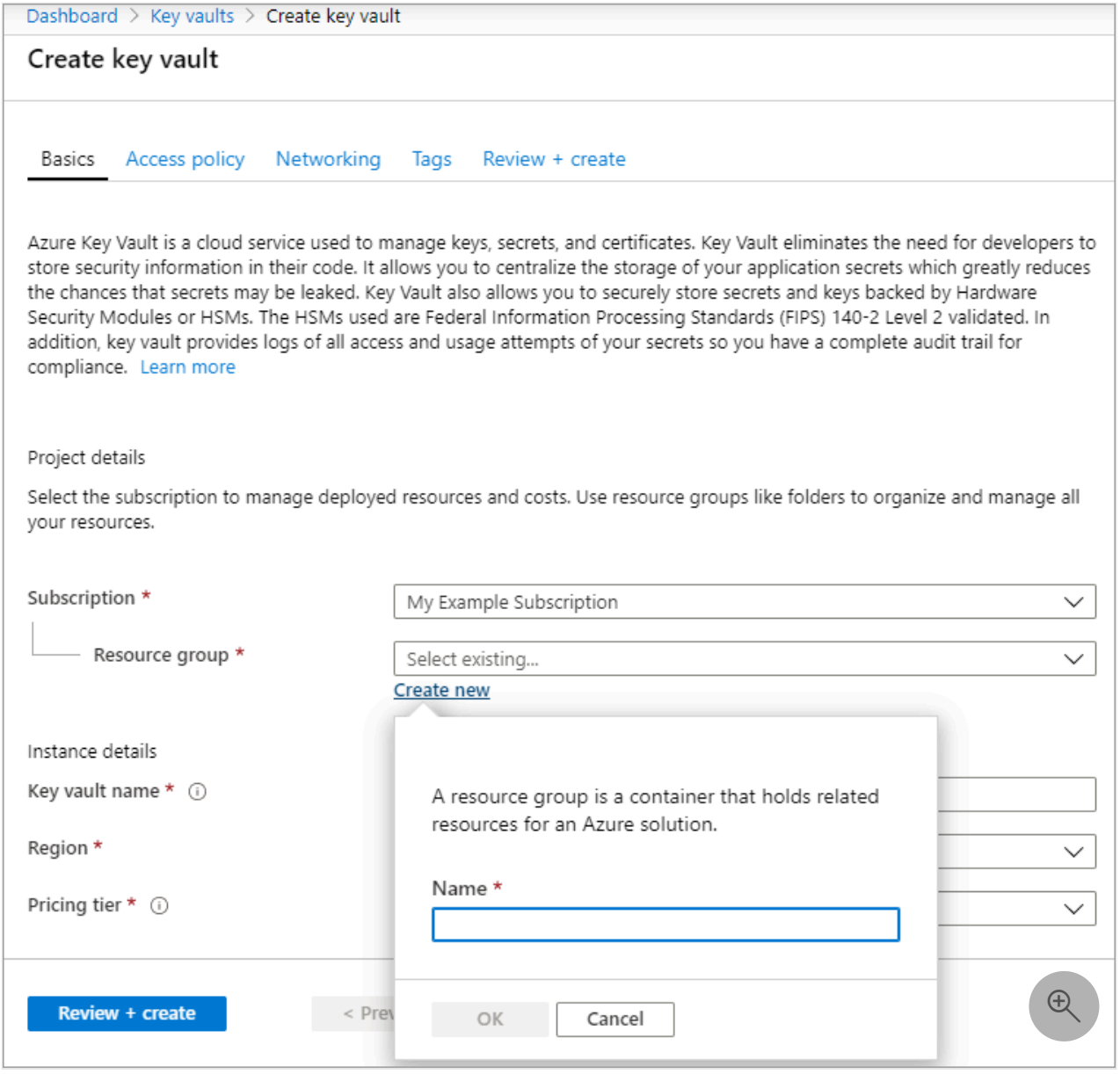
3. Select **+Create** to create a new Key Vault.

4. Create a new resource group.

5. Enter a key vault name, select a region, and select a pricing tier.

> **Note**
>
> When creating the Key Vault instance, you must enable soft delete and purge protection. Soft delete ensures that the Key Vault holds a deleted key for a given retention period (90 day default). Purge protection ensures that a deleted key cannot be permanently deleted until the retention period lapses. These settings protect you from losing data due to accidental deletion. These settings are mandatory when using a Key Vault for encrypting managed disks.

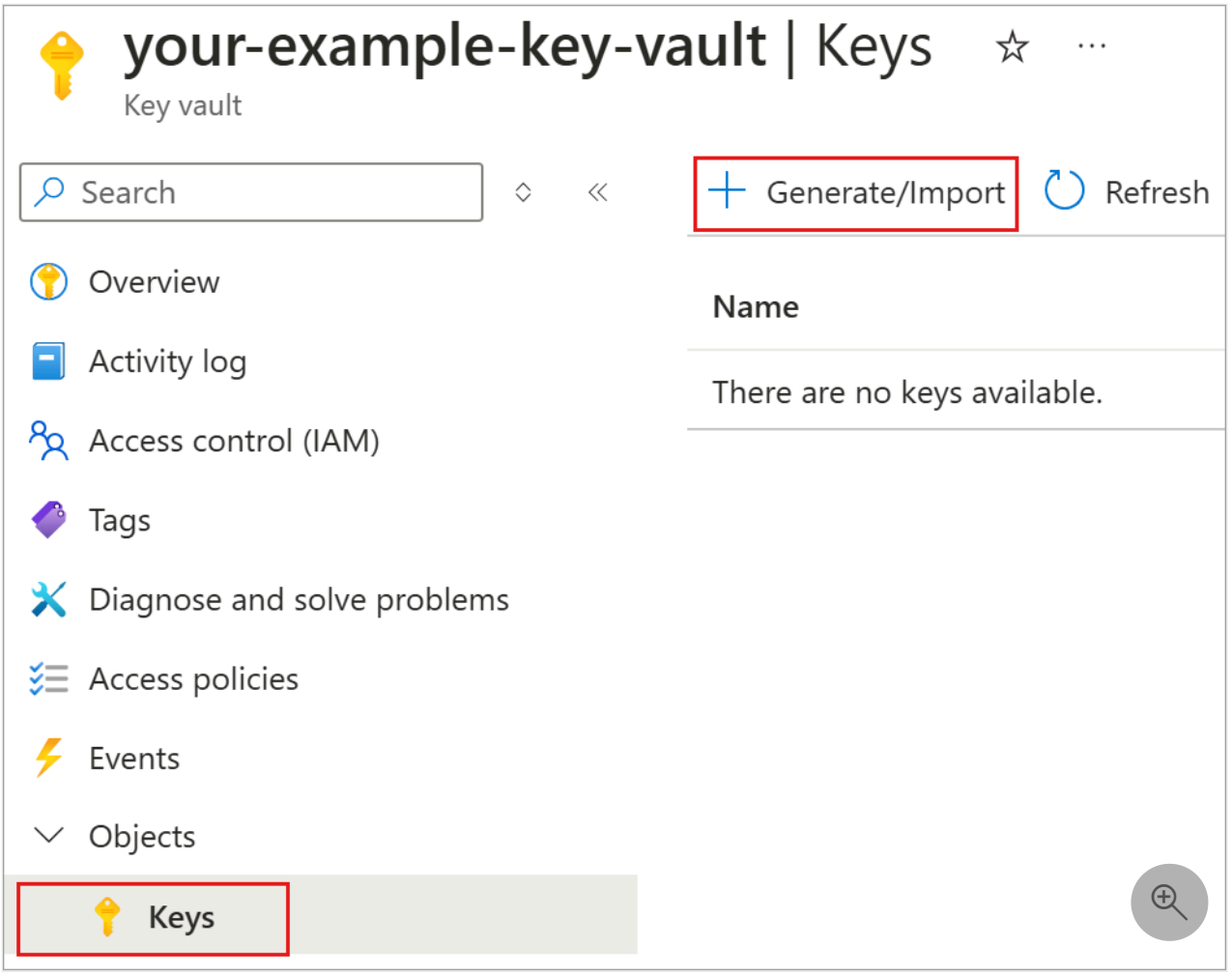
6. Select **Review + Create**, verify your choices, then select **Create**.



7. Once your key vault finishes deploying, select it.

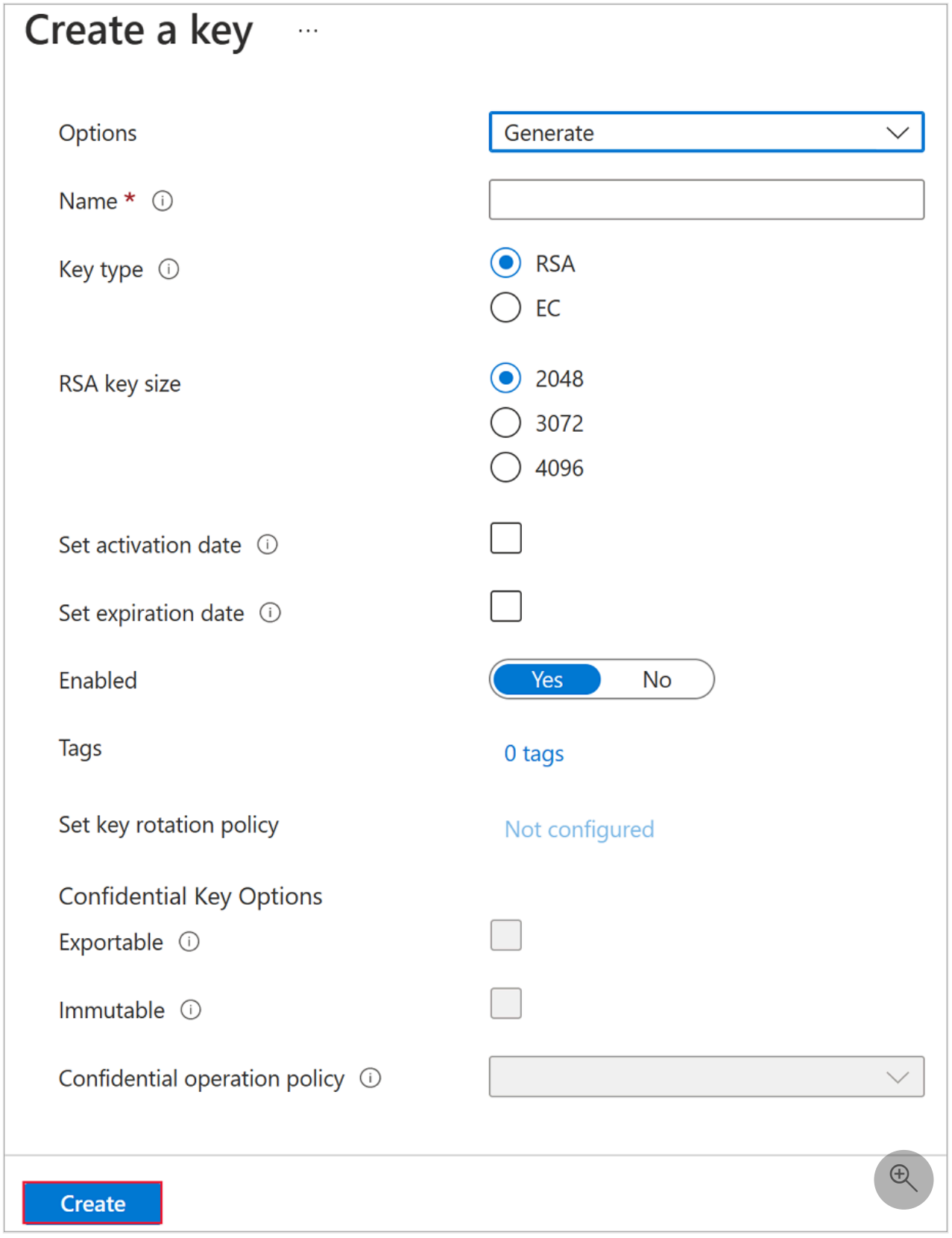
8. Select **Keys** under **Objects**.

9. Select **Generate/Import**.



10. Leave both **Key Type** set to **RSA** and **RSA Key Size** set to **2048**.

11. Fill in the remaining selections as you like and then select **Create**.



Add an Azure RBAC role

Now that you've created the Azure key vault and a key, you must add an Azure RBAC role, so you can use your Azure key vault with your disk encryption set.

1. Select **Access control (IAM)** and add a role.
2. Add either the **Key Vault Administrator**, **Owner**, or **Contributor** roles.

Set up your disk encryption set

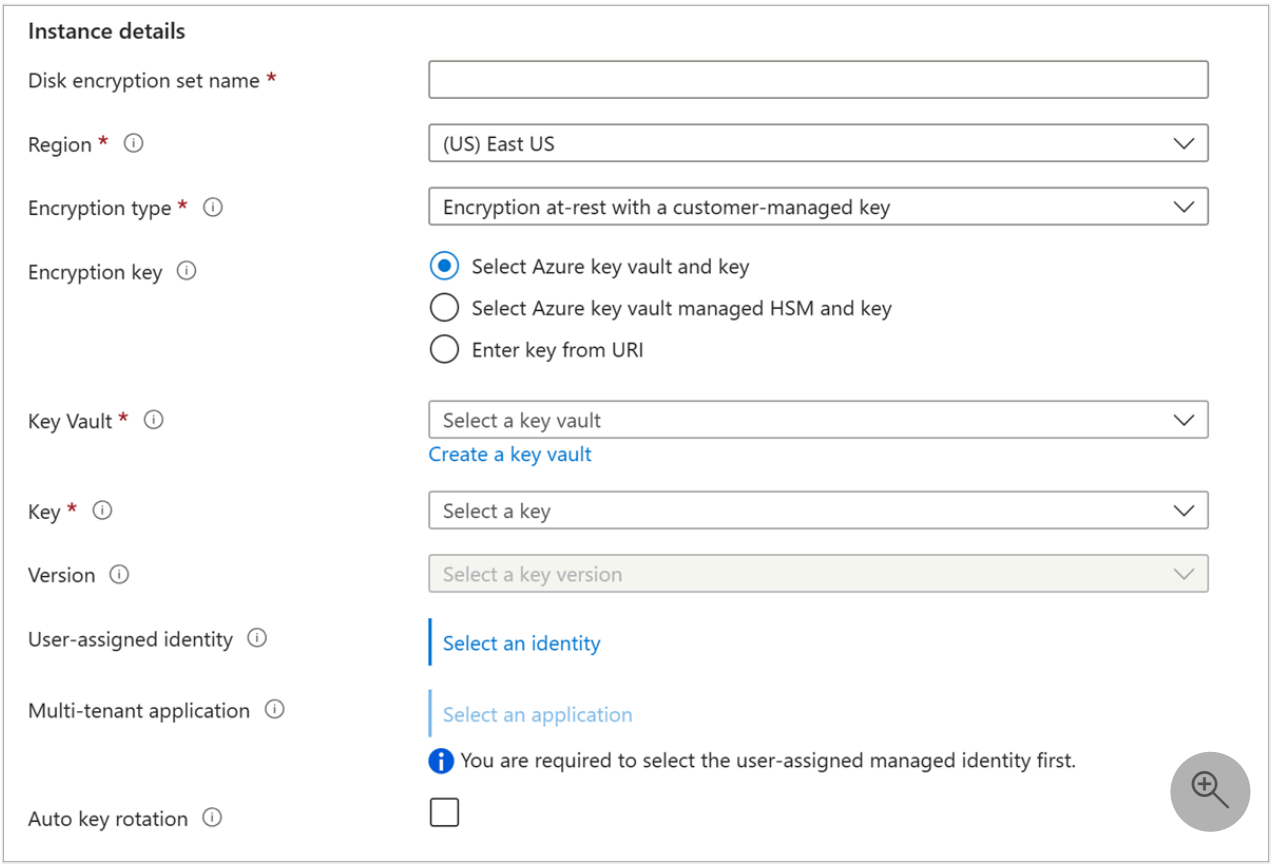
1. Search for **Disk Encryption Sets** and select it.
2. On the **Disk Encryption Sets** pane, select **+Create**.
3. Select your resource group, name your encryption set, and select the same region as your key vault.
4. For **Encryption type**, select **Encryption at-rest with a customer-managed key**.

   > ⓘ **Note**
   >
   > Once you create a disk encryption set with a particular encryption type, it cannot be changed. If you want to use a different encryption type, you must create a new disk encryption set.

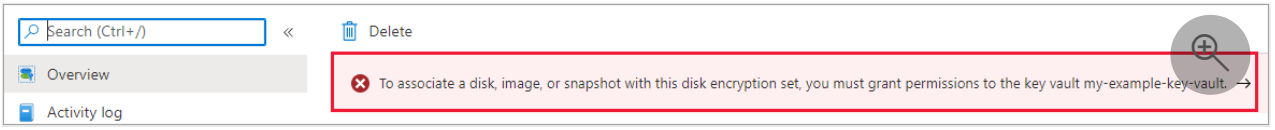
5. Make sure **Select Azure key vault and key** is selected.
6. Select the key vault and key you created previously, and the version.

   > ⓘ **Note**
   >
   > For managed disks with cross-tenant customer-managed keys, the selected key can be in a Key Vault in a different Microsoft Entra tenant.

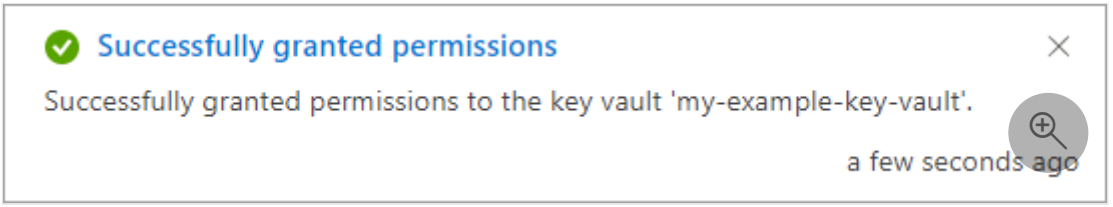
7. If you want to enable automatic rotation of customer managed keys, select **Auto key rotation**.
8. Select **Review + Create** and then **Create**.



9. Navigate to the disk encryption set once it's deployed, and select the displayed alert.



10. This will grant your key vault permissions to the disk encryption set.



Deploy a VM

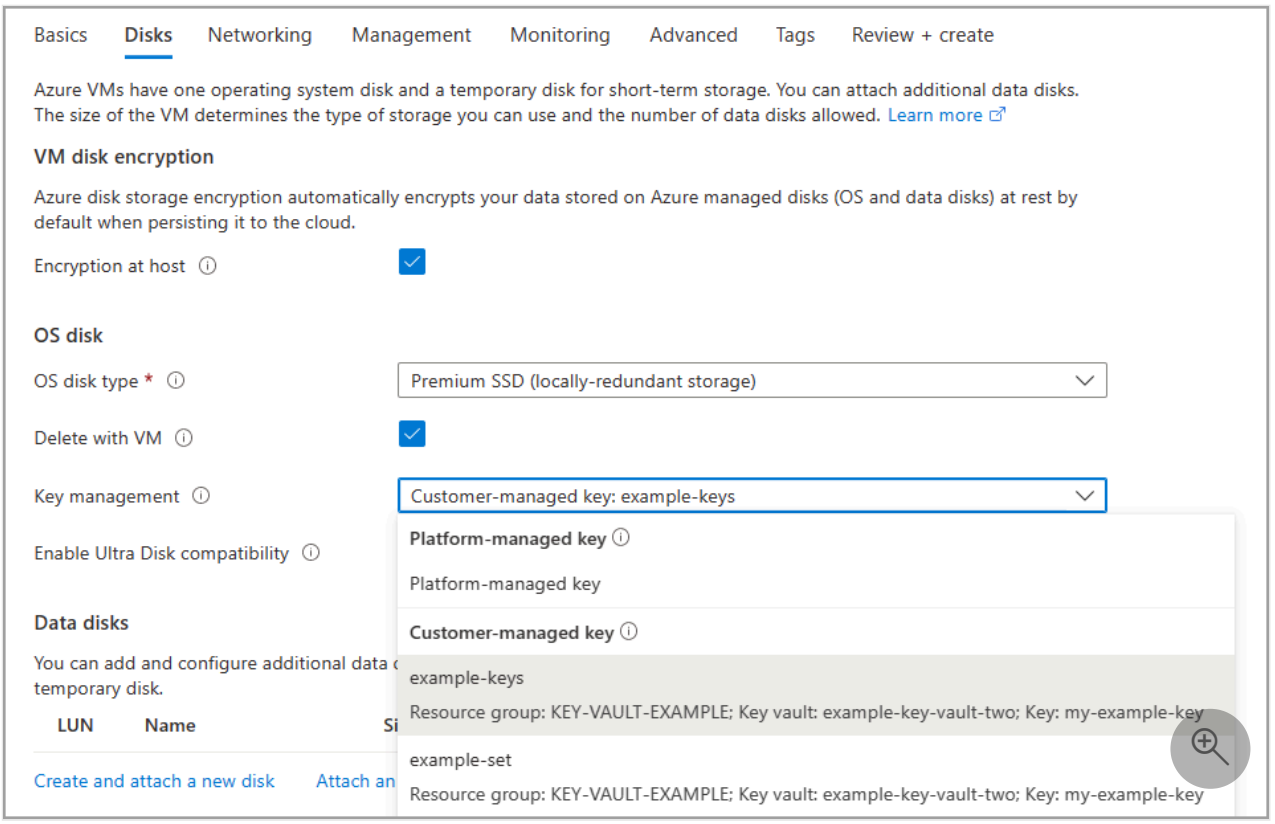
Now that you have setup an Azure Key Vault and disk encryption set, you can deploy a VM and it uses encryption at host.

1. Sign in to the Azure portal.
2. Search for **Virtual Machines** and select **+ Add** to create a VM.
3. Create a new virtual machine, select an appropriate region and a supported VM size.
4. Fill in the other values on the **Basic** pane as you like, then proceed to the **Disks** pane.

5. On the **Disks** pane, select **Encryption at host**.

6. Select **Key management** and select one of your customer-managed keys.

7. Make the remaining selections as you like.



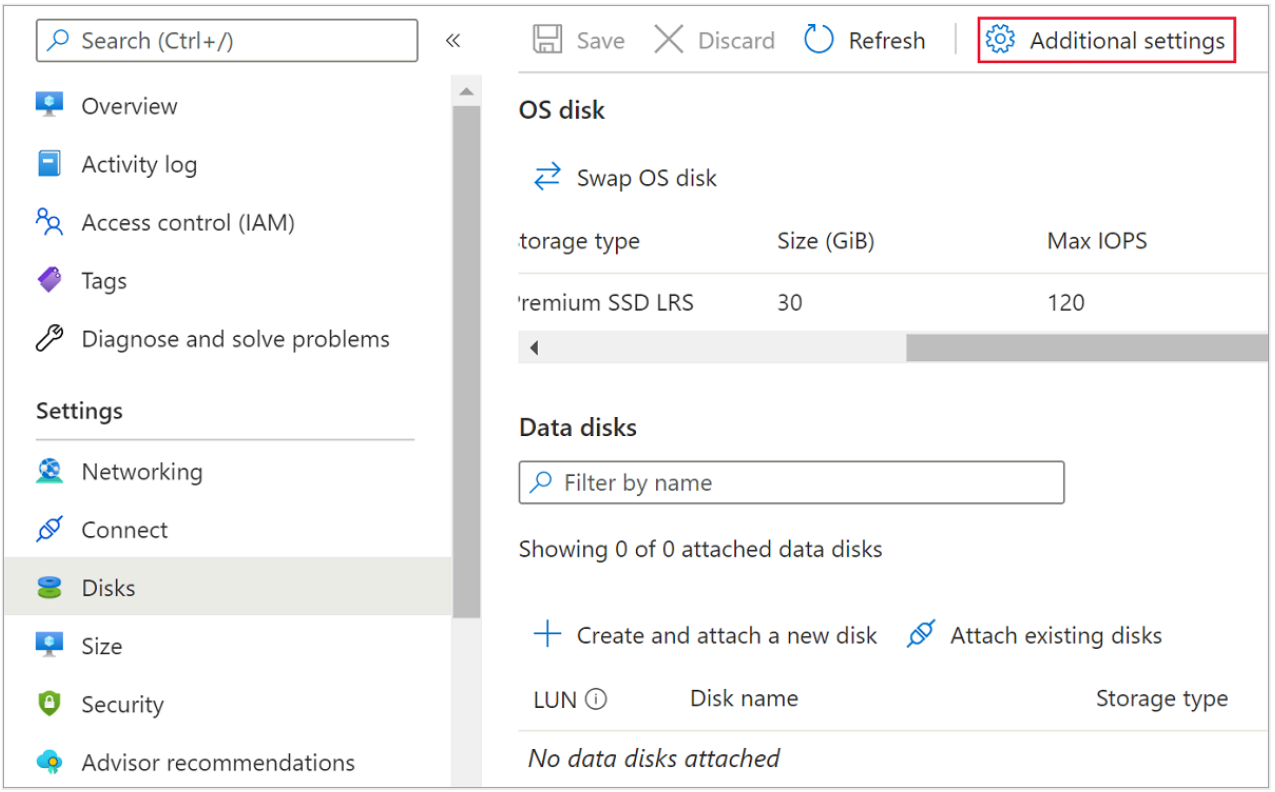
8. For the rest of the VM deployment process, make selections that fit your environment, and complete the deployment.

You've now deployed a VM with encryption at host enabled using customer-managed keys.

Disable host based encryption

Deallocate your VM first, encryption at host can't be disabled unless your VM is deallocated.

1. On your VM, select **Disks** and then select **Additional settings**.



2. Select **No** for **Encryption at host** then select **Save**.

Next steps

Azure Resource Manager template samples

Last updated on 04/02/2026

Use the Azure PowerShell module to enable end-to-end encryption using encryption at host

Article • 02/25/2025

Applies to: ✔ Windows VMs

When you enable encryption at host, data stored on the VM host is encrypted at rest and flows encrypted to the Storage service. For conceptual information on encryption at host, and other managed disk encryption types, see Encryption at host - End-to-end encryption for your VM data.

Restrictions

- Supported for 4k sector size Ultra Disks and Premium SSD v2.
- Only supported on 512e sector size Ultra Disks and Premium SSD v2 if they were created after 5/13/2023.
  - For disks created before this date, snapshot your disk and create a new disk using the snapshot.
- Can't be enabled on virtual machines (VMs) or virtual machine scale sets that currently or ever had Azure Disk Encryption enabled.
- Azure Disk Encryption can't be enabled on disks that have encryption at host enabled.
- The encryption can be enabled on existing virtual machine scale sets. However, only new VMs created after enabling the encryption are automatically encrypted.
- Existing VMs must be deallocated and reallocated in order to be encrypted.

Supported VM sizes

The complete list of supported VM sizes can be pulled programmatically. To learn how to retrieve them programmatically, refer to the Finding supported VM sizes section. Upgrading the VM size results in validation to check if the new VM size supports the EncryptionAtHost feature.

Prerequisites

You must enable the feature for your subscription before you use the EncryptionAtHost property for your VM/VMSS. Use the following steps to enable the feature for your

subscription:

1. Execute the following command to register the feature for your subscription

```PowerShell
 Register-AzProviderFeature -FeatureName "EncryptionAtHost" -
ProviderNamespace "Microsoft.Compute"
```

2. Check that the registration state is Registered (takes a few minutes) using the following command before trying out the feature.

```PowerShell
 Get-AzProviderFeature -FeatureName "EncryptionAtHost" -
ProviderNamespace "Microsoft.Compute"
```

Create an Azure Key Vault and DiskEncryptionSet

> **Note**
>
> This section only applies to configurations with customer-managed keys. If you're using platform-managed keys, you can skip to the Example scripts section.

Once the feature is enabled, you need to set up an Azure Key Vault and a DiskEncryptionSet, if you haven't already.

1. Make sure that you have installed latest Azure PowerShell version, and you are signed in to an Azure account in with Connect-AzAccount

2. Create an instance of Azure Key Vault and encryption key.

   When creating the Key Vault instance, you must enable purge protection. Purge protection ensures that a deleted key cannot be permanently deleted until the retention period lapses. These settings protect you from losing data due to accidental deletion. These settings are mandatory when using a Key Vault for encrypting managed disks.

```PowerShell
$ResourceGroupName="yourResourceGroupName"
$LocationName="westcentralus"
$keyVaultName="yourKeyVaultName"
$keyName="yourKeyName"
```

```
$keyDestination="Software"
$diskEncryptionSetName="yourDiskEncryptionSetName"

$keyVault = New-AzKeyVault -Name $keyVaultName `
-ResourceGroupName $ResourceGroupName `
-Location $LocationName `
-EnablePurgeProtection

$key = Add-AzKeyVaultKey -VaultName $keyVaultName `
      -Name $keyName `
      -Destination $keyDestination
```

3. Create an instance of a DiskEncryptionSet. You can set RotationToLatestKeyVersionEnabled equal to $true to enable automatic rotation of the key. When you enable automatic rotation, the system will automatically update all managed disks, snapshots, and images referencing the disk encryption set to use the new version of the key within one hour.

PowerShell

```
$desConfig=New-AzDiskEncryptionSetConfig -Location $LocationName `
    -SourceVaultId $keyVault.ResourceId `
    -KeyUrl $key.Key.Kid `
    -IdentityType SystemAssigned `
    -RotationToLatestKeyVersionEnabled $false

$des=New-AzDiskEncryptionSet -Name $diskEncryptionSetName `
       -ResourceGroupName $ResourceGroupName `
       -InputObject $desConfig
```

4. Grant the DiskEncryptionSet resource access to the key vault.

> **Note**
>
> It may take few minutes for Azure to create the identity of your DiskEncryptionSet in your Microsoft Entra ID. If you get an error like "Cannot find the Active Directory object" when running the following command, wait a few minutes and try again.

PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ObjectId
$des.Identity.PrincipalId -PermissionsToKeys wrapkey,unwrapkey,get
```

Use a key vault in a different subscription

Alternatively, you can manage your Azure Key Vaults centrally from a single subscription, and use the keys stored in the Key Vault to encrypt managed disks and snapshots in other subscriptions in your organization. This allows your security team to enforce and easily manage a robust security policy to a single subscription.

> **ⓘ Important**
>
> For this configuration, both your Key Vault and your disk encryption set must be in the same region and be using the same tenant.

The following script is an example of how you would configure a disk encryption set to use a key from a Key Vault in a different subscription, but same region:

Azure PowerShell

```
$sourceSubscriptionId="<sourceSubID>"
$sourceKeyVaultName="<sourceKVName>"
$sourceKeyName="<sourceKeyName>"

$targetSubscriptionId="<targetSubID>"
$targetResourceGroupName="<targetRGName>"
$targetDiskEncryptionSetName="<targetDiskEncSetName>"
$location="<targetRegion>"

Set-AzContext -Subscription $sourceSubscriptionId

$key = Get-AzKeyVaultKey -VaultName $sourceKeyVaultName -Name $sourceKeyName

Set-AzContext -Subscription $targetSubscriptionId

$desConfig=New-AzDiskEncryptionSetConfig -Location $location `
-KeyUrl $key.Key.Kid `
-IdentityType SystemAssigned `
-RotationToLatestKeyVersionEnabled $false

$des=New-AzDiskEncryptionSet -Name $targetDiskEncryptionSetName `
-ResourceGroupName $targetResourceGroupName `
-InputObject $desConfig
```

Enable encryption at host for disks attached to VM and Virtual Machine Scale Sets

You can enable encryption at host by setting a new property EncryptionAtHost under securityProfile of VMs or Virtual Machine Scale Sets using the API version **2020-06-01** and above.

```
"securityProfile": { "encryptionAtHost": "true" }
```

Example scripts

Create a VM with encryption at host enabled with customer-managed keys.

Create a VM with managed disks using the resource URI of the DiskEncryptionSet created earlier to encrypt cache of OS and data disks with customer-managed keys. The temp disks are encrypted with platform-managed keys.

PowerShell

```
$VMLocalAdminUser = "yourVMLocalAdminUserName"
$VMLocalAdminSecurePassword = ConvertTo-SecureString <password> -AsPlainText
-Force
$LocationName = "yourRegion"
$ResourceGroupName = "yourResourceGroupName"
$ComputerName = "yourComputerName"
$VMName = "yourVMName"
$VMSize = "yourVMSize"
$diskEncryptionSetName="yourdiskEncryptionSetName"

$NetworkName = "yourNetworkName"
$NICName = "yourNICName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -
AddressPrefix $SubnetAddressPrefix
$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName
$ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix
-Subnet $SingleSubnet
$NIC = New-AzNetworkInterface -Name $NICName -ResourceGroupName
$ResourceGroupName -Location $LocationName -SubnetId $Vnet.Subnets[0].Id

$Credential = New-Object System.Management.Automation.PSCredential
($VMLocalAdminUser, $VMLocalAdminSecurePassword);

# Enable encryption at host by specifying EncryptionAtHost parameter

$VirtualMachine = New-AzVMConfig -VMName $VMName -VMSize $VMSize -
EncryptionAtHost
$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -
ComputerName $ComputerName -Credential $Credential -ProvisionVMAgent -
EnableAutoUpdate
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName
```

```
'MicrosoftWindowsServer' -Offer 'WindowsServer' -Skus '2012-R2-Datacenter' -
Version latest

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName
$ResourceGroupName -Name $diskEncryptionSetName

# Enable encryption with a customer managed key for OS disk by setting
DiskEncryptionSetId property

$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name $($VMName
+"_OSDisk") -DiskEncryptionSetId $diskEncryptionSet.Id -CreateOption
FromImage

# Add a data disk encrypted with a customer managed key by setting
DiskEncryptionSetId property

$VirtualMachine = Add-AzVMDataDisk -VM $VirtualMachine -Name $($VMName
+"DataDisk1") -DiskSizeInGB 128 -StorageAccountType Premium_LRS -
CreateOption Empty -Lun 0 -DiskEncryptionSetId $diskEncryptionSet.Id

New-AzVM -ResourceGroupName $ResourceGroupName -Location $LocationName -VM
$VirtualMachine -Verbose
```

Create a VM with encryption at host enabled with platform-managed keys.

Create a VM with encryption at host enabled to encrypt cache of OS/data disks and temp disks with platform-managed keys.

PowerShell

```
$VMLocalAdminUser = "yourVMLocalAdminUserName"
$VMLocalAdminSecurePassword = ConvertTo-SecureString <password> -AsPlainText
-Force
$LocationName = "yourRegion"
$ResourceGroupName = "yourResourceGroupName"
$ComputerName = "yourComputerName"
$VMName = "yourVMName"
$VMSize = "yourVMSize"

$NetworkName = "yourNetworkName"
$NICName = "yourNICName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -
AddressPrefix $SubnetAddressPrefix
$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName
$ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix
-Subnet $SingleSubnet
```

```
$NIC = New-AzNetworkInterface -Name $NICName -ResourceGroupName
$ResourceGroupName -Location $LocationName -SubnetId $Vnet.Subnets[0].Id

$Credential = New-Object System.Management.Automation.PSCredential
($VMLocalAdminUser, $VMLocalAdminSecurePassword);

# Enable encryption at host by specifying EncryptionAtHost parameter

$VirtualMachine = New-AzVMConfig -VMName $VMName -VMSize $VMSize -
EncryptionAtHost

$VirtualMachine = Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -
ComputerName $ComputerName -Credential $Credential -ProvisionVMAgent -
EnableAutoUpdate
$VirtualMachine = Add-AzVMNetworkInterface -VM $VirtualMachine -Id $NIC.Id
$VirtualMachine = Set-AzVMSourceImage -VM $VirtualMachine -PublisherName
'MicrosoftWindowsServer' -Offer 'WindowsServer' -Skus '2012-R2-Datacenter' -
Version latest

$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name $($VMName
+"_OSDisk") -CreateOption FromImage

$VirtualMachine = Add-AzVMDataDisk -VM $VirtualMachine -Name $($VMName
+"DataDisk1") -DiskSizeInGB 128 -StorageAccountType Premium_LRS -
CreateOption Empty -Lun 0

New-AzVM -ResourceGroupName $ResourceGroupName -Location $LocationName -VM
$VirtualMachine
```

Update a VM to enable encryption at host.

PowerShell

```
$ResourceGroupName = "yourResourceGroupName"
$VMName = "yourVMName"

$VM = Get-AzVM -ResourceGroupName $ResourceGroupName -Name $VMName

Stop-AzVM -ResourceGroupName $ResourceGroupName -Name $VMName -Force

Update-AzVM -VM $VM -ResourceGroupName $ResourceGroupName -EncryptionAtHost
$true
```

Check the status of encryption at host for a VM

PowerShell

```
$ResourceGroupName = "yourResourceGroupName"
$VMName = "yourVMName"
```

```
$VM = Get-AzVM -ResourceGroupName $ResourceGroupName -Name $VMName

$VM.SecurityProfile.EncryptionAtHost
```

Disable encryption at host

You must deallocate your VM before you can disable encryption at host.

PowerShell

```
$ResourceGroupName = "yourResourceGroupName"
$VMName = "yourVMName"

$VM = Get-AzVM -ResourceGroupName $ResourceGroupName -Name $VMName

Stop-AzVM -ResourceGroupName $ResourceGroupName -Name $VMName -Force

Update-AzVM -VM $VM -ResourceGroupName $ResourceGroupName -EncryptionAtHost
$false
```

Create a Virtual Machine Scale Set with encryption at host enabled with customer-managed keys.

> **ⓘ Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Create a Virtual Machine Scale Set with managed disks using the resource URI of the DiskEncryptionSet created earlier to encrypt cache of OS and data disks with customer-managed keys. The temp disks are encrypted with platform-managed keys.

PowerShell

```
$VMLocalAdminUser = "yourLocalAdminUser"
$VMLocalAdminSecurePassword = ConvertTo-SecureString Password@123 -
AsPlainText -Force
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$ComputerNamePrefix = "yourComputerNamePrefix"
```

```
$VMScaleSetName = "yourVMSSName"
$VMSize = "Standard_DS3_v2"
$diskEncryptionSetName="yourDiskEncryptionSetName"

$NetworkName = "yourVNETName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -
AddressPrefix $SubnetAddressPrefix

$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName
$ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix
-Subnet $SingleSubnet

$ipConfig = New-AzVmssIpConfig -Name "myIPConfig" -SubnetId
$Vnet.Subnets[0].Id

# Enable encryption at host by specifying EncryptionAtHost parameter

$VMSS = New-AzVmssConfig -Location $LocationName -SkuCapacity 2 -SkuName
$VMSize -OrchestrationMode "Flexible" -EncryptionAtHost

$VMSS = Add-AzVmssNetworkInterfaceConfiguration -Name "myVMSSNetworkConfig"
-VirtualMachineScaleSet $VMSS -Primary $true -IpConfiguration $ipConfig

$diskEncryptionSet=Get-AzDiskEncryptionSet -ResourceGroupName
$ResourceGroupName -Name $diskEncryptionSetName

# Enable encryption with a customer managed key for the OS disk by setting
DiskEncryptionSetId property

$VMSS = Set-AzVmssStorageProfile $VMSS -OsDiskCreateOption "FromImage" -
DiskEncryptionSetId $diskEncryptionSet.Id -ImageReferenceOffer
'WindowsServer' -ImageReferenceSku '2012-R2-Datacenter' -
ImageReferenceVersion latest -ImageReferencePublisher
'MicrosoftWindowsServer'

$VMSS = Set-AzVmssOsProfile $VMSS -ComputerNamePrefix $ComputerNamePrefix -
AdminUsername $VMLocalAdminUser -AdminPassword $VMLocalAdminSecurePassword

# Add a data disk encrypted with a customer managed key by setting
DiskEncryptionSetId property

$VMSS = Add-AzVmssDataDisk -VirtualMachineScaleSet $VMSS -CreateOption Empty
-Lun 1 -DiskSizeGB 128 -StorageAccountType Premium_LRS -DiskEncryptionSetId
$diskEncryptionSet.Id
```

Create a Virtual Machine Scale Set with encryption at host enabled with platform-managed keys.

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to [Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub]

Create a Virtual Machine Scale Set with encryption at host enabled to encrypt cache of OS/data disks and temp disks with platform-managed keys.

PowerShell

```
$VMLocalAdminUser = "yourLocalAdminUser"
$VMLocalAdminSecurePassword = ConvertTo-SecureString Password@123 -
AsPlainText -Force
$LocationName = "westcentralus"
$ResourceGroupName = "yourResourceGroupName"
$ComputerNamePrefix = "yourComputerNamePrefix"
$VMScaleSetName = "yourVMSSName"
$VMSize = "Standard_DS3_v2"

$NetworkName = "yourVNETName"
$SubnetName = "yourSubnetName"
$SubnetAddressPrefix = "10.0.0.0/24"
$VnetAddressPrefix = "10.0.0.0/16"

$SingleSubnet = New-AzVirtualNetworkSubnetConfig -Name $SubnetName -
AddressPrefix $SubnetAddressPrefix

$Vnet = New-AzVirtualNetwork -Name $NetworkName -ResourceGroupName
$ResourceGroupName -Location $LocationName -AddressPrefix $VnetAddressPrefix
-Subnet $SingleSubnet

$ipConfig = New-AzVmssIpConfig -Name "myIPConfig" -SubnetId
$Vnet.Subnets[0].Id

# Enable encryption at host by specifying EncryptionAtHost parameter

$VMSS = New-AzVmssConfig -Location $LocationName -SkuCapacity 2 -SkuName
$VMSize -OrchestrationMode "Flexible" -EncryptionAtHost

$VMSS = Add-AzVmssNetworkInterfaceConfiguration -Name "myVMSSNetworkConfig"
-VirtualMachineScaleSet $VMSS -Primary $true -IpConfiguration $ipConfig

$VMSS = Set-AzVmssStorageProfile $VMSS -OsDiskCreateOption "FromImage" -
ImageReferenceOffer 'WindowsServer' -ImageReferenceSku '2012-R2-Datacenter'
-ImageReferenceVersion latest -ImageReferencePublisher
'MicrosoftWindowsServer'
```

```
$VMSS = Set-AzVmssOsProfile $VMSS -ComputerNamePrefix $ComputerNamePrefix -
AdminUsername $VMLocalAdminUser -AdminPassword $VMLocalAdminSecurePassword

$VMSS = Add-AzVmssDataDisk -VirtualMachineScaleSet $VMSS -CreateOption Empty
-Lun 1 -DiskSizeGB 128 -StorageAccountType Premium_LRS

$Credential = New-Object System.Management.Automation.PSCredential
($VMLocalAdminUser, $VMLocalAdminSecurePassword);

New-AzVmss -VirtualMachineScaleSet $VMSS -ResourceGroupName
$ResourceGroupName -VMScaleSetName $VMScaleSetName
```

Update a Virtual Machine Scale Set to enable encryption at host.

PowerShell

```
$ResourceGroupName = "yourResourceGroupName"
$VMScaleSetName = "yourVMSSName"

$VMSS = Get-AzVmss -ResourceGroupName $ResourceGroupName -Name
$VMScaleSetName

Update-AzVmss -VirtualMachineScaleSet $VMSS -Name $VMScaleSetName -
ResourceGroupName $ResourceGroupName -EncryptionAtHost $true
```

Check the status of encryption at host for a Virtual Machine Scale Set

PowerShell

```
$ResourceGroupName = "yourResourceGroupName"
$VMScaleSetName = "yourVMSSName"

$VMSS = Get-AzVmss -ResourceGroupName $ResourceGroupName -Name
$VMScaleSetName

$VMSS.VirtualMachineProfile.SecurityProfile.EncryptionAtHost
```

Update a Virtual Machine Scale Set to disable encryption at host.

You can disable encryption at host on your Virtual Machine Scale Set but, this will only affect VMs created after you disable encryption at host. For existing VMs, you must

deallocate the VM, disable encryption at host on that individual VM, then reallocate the VM.

PowerShell

```
$ResourceGroupName = "yourResourceGroupName"
$VMScaleSetName = "yourVMSSName"

$VMSS = Get-AzVmss -ResourceGroupName $ResourceGroupName -Name
$VMScaleSetName

Update-AzVmss -VirtualMachineScaleSet $VMSS -Name $VMScaleSetName -
ResourceGroupName $ResourceGroupName -EncryptionAtHost $false
```

Finding supported VM sizes

Legacy VM Sizes aren't supported. You can find the list of supported VM sizes by either:

Calling the Resource Skus API and checking that the `EncryptionAtHostSupported` capability is set to **True**.

JSON

```
    {
        "resourceType": "virtualMachines",
        "name": "Standard_DS1_v2",
        "tier": "Standard",
        "size": "DS1_v2",
        "family": "standardDSv2Family",
        "locations": [
        "CentralUS"
        ],
        "capabilities": [
        {
            "name": "EncryptionAtHostSupported",
            "value": "True"
        }
        ]
    }
```

Or, calling the Get-AzComputeResourceSku PowerShell cmdlet.

PowerShell

```
Get-AzComputeResourceSku -Location "centralus" |
    Where-Object { $_.ResourceType -eq 'virtualMachines' -and
$_.capabilities.where({ $_.Name -eq 'EncryptionAtHostSupported' },
```

```
'First').Value -eq 'True' } |
    Select-Object -ExpandProperty Name
```

Next steps

Now that you've created and configured these resources, you can use them to secure your managed disks. The following link contains example scripts, each with a respective scenario, that you can use to secure your managed disks.

Azure Resource Manager template samples

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use the Azure CLI to enable end-to-end encryption using encryption at host

06/26/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When you enable encryption at host, data stored on the VM host is encrypted at rest and flows encrypted to the Storage service. For conceptual information on encryption at host, and other managed disk encryption types, see Encryption at host - End-to-end encryption for your VM data.

Restrictions

- Can't be enabled on virtual machines (VMs) or virtual machine scale sets that currently or ever had Azure Disk Encryption enabled.
- Azure Disk Encryption can't be enabled on disks that have encryption at host enabled.
- The encryption can be enabled on existing virtual machine scale sets. However, only new VMs created after enabling the encryption are automatically encrypted.
- Existing VMs must be deallocated and reallocated in order to be encrypted.

The following restrictions only apply to Ultra Disks and Premium SSD v2:

- Disks using a 512e sector size need to have been created after 5/13/2023.
  - If your disk was created before this date, snapshot your disk, and create a new disk using the snapshot.

Supported VM sizes

The complete list of supported VM sizes can be pulled programmatically. To learn how to retrieve them programmatically, see the Finding supported VM sizes section. Upgrading the VM size results in validation to check if the new VM size supports the EncryptionAtHost feature.

Prerequisites

You must enable the feature for your subscription before you use the EncryptionAtHost property for your VM/VMSS. Use the following steps to enable the feature for your subscription:

- Execute the following command to register the feature for your subscription

Azure CLI

```
az feature register --namespace Microsoft.Compute --name EncryptionAtHost
```

- Check that the registration state is **Registered** (takes a few minutes) using the command below before trying out the feature.

Azure CLI

```
az feature show --namespace Microsoft.Compute --name EncryptionAtHost
```

Create resources

> ⓘ **Note**
>
> This section only applies to configurations with customer-managed keys. If you're using platform-managed keys, you can skip to the **Example scripts** section.

Once the feature is enabled, you need to set up a DiskEncryptionSet and either an Azure Key Vault or an Azure Key Vault Managed HSM.

Azure Key Vault

- Install the latest Azure CLI and log to an Azure account in with az login.
- Create an Azure Key Vault and encryption key.

When creating the Key Vault, you must enable purge protection. Purge protection ensures that a deleted key can't be permanently deleted until the retention period lapses. These settings protect you from losing data due to accidental deletion. These settings are mandatory when using a Key Vault for encrypting managed disks.

> ⓘ **Important**
>
> Don't camel case the region, if you do so, you may experience problems when assigning additional disks to the resource in the Azure portal.

Azure CLI

```
subscriptionId=yourSubscriptionID
rgName=yourResourceGroupName
location=westcentralus
```

```
keyVaultName=yourKeyVaultName
keyName=yourKeyName
diskEncryptionSetName=yourDiskEncryptionSetName
diskName=yourDiskName

az account set --subscription $subscriptionId

az group create --resource-group $rgName --location $location

az keyvault create -n $keyVaultName \
-g $rgName \
-l $location \
--enable-purge-protection true

az keyvault key create --vault-name $keyVaultName \
-n $keyName \
--protection software
```

- Create a DiskEncryptionSet. You can set *enable-auto-key-rotation* to **true** to enable automatic rotation of the key. When you enable automatic rotation, the system automatically updates all managed disks, snapshots, and images referencing the disk encryption set to use the new version of the key within one hour.

Azure CLI

```
keyVaultKeyUrl=$(az keyvault key show --vault-name $keyVaultName --name $keyName -
-query [key.kid] -o tsv)

az disk-encryption-set create -n $diskEncryptionSetName \
-l $location \
-g $rgName \
--key-url $keyVaultKeyUrl \
--enable-auto-key-rotation false
```

- Grant the DiskEncryptionSet resource access to the key vault.

> **Note**
>
> It may take few minutes for Azure to create the identity of your DiskEncryptionSet in your Microsoft Entra ID. If you get an error like "Cannot find the Active Directory object" when running the following command, wait a few minutes and try again.

Azure CLI

```
desIdentity=$(az disk-encryption-set show -n $diskEncryptionSetName -g $rgName --
query [identity.principalId] -o tsv)

az keyvault set-policy -n $keyVaultName \
```

```
-g $rgName \
--object-id $desIdentity \
--key-permissions wrapkey unwrapkey get
```

Azure Key Vault Managed HSM

Alternatively, you can use a Managed HSM to handle your keys.

To do this, you must complete the following prerequisites:

- Install the latest Azure CLI and log in to an Azure account in with az login.
- Create and configure a managed HSM.
- Assign permissions to a user, so they can manage your Managed HSM.

Configuration

Once you've created a Managed HSM and added permissions, enable purge protection and create an encryption key.

Azure CLI

```
subscriptionId=yourSubscriptionID
rgName=yourResourceGroupName
location=westcentralus
keyVaultName=yourKeyVaultName
keyName=yourKeyName
diskEncryptionSetName=yourDiskEncryptionSetName
diskName=yourDiskName

az account set --subscription $subscriptionId

az keyvault update-hsm --subscription $subscriptionId -g $rgName --hsm-name
$keyVaultName --enable-purge-protection true

az keyvault key create --hsm-name  $keyVaultName --name $keyName --ops wrapKey
unwrapKey --kty RSA-HSM --size 2048
```

Then, create a DiskEncryptionSet.

Azure CLI

```
keyVaultKeyUrl=$(az keyvault key show --vault-name $keyVaultName --name $keyName -
-query [key.kid] -o tsv)

az disk-encryption-set create -n $diskEncryptionSetName \
-l $location \
-g $rgName \
```

```
--key-url $keyVaultKeyUrl \
--enable-auto-key-rotation false
```

Finally, grant the DiskEncryptionSet access to the Managed HSM.

Azure CLI

```
desIdentity=$(az disk-encryption-set show -n $diskEncryptionSetName -g $rgName --
query [identity.principalId] -o tsv)

az keyvault role assignment create --hsm-name $keyVaultName --role "Managed HSM
Crypto Service Encryption User" --assignee $desIdentity --scope /keys
```

Example scripts

Create a VM with encryption at host enabled with customer-managed keys

Create a VM with managed disks using the resource URI of the DiskEncryptionSet created earlier to encrypt cache of OS and data disks with customer-managed keys. The temp disks are encrypted with platform-managed keys.

Azure CLI

```
rgName=yourRGName
vmName=yourVMName
location=eastus
vmSize=Standard_DS2_v2
image=LinuxImageURN
diskEncryptionSetName=yourDiskEncryptionSetName

diskEncryptionSetId=$(az disk-encryption-set show -n $diskEncryptionSetName -g
$rgName --query [id] -o tsv)

az vm create -g $rgName \
-n $vmName \
-l $location \
--encryption-at-host \
--image $image \
--size $vmSize \
--generate-ssh-keys \
--os-disk-encryption-set $diskEncryptionSetId \
--data-disk-sizes-gb 128 128 \
--data-disk-encryption-sets $diskEncryptionSetId $diskEncryptionSetId
```

Create a VM with encryption at host enabled with platform-managed keys

Create a VM with encryption at host enabled to encrypt cache of OS/data disks and temp disks with platform-managed keys.

Azure CLI

```
rgName=yourRGName
vmName=yourVMName
location=eastus
vmSize=Standard_DS2_v2
image=LinuxImageURN

az vm create -g $rgName \
-n $vmName \
-l $location \
--encryption-at-host \
--image $image \
--size $vmSize \
--generate-ssh-keys \
--data-disk-sizes-gb 128 128 \
```

Update a VM to enable encryption at host

Azure CLI

```
rgName=yourRGName
vmName=yourVMName

az vm update -n $vmName \
-g $rgName \
--set securityProfile.encryptionAtHost=true
```

Check the status of encryption at host for a VM

Azure CLI

```
rgName=yourRGName
vmName=yourVMName

az vm show -n $vmName \
-g $rgName \
--query [securityProfile.encryptionAtHost] -o tsv
```

Update a VM to disable encryption at host

You must deallocate your VM before you can disable encryption at host.

Azure CLI

```
rgName=yourRGName
vmName=yourVMName

az vm update -n $vmName \
-g $rgName \
--set securityProfile.encryptionAtHost=false
```

Create a Virtual Machine Scale Set with encryption at host enabled with customer-managed keys

Create a Virtual Machine Scale Set with managed disks using the resource URI of the DiskEncryptionSet created earlier to encrypt cache of OS and data disks with customer-managed keys. The temp disks are encrypted with platform-managed keys.

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Azure CLI

```
rgName=yourRGName
vmssName=yourVMSSName
location=westus2
vmSize=Standard_DS3_V2
image=Ubuntu2204
diskEncryptionSetName=yourDiskEncryptionSetName

diskEncryptionSetId=$(az disk-encryption-set show -n $diskEncryptionSetName -g
$rgName --query [id] -o tsv)

az vmss create -g $rgName \
-n $vmssName \
--encryption-at-host \
--image $image \
--orchestration-mode flexible \
--admin-username azureuser \
--generate-ssh-keys \
```

```
--os-disk-encryption-set $diskEncryptionSetId \
--data-disk-sizes-gb 64 128 \
--data-disk-encryption-sets $diskEncryptionSetId $diskEncryptionSetId
```

Create a Virtual Machine Scale Set with encryption at host enabled with platform-managed keys

Create a Virtual Machine Scale Set with encryption at host enabled to encrypt cache of OS/data disks and temp disks with platform-managed keys.

> **Important**
>
> Starting November 2023, VM scale sets created using PowerShell and Azure CLI will default to Flexible Orchestration Mode if no orchestration mode is specified. For more information about this change and what actions you should take, go to Breaking Change for VMSS PowerShell/CLI Customers - Microsoft Community Hub

Azure CLI

```
rgName=yourRGName
vmssName=yourVMSSName
location=westus2
vmSize=Standard_DS3_V2
image=Ubuntu2204

az vmss create -g $rgName \
-n $vmssName \
--encryption-at-host \
--image $image \
--orchestration-mode flexible \
--admin-username azureuser \
--generate-ssh-keys \
--data-disk-sizes-gb 64 128 \
```

Update a Virtual Machine Scale Set to enable encryption at host

Azure CLI

```
rgName=yourRGName
vmssName=yourVMName

az vmss update -n $vmssName \
```

```
-g $rgName \
--set virtualMachineProfile.securityProfile.encryptionAtHost=true
```

Check the status of encryption at host for a Virtual Machine Scale Set

Azure CLI

```
rgName=yourRGName
vmssName=yourVMName

az vmss show -n $vmssName \
-g $rgName \
--query [virtualMachineProfile.securityProfile.encryptionAtHost] -o tsv
```

Update a Virtual Machine Scale Set to disable encryption at host

You can disable encryption at host on your Virtual Machine Scale Set but, this will only affect VMs created after you disable encryption at host. For existing VMs, you must deallocate the VM, disable encryption at host on that individual VM, then reallocate the VM.

Azure CLI

```
rgName=yourRGName
vmssName=yourVMName

az vmss update -n $vmssName \
-g $rgName \
--set virtualMachineProfile.securityProfile.encryptionAtHost=false
```

Finding supported VM sizes

Legacy VM Sizes aren't supported. You can find the list of supported VM sizes by either using resource SKU APIs or the Azure CLI.

When calling the Resource Skus API, check that the `EncryptionAtHostSupported` capability is set to **True**.

JSON

```
    {
        "resourceType": "virtualMachines",
```

```
        "name": "Standard_DS1_v2",
        "tier": "Standard",
        "size": "DS1_v2",
        "family": "standardDSv2Family",
        "locations": [
        "CentralUS"
        ],
        "capabilities": [
        {
            "name": "EncryptionAtHostSupported",
            "value": "True"
        }
        ]
    }
```

For the Azure CLI, use the az vm image list-skus command.

Azure CLI

```
location=centralus

az vm list-skus --location $location --all \
--resource-type virtualMachines \
--query "[?capabilities[?name=='EncryptionAtHostSupported' && value=='True']].
{VMName:name, EncryptionAtHost:capabilities[?
name=='EncryptionAtHostSupported'].value | [0]}" \
--output table
```

Next steps

Now that you've created and configured these resources, you can use them to secure your managed disks. The following link contains example scripts, each with a respective scenario, that you can use to secure your managed disks.

Azure Resource Manager template samples

Enable double encryption at rest for managed disks

Article • 09/27/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔

Azure Disk Storage supports double encryption at rest for managed disks. For conceptual information on double encryption at rest, and other managed disk encryption types, see the Double encryption at rest section of our disk encryption article.

Restrictions

Double encryption at rest isn't currently supported with either Ultra Disks or Premium SSD v2 disks.

Prerequisites

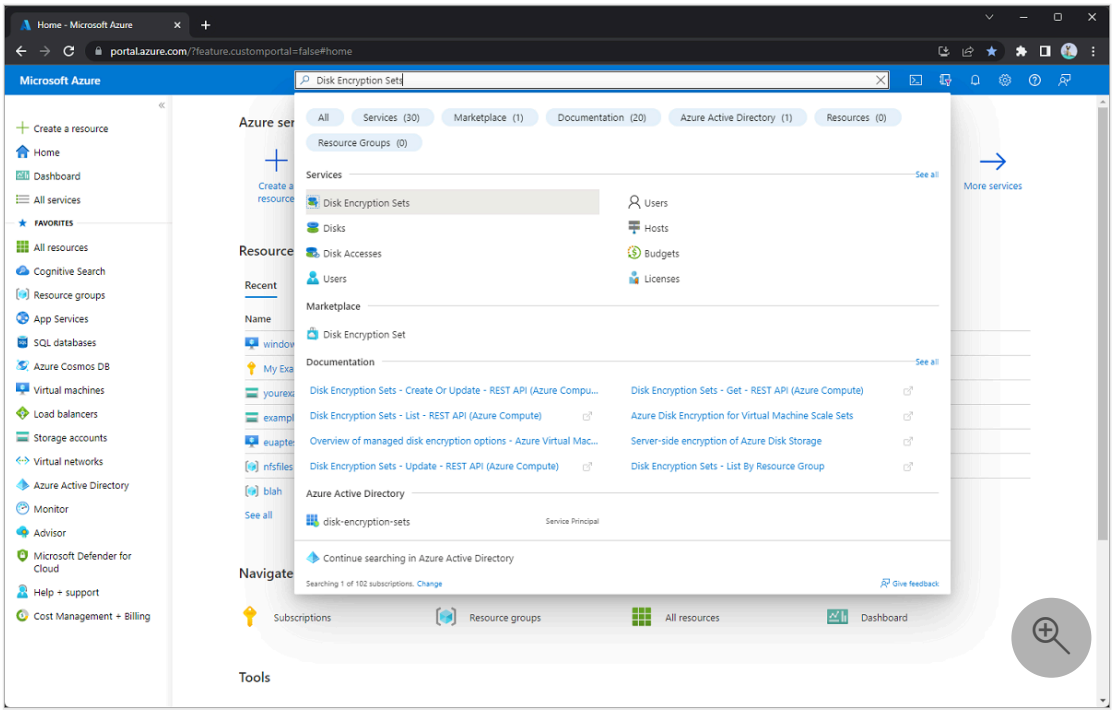
If you're going to use Azure CLI, install the latest Azure CLI and sign in to an Azure account with az login.

If you're going to use the Azure PowerShell module, install the latest Azure PowerShell version, and sign in to an Azure account using Connect-AzAccount.

Getting started

Azure portal

1. Sign in to the Azure portal.
2. Search for and select **Disk Encryption Sets**.



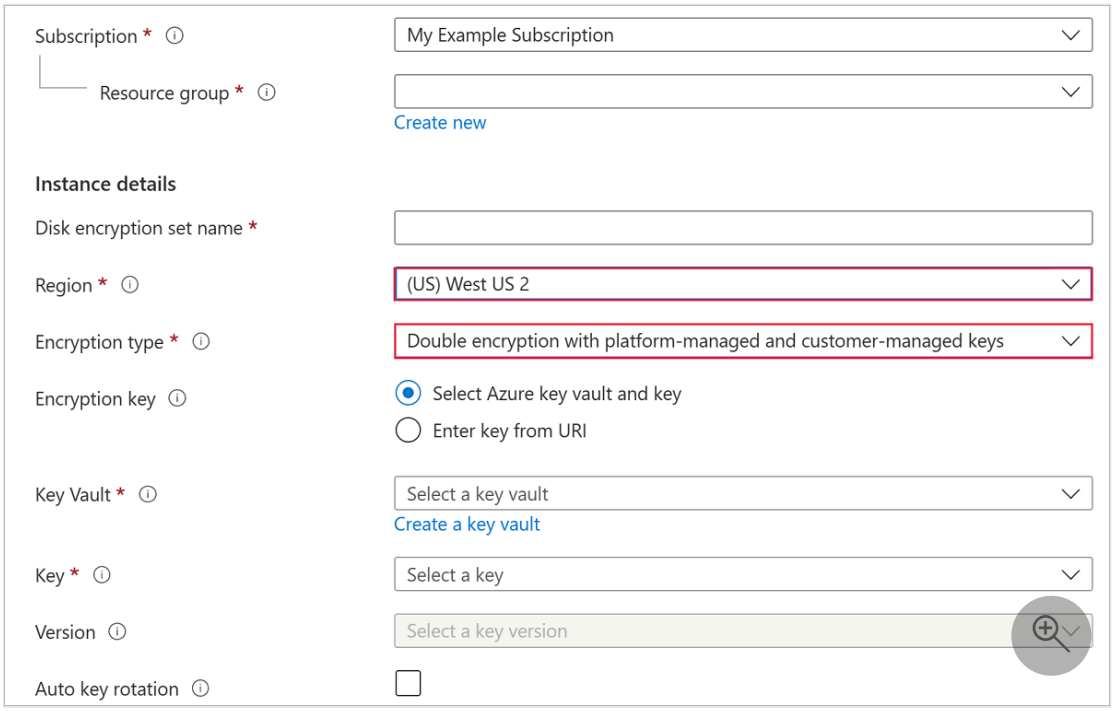
3. Select + **Create**.

4. Select one of the supported regions.

5. For **Encryption type**, select **Double encryption with platform-managed and customer-managed keys**.

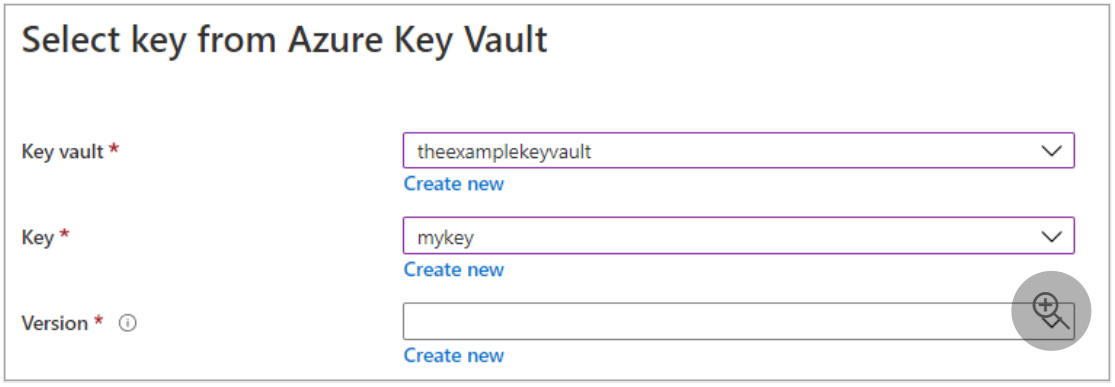
> ⓘ **Note**
>
> Once you create a disk encryption set with a particular encryption type, it cannot be changed. If you want to use a different encryption type, you must create a new disk encryption set.

6. Fill in the remaining info.



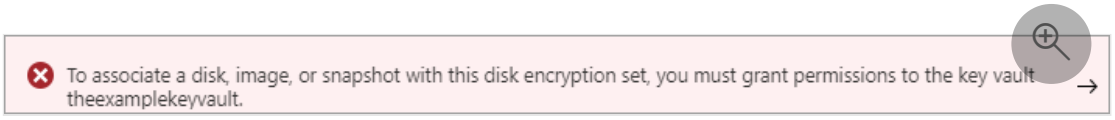
7. Select an Azure Key Vault and key, or create a new one if necessary.

> **Note**
>
> If you create a Key Vault instance, you must enable soft delete and purge protection. These settings are mandatory when using a Key Vault for encrypting managed disks, and protect you from losing data due to accidental deletion.

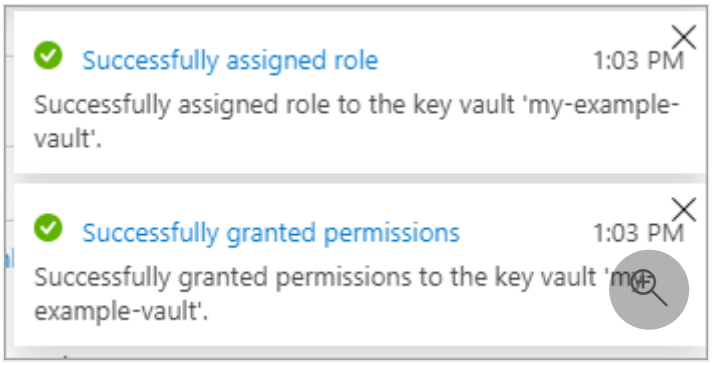


8. Select **Create**.

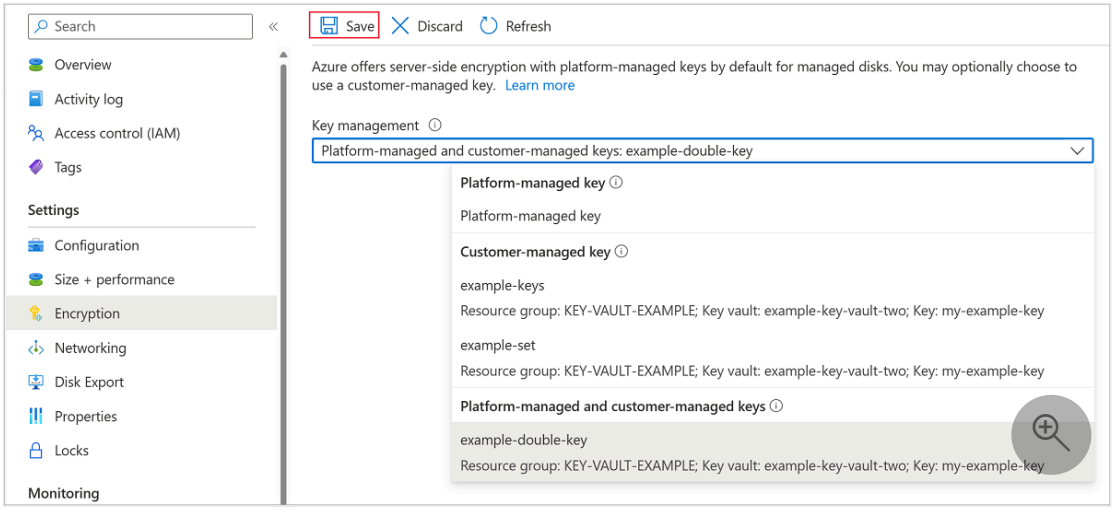
9. Navigate to the disk encryption set you created, and select the error that is displayed. This will configure your disk encryption set to work.



A notification should pop up and succeed. Doing this will allow you to use the disk encryption set with your key vault.



10. Navigate to your disk.
11. Select **Encryption**.
12. For **Key management**, select one of the keys under **Platform-managed and customer-managed keys**.
13. select **Save**.



You have now enabled double encryption at rest on your managed disk.

Next steps

- Azure PowerShell - Enable customer-managed keys with server-side encryption - managed disks
- Azure Resource Manager template samples
- Enable customer-managed keys with server-side encryption - Examples

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Disk Encryption for Linux VMs

Summarize this article for me

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

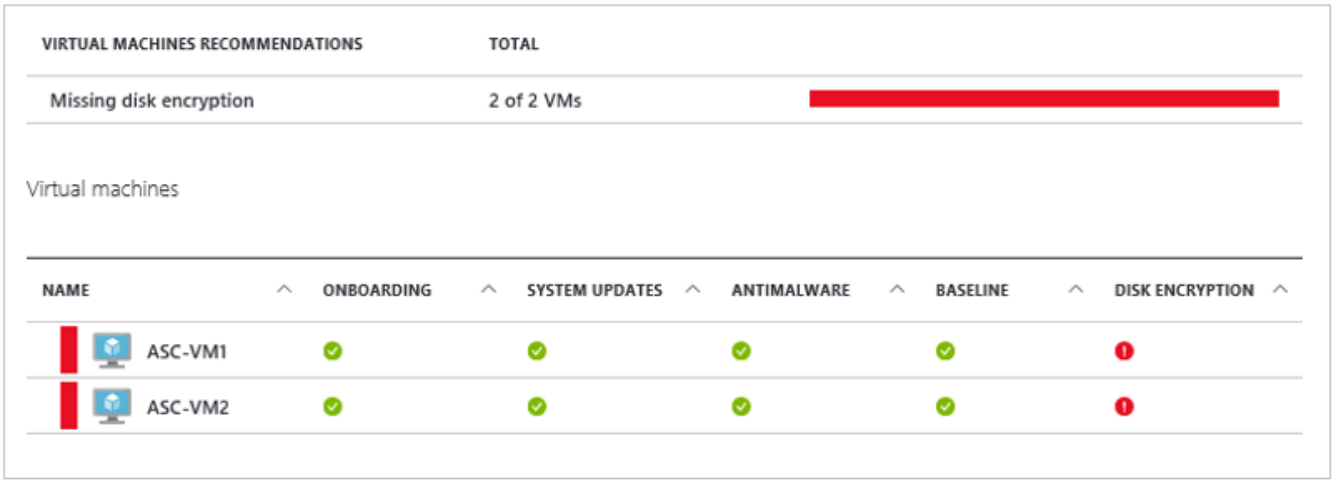
> **Caution**
>
> **CentOS End Of Life Notice**: This article references CentOS, which has reached End Of Life status. If you're using CentOS, review your deployment plans and consider migrating to a supported Linux distribution. For migration guidance and timelines, see CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure Disk Encryption helps protect and safeguard your data to meet your organizational security and compliance commitments. It uses the DM-Crypt feature of Linux to provide volume encryption for the OS and data disks of Azure virtual machines (VMs), and is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets.

Azure Disk Encryption is zone resilient, the same way as Virtual Machines. For details, see Azure Services that support Availability Zones.

If you use Microsoft Defender for Cloud, you're alerted if you have VMs that aren't encrypted. The alerts show as High Severity and the recommendation is to encrypt these VMs.



> ⚠ **Warning**
>
> - If you have previously used Azure Disk Encryption with Microsoft Entra ID to encrypt a VM, you must continue to use this option to encrypt your VM. See Azure Disk Encryption with Microsoft Entra ID (previous release) for details.
> - Certain recommendations might increase data, network, or compute resource usage, resulting in additional license or subscription costs. You must have a valid active Azure subscription to create resources in Azure in the supported regions.

You can learn the fundamentals of Azure Disk Encryption for Linux in just a few minutes with the Create and encrypt a Linux VM with Azure CLI quickstart or the Create and encrypt a Linux VM with Azure PowerShell quickstart.

Supported VMs and operating systems

Supported VMs

Linux VMs are available in a range of sizes. Azure Disk Encryption is supported on Generation 1 and Generation 2 VMs. Azure Disk Encryption is also available for VMs with premium storage.

See Azure VM sizes with no local temporary disk.

Azure Disk Encryption is not available on Basic, A-series VMs,v6 series VMs, v7 series VMs, or newer, or on virtual machines that do not meet these minimum memory requirements:

Memory requirements

Expand table

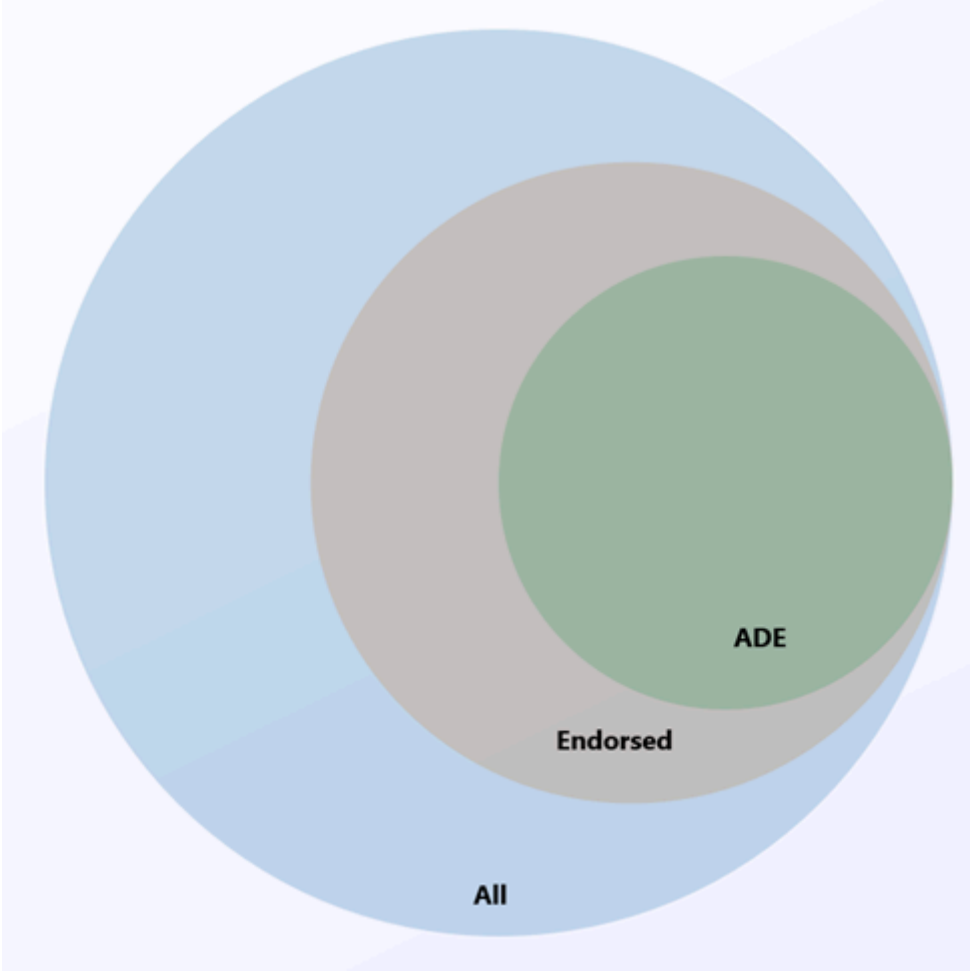
Virtual machine	Minimum memory requirement
Linux VMs when only encrypting data volumes	2 GB
Linux VMs when encrypting both data and OS volumes, and where the root (/) file system usage is 4 GB or less	8 GB
Linux VMs when encrypting both data and OS volumes, and where the root (/) file system usage is greater than 4 GB	The root file system usage * 2. For instance, a 16 GB of root file system usage requires at least 32 GB of RAM

Once the OS disk encryption process is complete on Linux virtual machines, the VM can be configured to run with less memory.

For more exceptions, see Azure Disk Encryption: Restrictions.

Supported operating systems

Azure Disk Encryption is supported on a subset of the Azure-endorsed Linux distributions, which is itself a subset of all Linux server possible distributions.



Linux server distributions that are not endorsed by Azure do not support Azure Disk Encryption; of those that are endorsed, only the following distributions and versions support Azure Disk Encryption:

Expand table

Publisher	Offer	SKU	URN	Volume type supported for encryption
Canonical	Ubuntu	24.04-LTS	Canonical:ubuntu-24_04-lts-daily:server-gen1:latest	OS and data disk
Canonical	Ubuntu	24.04-LTS Gen 2	Canonical:ubuntu-24_04-lts:server:latest	OS and data disk
Canonical	Ubuntu	22.04-LTS	Canonical:0001-com-ubuntu-server-jammy:22_04-lts:latest	OS and data disk
Canonical	Ubuntu	22.04-LTS Gen2	Canonical:0001-com-ubuntu-server-jammy:22_04-lts-gen2:latest	OS and data disk
Canonical	Ubuntu	20.04-LTS	Canonical:0001-com-ubuntu-server-focal:20_04-lts:latest	OS and data disk
Canonical	Ubuntu	20.04-DAILY-LTS	Canonical:0001-com-ubuntu-server-focal-daily:20_04-daily-lts:latest	OS and data disk
Canonical	Ubuntu	20.04-LTS Gen2	Canonical:0001-com-ubuntu-server-focal:20_04-lts-gen2:latest	OS and data disk
Canonical	Ubuntu	20.04-DAILY-LTS Gen2	Canonical:0001-com-ubuntu-server-focal-daily:20_04-daily-lts-gen2:latest	OS and data disk
Canonical	Ubuntu	18.04-LTS	Canonical:UbuntuServer:18.04-LTS:latest	OS and data disk
Canonical	Ubuntu 18.04	18.04-DAILY-LTS	Canonical:UbuntuServer:18.04-DAILY-LTS:latest	OS and data disk
MicrosoftCBLMariner	cbl-mariner	cbl-mariner-2	MicrosoftCBLMariner:cbl-mariner:cbl-mariner-2:latest*	OS and data disk
MicrosoftCBLMariner	cbl-mariner	cbl-mariner-2-gen2	MicrosoftCBLMariner:cbl-mariner:cbl-mariner-2-gen2:latest*	OS and data disk
OpenLogic	CentOS 8-LVM	8-LVM	OpenLogic:CentOS-LVM:8-LVM:latest	OS and data disk

Publisher	Offer	SKU	URN	Volume type supported for encryption
OpenLogic	CentOS 8.4	8_4	OpenLogic:CentOS:8_4:latest	OS and data disk
OpenLogic	CentOS 8.3	8_3	OpenLogic:CentOS:8_3:latest	OS and data disk
OpenLogic	CentOS 8.2	8_2	OpenLogic:CentOS:8_2:latest	OS and data disk
OpenLogic	CentOS 7-LVM	7-LVM	OpenLogic:CentOS-LVM:7-LVM:7.9.2021020400	OS and data disk
OpenLogic	CentOS 7.9	7_9	OpenLogic:CentOS:7_9:latest	OS and data disk
OpenLogic	CentOS 7.8	7_8	OpenLogic:CentOS:7_8:latest	OS and data disk
OpenLogic	CentOS 7.7	7.7	OpenLogic:CentOS:7.7:latest	OS and data disk
OpenLogic	CentOS 7.6	7.6	OpenLogic:CentOS:7.6:latest	OS and data disk
OpenLogic	CentOS 7.5	7.5	OpenLogic:CentOS:7.5:latest	OS and data disk
OpenLogic	CentOS 7.4	7.4	OpenLogic:CentOS:7.4:latest	OS and data disk
OpenLogic	CentOS 6.8	6.8	OpenLogic:CentOS:6.8:latest	Data disk only
Oracle	Oracle Linux 8.6	8.6	Oracle:Oracle-Linux:ol86-lvm:latest	OS and data disk (see note)
Oracle	Oracle Linux 8.6 Gen 2	8.6	Oracle:Oracle-Linux:ol86-lvm-gen2:latest	OS and data disk (see note)
Oracle	Oracle Linux 8.5	8.5	Oracle:Oracle-Linux:ol85-lvm:latest	OS and data disk (see note)
Oracle	Oracle Linux 8.5 Gen 2	8.5	Oracle:Oracle-Linux:ol85-lvm-gen2:latest	OS and data disk (see note)
RedHat	RHEL 9.4	9.4	RedHat:RHEL:9_4:latest	OS and data disk (see note)

Publisher	Offer	SKU	URN	Volume type supported for encryption
RedHat	RHEL 9.4 Gen 2	9.4	RedHat:RHEL:94_gen2:latest	OS and data disk (see note)
RedHat	RHEL 9.3	9.3	RedHat:RHEL:9_3:latest	OS and data disk (see note)
RedHat	RHEL 9.3 Gen 2	9.3	RedHat:RHEL:93-gen2:latest	OS and data disk (see note)
RedHat	RHEL 9.2	9.2	RedHat:RHEL:9_2:latest	OS and data disk (see note)
RedHat	RHEL 9.2 Gen 2	9.2	RedHat:RHEL:92-gen2:latest	OS and data disk (see note)
RedHat	RHEL 9.0	9.0	RedHat:RHEL:9_0:latest	OS and data disk (see note)
RedHat	RHEL 9.0 Gen 2	9.0	RedHat:RHEL:90-gen2:latest	OS and data disk (see note)
RedHat	RHEL 9-lvm	9-lvm	RedHat:RHEL:9-lvm:latest	OS and data disk (see note)
RedHat	RHEL 9-lvm Gen 2	9-lvm-gen2	RedHat:RHEL:9-lvm-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.9	8.9	RedHat:RHEL:8_9:latest	OS and data disk (see note)
RedHat	RHEL 8.9 Gen 2	8.9	RedHat:RHEL:89-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.8	8.8	RedHat:RHEL:8_8:latest	OS and data disk (see note)
RedHat	RHEL 8.8 Gen 2	8.8	RedHat:RHEL:88-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.7	8.7	RedHat:RHEL:8_7:latest	OS and data disk (see note)
RedHat	RHEL 8.7 Gen 2	8.7	RedHat:RHEL:87-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.6	8.6	RedHat:RHEL:8_6:latest	OS and data disk (see note)

Publisher	Offer	SKU	URN	Volume type supported for encryption
RedHat	RHEL 8.6 Gen 2	8.6	RedHat:RHEL:86-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.5	8.5	RedHat:RHEL:8_5:latest	OS and data disk (see note)
RedHat	RHEL 8.5 Gen 2	8.5	RedHat:RHEL:85-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.4	8.4	RedHat:RHEL:8.4:latest	OS and data disk (see note)
RedHat	RHEL 8.3	8.3	RedHat:RHEL:8.3:latest	OS and data disk (see note)
RedHat	RHEL 8-LVM	8-LVM	RedHat:RHEL:8-LVM:latest	OS and data disk (see note)
RedHat	RHEL 8-LVM Gen 2	8-lvm-gen2	RedHat:RHEL:8-lvm-gen2:latest	OS and data disk (see note)
RedHat	RHEL 8.2	8.2	RedHat:RHEL:8.2:latest	OS and data disk (see note)
RedHat	RHEL 8.1	8.1	RedHat:RHEL:8.1:latest	OS and data disk (see note)
RedHat	RHEL 7-LVM	7-LVM	RedHat:RHEL:7-LVM:7.9.2020111202	OS and data disk (see note)
RedHat	RHEL 7.9	7_9	RedHat:RHEL:7_9:latest	OS and data disk (see note)
RedHat	RHEL 7.8	7.8	RedHat:RHEL:7.8:latest	OS and data disk (see note)
RedHat	RHEL 7.7	7.7	RedHat:RHEL:7.7:latest	OS and data disk (see note)
RedHat	RHEL 7.6	7.6	RedHat:RHEL:7.6:latest	OS and data disk (see note)
RedHat	RHEL 7.5	7.5	RedHat:RHEL:7.5:latest	OS and data disk (see note)
RedHat	RHEL 7.4	7.4	RedHat:RHEL:7.4:latest	OS and data disk (see note)

Publisher	Offer	SKU	URN	Volume type supported for encryption
RedHat	RHEL 6.8	6.8	RedHat:RHEL:6.8:latest	Data disk (see note)
RedHat	RHEL 6.7	6.7	RedHat:RHEL:6.7:latest	Data disk (see note)
SUSE	openSUSE 42.3	42.3	SUSE:openSUSE-Leap:42.3:latest	Data disk only
SUSE	SLES 12-SP4	12-SP4	SUSE:SLES:12-SP4:latest	Data disk only
SUSE	SLES HPC 12-SP3	12-SP3	SUSE:SLES-HPC:12-SP3:latest	Data disk only

* For image versions greater than or equal to May 2023.

> ⓘ **Note**
>
> RHEL:
>
> - The new Azure Disk Encryption implementation is supported for RHEL OS and data disk for RHEL7 pay-as-you-go images.
> - ADE is also supported for RHEL Bring-Your-Own-Subscription Gold Images, but only **after** the subscription has been registered . For more information, see Red Hat Enterprise Linux Bring-Your-Own-Subscription Gold Images in Azure
>
> All distros:
>
> - ADE support for a particular offer type does not extend beyond the end-of-life date provided by the publisher.
> - The legacy ADE solution (using Microsoft Entra credentials) is not recommended for new VMs and is not compatible with RHEL versions later than RHEL 7.8 or with Python 3 as default.

Additional VM requirements

Azure Disk Encryption requires the dm-crypt and vfat modules to be present on the system. Removing or disabling vfat from the default image will prevent the system from reading the

key volume and obtaining the key needed to unlock the disks on subsequent reboots. System hardening steps that remove the vfat module from the system or enforce expanding the OS mountpoints/folders on data drives are not compatible with Azure Disk Encryption.

Before enabling encryption, the data disks to be encrypted must be properly listed in /etc/fstab. Use the "nofail" option when creating entries, and choose a persistent block device name (as device names in the "/dev/sdX" format may not be associated with the same disk across reboots, particularly after encryption; for more detail on this behavior, see: Troubleshoot Linux VM device name changes).

Make sure the /etc/fstab settings are configured properly for mounting. To configure these settings, run the mount -a command or reboot the VM and trigger the remount that way. Once that is complete, check the output of the lsblk command to verify that the drive is still mounted.

- If the /etc/fstab file doesn't mount the drive properly before enabling encryption, Azure Disk Encryption won't be able to mount it properly.
- The Azure Disk Encryption process will move the mount information out of /etc/fstab and into its own configuration file as part of the encryption process. Don't be alarmed to see the entry missing from /etc/fstab after data drive encryption completes.
- Before starting encryption, be sure to stop all services and processes that could be writing to mounted data disks and disable them, so that they do not restart automatically after a reboot. These could keep files open on these partitions, preventing the encryption procedure to remount them, causing failure of the encryption.
- After reboot, it will take time for the Azure Disk Encryption process to mount the newly encrypted disks. They won't be immediately available after a reboot. The process needs time to start, unlock, and then mount the encrypted drives before being available for other processes to access. This process may take more than a minute after reboot depending on the system characteristics.

Here is an example of the commands used to mount the data disks and create the necessary /etc/fstab entries:

Bash

```
sudo UUID0="$(blkid -s UUID -o value /dev/sda1)"
sudo UUID1="$(blkid -s UUID -o value /dev/sda2)"
sudo mkdir /data0
sudo mkdir /data1
sudo echo "UUID=$UUID0 /data0 ext4 defaults,nofail 0 0" >>/etc/fstab
sudo echo "UUID=$UUID1 /data1 ext4 defaults,nofail 0 0" >>/etc/fstab
sudo mount -a
```

Networking requirements

To enable the Azure Disk Encryption feature, the Linux VMs must meet the following network endpoint configuration requirements:

- The Linux VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see Azure Key Vault behind a firewall.

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see Creating and configuring a key vault for Azure Disk Encryption.

Terminology

The following table defines some of the common terms used in Azure disk encryption documentation:

Expand table

Terminology	Definition
Azure Key Vault	Key Vault is a cryptographic, key management service that's based on Federal Information Processing Standards (FIPS) validated hardware security modules. These standards help to safeguard your cryptographic keys and sensitive secrets. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption.
Azure CLI	The Azure CLI is optimized for managing and administering Azure resources from the command line.
DM-Crypt	DM-Crypt is the Linux-based, transparent disk-encryption subsystem that's used to enable disk encryption on Linux VMs.
Key encryption key (KEK)	The asymmetric key (RSA 2048) that you can use to protect or wrap the secret. You can provide a hardware security module (HSM)-protected key or software-protected key. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption.

Terminology	Definition
PowerShell cmdlets	For more information, see Azure PowerShell cmdlets.

Next steps

- Quickstart - Create and encrypt a Linux VM with Azure CLI
- Quickstart - Create and encrypt a Linux VM with Azure PowerShell
- Azure Disk Encryption scenarios on Linux VMs
- Migrate from Azure Disk Encryption to server-side encryption
- Azure Disk Encryption prerequisites CLI script
- Azure Disk Encryption prerequisites PowerShell script
- Creating and configuring a key vault for Azure Disk Encryption

---

Last updated on 02/20/2026

Quickstart: Create and encrypt a Linux VM with the Azure CLI

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This quickstart shows you how to use the Azure CLI to create and encrypt a Linux virtual machine (VM).

If you don't have an Azure subscription, create a free account before you begin.

If you choose to install and use the Azure CLI locally, this quickstart requires that you are running the Azure CLI version 2.0.30 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Create a resource group

Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

Azure CLI

```
az group create --name "myResourceGroup" --location "eastus"
```

Create a virtual machine

Create a VM with az vm create. The following example creates a VM named *myVM*.

Azure CLI

```
az vm create \
    --resource-group "myResourceGroup" \
    --name "myVM" \
    --image "Canonical:UbuntuServer:20.04-LTS:latest" \
    --size "Standard_D2S_V3"\
    --generate-ssh-keys
```

> **Note**
>
> Any ADE supported Linux image version could be used instead of an Ubuntu VM. Replace `Canonical:UbuntuServer:20.04-LTS:latest` accordingly.

It takes a few minutes to create the VM and supporting resources. The following example output shows the VM create operation was successful.

JSON


```
{
  "fqdns": "",
  "id":
"/subscriptions/<guid>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "eastus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.174.34.95",
  "resourceGroup": "myResourceGroup"
}
```


Create a Key Vault configured for encryption keys

Azure disk encryption stores its encryption key in an Azure Key Vault. Create a Key Vault with az keyvault create. To enable the Key Vault to store encryption keys, use the `--enabled-for-disk-encryption` parameter.

> **Important**
>
> Every key vault must have a name that is unique across Azure. Replace <your-unique-keyvault-name> with the name you choose.

Azure CLI

```
az keyvault create --name "<your-unique-keyvault-name>" --resource-group
"myResourceGroup" --location "eastus" --enabled-for-disk-encryption
```

Encrypt the virtual machine

Encrypt your VM with az vm encryption, providing your unique Key Vault name to the `--disk-encryption-keyvault parameter.

Azure CLI

```
az vm encryption enable -g "MyResourceGroup" --name "myVM" --disk-encryption-
keyvault "<your-unique-keyvault-name>"
```

After a moment the process returns, "The encryption request was accepted. Use 'show' command to monitor the progress.". The "show" command is az vm show.

Azure CLI

```
az vm encryption show --name "myVM" -g "MyResourceGroup"
```

When encryption is enabled, you will see "EnableEncryption" in the returned output:

Output

```
"EncryptionOperation": "EnableEncryption"
```

Clean up resources

When no longer needed, you can use the az group delete command to remove the resource group, VM, and Key Vault.

Azure CLI

```
az group delete --name "myResourceGroup"
```

Next steps

In this quickstart, you created a virtual machine, created a Key Vault that was enabled for encryption keys, and encrypted the VM. Advance to the next article to learn more about more Azure Disk Encryption for Linux VMs.

Azure Disk Encryption overview

Last updated on 10/10/2025

Quickstart: Create and encrypt a Linux VM in Azure with Azure PowerShell

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The Azure PowerShell module is used to create and manage Azure resources from the PowerShell command line or in scripts. This quickstart shows you how to use the Azure PowerShell module to create a Linux virtual machine (VM), create a Key Vault for the storage of encryption keys, and encrypt the VM. This quickstart uses the Ubuntu 16.04 LTS marketplace image from Canonical and a VM Standard_D2S_V3 size. However, any ADE supported Linux image version could be used instead of an Ubuntu VM.

If you don't have an Azure subscription, create a free account before you begin.

Create a resource group

Create an Azure resource group with New-AzResourceGroup. A resource group is a logical container into which Azure resources are deployed and managed:

Azure PowerShell

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create a virtual machine

Create an Azure virtual machine with New-AzVM, passing it to the VM configuration object you created in the previous step.

PowerShell

```
$cred = Get-Credential

New-AzVM -Name MyVm -Credential $cred -ResourceGroupName MyResourceGroup -Image
Canonical:UbuntuServer:18.04-LTS:latest -Size Standard_D2S_V3
```

It takes a few minutes for your VM to be deployed.

Create a Key Vault configured for encryption keys

Azure disk encryption stores its encryption key in an Azure Key Vault. Create a Key Vault with New-AzKeyvault. To enable the Key Vault to store encryption keys, use the -EnabledForDiskEncryption parameter.

> **Important**
>
> Every key vault must have a name that is unique across Azure. In the following examples, replace <your-unique-keyvault-name> with the name you choose.

Azure PowerShell

```
New-AzKeyvault -name "<your-unique-keyvault-name>" -ResourceGroupName
"myResourceGroup" -Location EastUS -EnabledForDiskEncryption
```

Encrypt the virtual machine

Encrypt your VM with Set-AzVmDiskEncryptionExtension.

Set-AzVmDiskEncryptionExtension requires some values from your Key Vault object. You can obtain these values by passing the unique name of your key vault to Get-AzKeyvault.

Azure PowerShell

```
$KeyVault = Get-AzKeyVault -VaultName "<your-unique-keyvault-name>" -
ResourceGroupName "MyResourceGroup"

Set-AzVMDiskEncryptionExtension -ResourceGroupName MyResourceGroup -VMName "MyVM" -
DiskEncryptionKeyVaultUrl $KeyVault.VaultUri -DiskEncryptionKeyVaultId
$KeyVault.ResourceId -SkipVmBackup -VolumeType All
```

After a few minutes the process returns:

Output

```
RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

You can verify the encryption process by running Get-AzVmDiskEncryptionStatus.

Azure PowerShell

```
Get-AzVmDiskEncryptionStatus -VMName MyVM -ResourceGroupName MyResourceGroup
```

When encryption is enabled, you see this information in the returned output:

Output

```
OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : NotMounted
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : OS disk encryption started
```

Clean up resources

When no longer needed, you can use the Remove-AzResourceGroup cmdlet to remove the resource group, VM, and all related resources:

Azure PowerShell

```
Remove-AzResourceGroup -Name "myResourceGroup"
```

Next steps

In this quickstart you created a virtual machine, created a Key Vault enabled for encryption keys, and encrypted the VM. Advance to the next article to learn more about Azure Disk Encryption for Linux VMs.

Azure Disk Encryption overview

Last updated on 10/10/2025

Quickstart: Create and encrypt a virtual machine with the Azure portal

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure virtual machines (VMs) can be created through the Azure portal. The Azure portal is a browser-based user interface to create VMs and their associated resources. In this quickstart you'll use the Azure portal to deploy a Linux virtual machine (VM) running Ubuntu 18.04 LTS, create a key vault for the storage of encryption keys, and encrypt the VM. However, any ADE supported Linux image version could be used instead of an Ubuntu VM.

If you don't have an Azure subscription, create a free account before you begin.

Sign in to Azure

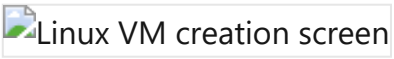
Sign in to the Azure portal.

Create a virtual machine

1. Choose **Create a resource** in the upper left corner of the Azure portal.
2. In the New page, under Popular, select **Ubuntu Server 18.04 LTS**.
3. In the Basics tab, under Project details, verify sure the correct subscription is selected.
4. For "Resource Group", select **Create new**. Enter *myResourceGroup* as the name and select **Ok**.
5. For **Virtual machine name**, enter *MyVM*.
6. For **Region**, select *(US) East US*.

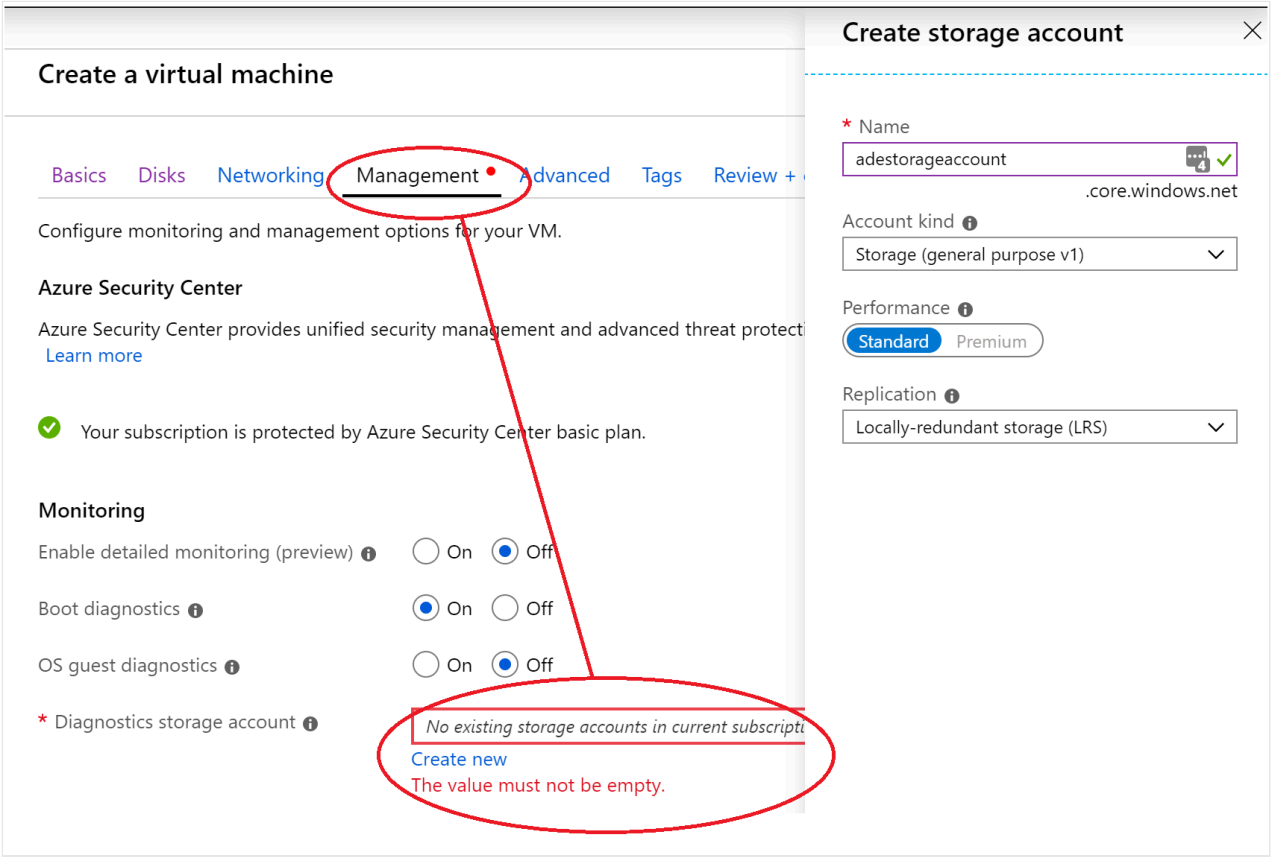
7. Make sure the **Size** is *Standard D2s v3*.

8. Under **Administrator account**, select *Password* as the **Authentication type**. Enter a user name and a password.



> ⚠ **Warning**
>
> The "Disks" tab features an "Encryption Type" field under **Disk options**. This field is used to specify encryption options for Managed Disks + CMK, **not** for Azure Disk Encryption.
>
> To avoid confusion, we suggest you skip the *Disks* tab entirely while completing this tutorial.

9. Select the "Monitoring" tab and verify that you have a Diagnostics Storage Account. If you have no storage accounts, select *Create New*, name your storage account *myStorageAccount*, and select "Ok"



10. Select "Review + Create".

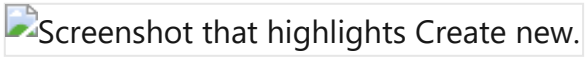
11. On the **Create a virtual machine** page, you can see the details about the VM you are about to create. When you are ready, select **Create**.

It takes a few minutes for your VM to be deployed. When the deployment is finished, move on to the next section.

Encrypt the virtual machine

1. When the VM deployment is complete, select **Go to resource**.
2. On the left-hand sidebar, select **Disks**.
3. On the top bar, select **Additional Settings** .
4. Under **Encryption settings** > **Disks to encrypt**, select **OS and data disks**.

   
   Screenshot that highlights OS and data disks.
5. Under **Encryption settings**, choose **Select a key vault and key for encryption**.
6. On the **Select key from Azure Key Vault** screen, select **Create New**.

   
   Screenshot that highlights Create new.
7. To the left of **Key vault and key**, select **Click to select a key**.
8. On the **Select key from Azure Key Vault**, under the **Key Vault** field, select **Create new**.
9. On the **Create key vault** screen, ensure that the Resource Group is *myResourceGroup*, and give your key vault a name. Every key vault across Azure must have a unique name.
10. On the **Access Policies** tab, check the **Azure Disk Encryption for volume encryption** box.

    
    disks and encryption selection
11. Select **Review + create**.
12. After the key vault passes validation, select **Create**. This returns you to the **Select key from Azure Key Vault** screen.
13. Leave the **Key** field blank and choose **Select**.
14. At the top of the encryption screen, click **Save**. A popup will warn you that the VM will reboot. Select **Yes**.

Clean up resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the virtual machine, select Delete, then

confirm the name of the resource group to delete.

Next steps

In this quickstart, you created a Key Vault that was enabled for encryption keys, created a virtual machine, and enabled the virtual machine for encryption.

Azure Disk Encryption overview

Last updated on 11/20/2025

Azure Disk Encryption scenarios on Linux VMs

Summarize this article for me

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure Disk Encryption for Linux virtual machines (VMs) uses the DM-Crypt feature of Linux to provide full disk encryption of the OS disk and data disks. Additionally, it provides encryption of the temporary disk when using the EncryptFormatAll feature.

Azure Disk Encryption is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets. For an overview of the service, see Azure Disk Encryption for Linux VMs.

Prerequisites

You can only apply disk encryption to virtual machines of supported VM sizes and operating systems. You must also meet the following prerequisites:

- Requirements for VMs
- Networking requirements
- Encryption key storage requirements

In all cases, you should take a snapshot and/or create a backup before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the Set-AzVMDiskEncryptionExtension cmdlet to encrypt managed disks by specifying the -skipVmBackup parameter. For more information about how to back up and restore encrypted VMs, see the Azure Backup article.

Restrictions

If you previously used Azure Disk Encryption with Microsoft Entra ID to encrypt a virtual machine, you must continue use this option to encrypt your virtual machine. See Azure Disk Encryption with Microsoft Entra ID (previous release) for details.

When encrypting Linux OS volumes, the VM should be considered unavailable. We strongly recommend to avoid SSH logins while the encryption is in progress to avoid issues blocking any open files that need to be accessed during the encryption process. To check progress, use the Get-AzVMDiskEncryptionStatus PowerShell cmdlet or the vm encryption show CLI command. This process can be expected to take a few hours for a 30GB OS volume, plus extra time for encrypting data volumes. Data volume encryption time is proportional to the size and quantity of the data volumes unless the encrypt format all option is used.

Disabling encryption on Linux VMs is only supported for data volumes. Disabling encryption is not supported on data or OS volumes if the OS volume is encrypted.

Azure Disk Encryption does not work for the following Linux scenarios, features, and technology:

- Encrypting basic tier VM or VMs created through the classic VM creation method.
- Encrypting v6 series VMs with temporary disks (Ddsv6, Dldsv6, Edsv6, Dadsv6, Daldsv6, Eadsv6, Dpdsv6, Dpldsv6, Epdsv6, or Endsv6). For more information, see the individual pages for each of these VM sizes listed on Sizes for virtual machines in Azure
- Encrypting V7 series and newer VM sizes.
- Disabling encryption on an OS drive or data drive of a Linux VM when the OS drive is encrypted.
- Encrypting the OS drive for Linux Virtual Machine Scale Sets.
- Encrypting custom images on Linux VMs.
- Integration with an on-premises key management system.
- Azure Files (shared file system).
- Network File System (NFS).
- Dynamic volumes.
- Ephemeral OS disks.
- Encryption of shared/distributed file systems like (but not limited to): DFS, GFS, DRDB, and CephFS.
- Moving an encrypted VM to another subscription or region.
- Creating an image or snapshot of an encrypted VM and using it to deploy more VMs.
- Kernel Crash Dump (kdump).
- Oracle ACFS (ASM Cluster File System).
- NVMe disks such as those on High performance computing VM sizes or Storage optimized VM sizes.

- A VM with "nested mount points"; that is, multiple mount points in a single path (such as "/1stmountpoint/data/2ndmountpoint").
- A VM with a data drive mounted on top of an OS folder.
- A VM on which a root (OS disk) logical volume is extended using a data disk.
- Resizing of the OS disk.
- 'L' family storage optimized VM size series.
- M-series VMs with Write Accelerator disks.
- Applying ADE to a VM that has disks encrypted with Encryption at Host or server-side encryption with customer-managed keys (SSE + CMK). Applying SSE + CMK to a data disk or adding a data disk with SSE + CMK configured to a VM encrypted with ADE is an unsupported scenario as well.
- Migrating a VM that is encrypted with ADE, or was **ever** encrypted with ADE, to Encryption at Host or server-side encryption with customer-managed keys.
- Encrypting VMs in failover clusters.
- Encryption of Azure Ultra Disks.
- Encryption of Premium SSD v2 disks.
- Encryption of VMs in subscriptions that have the Secrets should have the specified maximum validity period policy enabled with the DENY effect.

Install tools and connect to Azure

Azure Disk Encryption can be enabled and managed through the Azure CLI and Azure PowerShell. To do so, you must install the tools locally and connect to your Azure subscription.

Azure CLI

The Azure CLI 2.0 is a command-line tool for managing Azure resources. The CLI is designed to flexibly query data, support long-running operations as non-blocking processes, and make scripting easy. You can install it locally by following the steps in Install the Azure CLI.

To Sign in to your Azure account with the Azure CLI, use the az login command.

Azure CLI

```
az login
```

If you would like to select a tenant to sign in under, use:

Azure CLI

```
az login --tenant <tenant>
```

If you have multiple subscriptions and want to specify a specific one, get your subscription list with az account list and specify with az account set.

Azure CLI

```
az account list
az account set --subscription "<subscription name or ID>"
```

For more information, see Get started with Azure CLI 2.0.

Enable encryption on an existing or running Linux VM

In this scenario, you can enable encryption by using the Resource Manager template, PowerShell cmdlets, or CLI commands. If you need schema information for the virtual machine extension, see the Azure Disk Encryption for Linux extension article.

> **ⓘ Important**
>
> It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and before enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or through Azure Backup. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzVMDiskEncryptionExtension command fails against managed disk based VMs until a backup is made and this parameter is specified.
>
> Encrypting or disabling encryption may cause the VM to reboot.

To disable the encryption, see Disable encryption and remove the encryption extension.

Using Azure CLI

You can enable disk encryption on your encrypted VHD by installing and using the Azure CLI command-line tool. You can use it in your browser with Azure Cloud Shell, or you can install it on your local machine and use it in any PowerShell session. To enable encryption on existing or running Linux VMs in Azure, use the following CLI commands:

Use the az vm encryption enable command to enable encryption on a running virtual machine in Azure.

- **Encrypt a running VM:**

  Azure CLI

  ```
  az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
  ```

- **Encrypt a running VM using KEK:**

  Azure CLI

  ```
  az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault  "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
  ```

  > **Note**
  >
  > The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
  > The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of a VM, use the az vm encryption show command.

  Azure CLI

  ```
  az vm encryption show --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup"
  ```

To disable the encryption, see Disable encryption and remove the encryption extension.

Use EncryptFormatAll feature for data disks on Linux VMs

The **EncryptFormatAll** parameter reduces the time for Linux data disks to be encrypted. Partitions meeting certain criteria are formatted, along with their current file systems, then remounted back to where they were before command execution. If you wish to exclude a data disk that meets the criteria, you can unmount it before running the command.

After running this command, any drives that were mounted previously are formatted, and the encryption layer will be started on top of the now empty drive. When this option is selected, the temporary disk attached to the VM is also encrypted. If the temporary disk is reset, Azure Disk Encryption will reformat and re-encrypt the VM's temporary disk at the next opportunity. Once the resource disk is encrypted, the Microsoft Azure Linux Agent is not able to manage the resource disk and enable the swap file, but you may manually configure the swap file.

> ⚠ **Warning**
>
> EncryptFormatAll shouldn't be used when there is needed data on a VM's data volumes. You may exclude disks from encryption by unmounting them. You should first try out the EncryptFormatAll first on a test VM, understand the feature parameter and its implication before trying it on the production VM. The EncryptFormatAll option formats the data disk and all the data on it will be lost. Before proceeding, verify that disks you wish to exclude are properly unmounted.
>
> If you're setting this parameter while updating encryption settings, it might lead to a reboot before the actual encryption. In this case, you should remove the disk you don't want formatted from the fstab file. Similarly, you should add the partition you want encrypt-formatted to the fstab file before initiating the encryption operation.

EncryptFormatAll criteria

The parameter goes through all partitions and encrypts them as long as they meet **all** of the criteria below:

- Is not a root/OS/boot partition
- Is not already encrypted
- Is not a BEK volume
- Is not a RAID volume
- Is not an LVM volume
- Is mounted

Encrypt the disks that compose the RAID or LVM volume rather than the RAID or LVM volume.

Use the EncryptFormatAll parameter with Azure CLI

Use the az vm encryption enable command to enable encryption on a running virtual machine in Azure.

- **Encrypt a running VM using EncryptFormatAll:**

  Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type "data" --encrypt-format-all
```

Use the EncryptFormatAll parameter with Logical Volume Manager (LVM)

We recommend an LVM-on-crypt setup. For detailed instructions about the LVM on crypt configuration, see Configure LVM and RAID on ADE encrypted devices.

New VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using PowerShell cmdlets or CLI commands.

Use the instructions in the Azure Disk encryption same scripts for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- Prepare a pre-encrypted Linux VHD

> ⓘ **Important**
>
> It is mandatory to snapshot and/or backup a managed disk based VM instance outside of, and before enabling Azure Disk Encryption. A snapshot of the managed disk can be taken from the portal, or **Azure Backup** can be used. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. Once a backup is made, the Set-AzVMDiskEncryptionExtension cmdlet can be used to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzVMDiskEncryptionExtension

command fails against managed disk based VMs until a backup is made and this parameter is specified.

Encrypting or disabling encryption may cause the VM to reboot.

Use Azure PowerShell to encrypt VMs with pre-encrypted VHDs

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet Set-AzVMOSDisk. This example gives you some common parameters.

Azure PowerShell

```
$VirtualMachine = New-AzVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri
"os.vhd" Caching ReadWrite -Linux -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa
" -DiskEncryptionKeyVaultId "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myresourcegroup/providers/Microsoft.KeyVault/vaults/myt
estvault"
New-AzVM -VM $VirtualMachine -ResourceGroupName "MyVirtualMachineResourceGroup"
```

Enable encryption on a newly added data disk

You can add a new data disk using az vm disk attach, or through the Azure portal. Before you can encrypt, you need to mount the newly attached data disk first. You must request encryption of the data drive since the drive is not usable while encryption is in progress.

Using Azure CLI

If the VM was previously encrypted with "All", then the --volume-type parameter should remain "All". All includes both OS and data disks. If the VM was previously encrypted with a volume type of "OS", then the --volume-type parameter should be changed to "All" so that both the OS and the new data disk are included. If the VM was encrypted with only the volume type of "Data", then it can remain "Data" as demonstrated below. Adding and attaching a new data disk to a VM is not sufficient preparation for encryption. The newly attached disk must also be formatted and properly mounted within the VM before enabling encryption. On Linux, the disk must be mounted in /etc/fstab with a persistent block device name.

In contrast to PowerShell syntax, the CLI does not require the user to provide a unique sequence version when enabling encryption. The CLI automatically generates and uses its

own unique sequence version value.

- **Encrypt data volumes of a running VM:**

  Azure CLI

  ```
  az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type "Data"
  ```

- **Encrypt data volumes of a running VM using KEK:**

  Azure CLI

  ```
  az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --disk-encryption-keyvault  "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "Data"
  ```

Disable encryption and remove the encryption extension

You can disable the Azure disk encryption, and you can remove the Azure disk encryption extension. Disabling and removing are two distinct operations.

To remove ADE, it is recommended that you first disable encryption, and then remove the extension. If you remove the encryption extension without disabling ADE, the disks remain encrypted. If you disable encryption **after** removing the extension, the extension will be reinstalled (to perform the decrypt operation) and will need to be removed a second time.

> ⚠ **Warning**
>
> You can **not** disable encryption if the OS disk is encrypted. (OS disks are encrypted when the original encryption operation specifies volumeType=ALL or volumeType=OS.)
>
> Disabling encryption works only when data disks are encrypted but the OS disk is not.

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template. Disabling encryption does **not** remove the extension (see Remove the encryption

extension).

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the Disable-AzVMDiskEncryption cmdlet.

  Azure PowerShell

  ```
  Disable-AzVMDiskEncryption -ResourceGroupName "MyVirtualMachineResourceGroup"
  -VMName "MySecureVM" -VolumeType "data"
  ```

- **Disable encryption with the Azure CLI:** To disable encryption, use the az vm encryption disable command.

  Azure CLI

  ```
  az vm encryption disable --name "MySecureVM" --resource-group
  "MyVirtualMachineResourceGroup" --volume-type "data"
  ```

- **Disable encryption with a Resource Manager template:**

  1. Click **Deploy to Azure** from the Disable disk encryption on running Linux VM template.
  2. Select the subscription, resource group, location, VM, volume type, legal terms, and agreement.
  3. Click **Purchase** to disable disk encryption on a running Linux VM.

> ⚠ **Warning**
>
> Once decryption starts, it is advisable not to interfere with the process.
>
> To check the progress of decryption, use the **Get-AzVMDiskEncryptionStatus** PowerShell cmdlet or the **az vm encryption show** CLI command, once the decryption process is completed, you can proceed to remove the encryption extension

Remove the encryption extension

If you want to decrypt your disks and remove the encryption extension, you must disable encryption **before** removing the extension; see disable encryption.

You can remove the encryption extension using Azure PowerShell or the Azure CLI.

- **Disable disk encryption with Azure PowerShell:** To remove the encryption, use the Remove-AzVMDiskEncryptionExtension cmdlet.

Azure PowerShell

```
Remove-AzVMDiskEncryptionExtension -ResourceGroupName
"MyVirtualMachineResourceGroup" -VMName "MySecureVM"
```

- **Disable encryption with the Azure CLI:** To remove encryption, use the az vm extension delete command.

Azure CLI

```
az vm extension delete -g "MyVirtualMachineResourceGroup" --vm-name
"MySecureVM" -n "AzureDiskEncryptionForLinux"
```

Next steps

- Azure Disk Encryption overview
- Azure Disk Encryption sample scripts
- Azure Disk Encryption troubleshooting

Last updated on 02/20/2026

Creating and configuring a key vault for Azure Disk Encryption

09/23/2025

> **ⓘ Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure Disk Encryption uses Azure Key Vault to control and manage disk encryption keys and secrets. For more information about key vaults, see Get started with Azure Key Vault and Secure your key vault.

> **⚠ Warning**
>
> - If you have previously used Azure Disk Encryption with Microsoft Entra ID to encrypt a VM, you must continue to use this option to encrypt your VM. See Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release) for details.

Creating and configuring a key vault for use with Azure Disk Encryption involves three steps:

1. Creating a resource group, if needed.
2. Creating a key vault.
3. Setting key vault advanced access policies.

These steps are illustrated in the following quickstarts:

- Create and encrypt a Linux VM with Azure CLI
- Create and encrypt a Linux VM with Azure PowerShell

You may also, if you wish, generate or import a key encryption key (KEK).

> **ⓘ Note**

The steps in this article are automated in the Azure Disk Encryption prerequisites CLI script and Azure Disk Encryption prerequisites PowerShell script.

Install tools and connect to Azure

The steps in this article can be completed with the Azure CLI, the Azure PowerShell Az PowerShell module module, or the Azure portal.

While the portal is accessible through your browser, Azure CLI and Azure PowerShell require local installation; see Azure Disk Encryption for Linux: Install tools for details.

Connect to your Azure account

Before using the Azure CLI or Azure PowerShell, you must first connect to your Azure subscription. You do so by Signing in with Azure CLI, Signing in with Azure PowerShell, or supplying your credentials to the Azure portal when prompted.

Azure CLI

```
az login
```

Azure PowerShell

```
Connect-AzAccount
```

Create a resource group

If you already have a resource group, you can skip to Create a key vault.

A resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group using the az group create Azure CLI command, the New-AzResourceGroup Azure PowerShell command, or from the Azure portal.

Azure portal

Azure CLI

Azure CLI

```
az group create --name "myResourceGroup" --location eastus
```

Azure PowerShell

Azure PowerShell

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create a key vault

If you already have a key vault, you can skip to Set key vault advanced access policies.

Create a key vault using the az keyvault create Azure CLI command, the New-AzKeyvault Azure PowerShell command, the Azure portal, or a Resource Manager template.

> ⚠ **Warning**
>
> To ensure that encryption secrets don't cross regional boundaries, you must create and use a key vault that's in the **same region and tenant** as the VMs to be encrypted.

Each Key Vault must have a unique name. Replace <your-unique-keyvault-name> with the name of your key vault in the following examples.

Azure CLI

When creating a key vault by using the Azure CLI, add the "--enabled-for-disk-encryption" flag.

Azure CLI

```
az keyvault create --name "<your-unique-keyvault-name>" --resource-group
"myResourceGroup" --location "eastus" --enabled-for-disk-encryption
```

Azure PowerShell

When creating a key vault using Azure PowerShell, add the "-EnabledForDiskEncryption" flag.

Azure PowerShell

```
New-AzKeyvault -name "<your-unique-keyvault-name>" -ResourceGroupName
"myResourceGroup" -Location "eastus" -EnabledForDiskEncryption
```

Resource Manager template

You can also create a key vault by using the Resource Manager template.

1. On the Azure Quickstart Template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then click **Purchase**.

Set key vault advanced access policies

> **Important**
>
> Newly-created key vaults have soft-delete on by default. If you are using a pre-existing key vault, you **must** enable soft-delete. See Azure Key Vault soft-delete overview.

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the VM for booting and decrypting the volumes.

If you didn't enable your key vault for disk encryption, deployment, or template deployment at the time of creation (as demonstrated in the previous step), you must update its advanced access policies.

Azure CLI

Use az keyvault update to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

  Azure CLI

  ```
  az keyvault update --name "<your-unique-keyvault-name>" --resource-group
  "MyResourceGroup" --enabled-for-disk-encryption "true"
  ```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

Azure CLI

```
az keyvault update --name "<your-unique-keyvault-name>" --resource-group
"MyResourceGroup" --enabled-for-deployment "true"
```

- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

Azure CLI

```
az keyvault update --name "<your-unique-keyvault-name>" --resource-group
"MyResourceGroup" --enabled-for-template-deployment "true"
```

Azure PowerShell

Use the key vault PowerShell cmdlet Set-AzKeyVaultAccessPolicy to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

Azure PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName "<your-unique-keyvault-name>" -
ResourceGroupName "MyResourceGroup" -EnabledForDiskEncryption
```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft.Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in resource creation, for example when creating a virtual machine.

Azure PowerShell

```
 Set-AzKeyVaultAccessPolicy -VaultName "<your-unique-keyvault-name>" -
ResourceGroupName "MyResourceGroup" -EnabledForDeployment
```

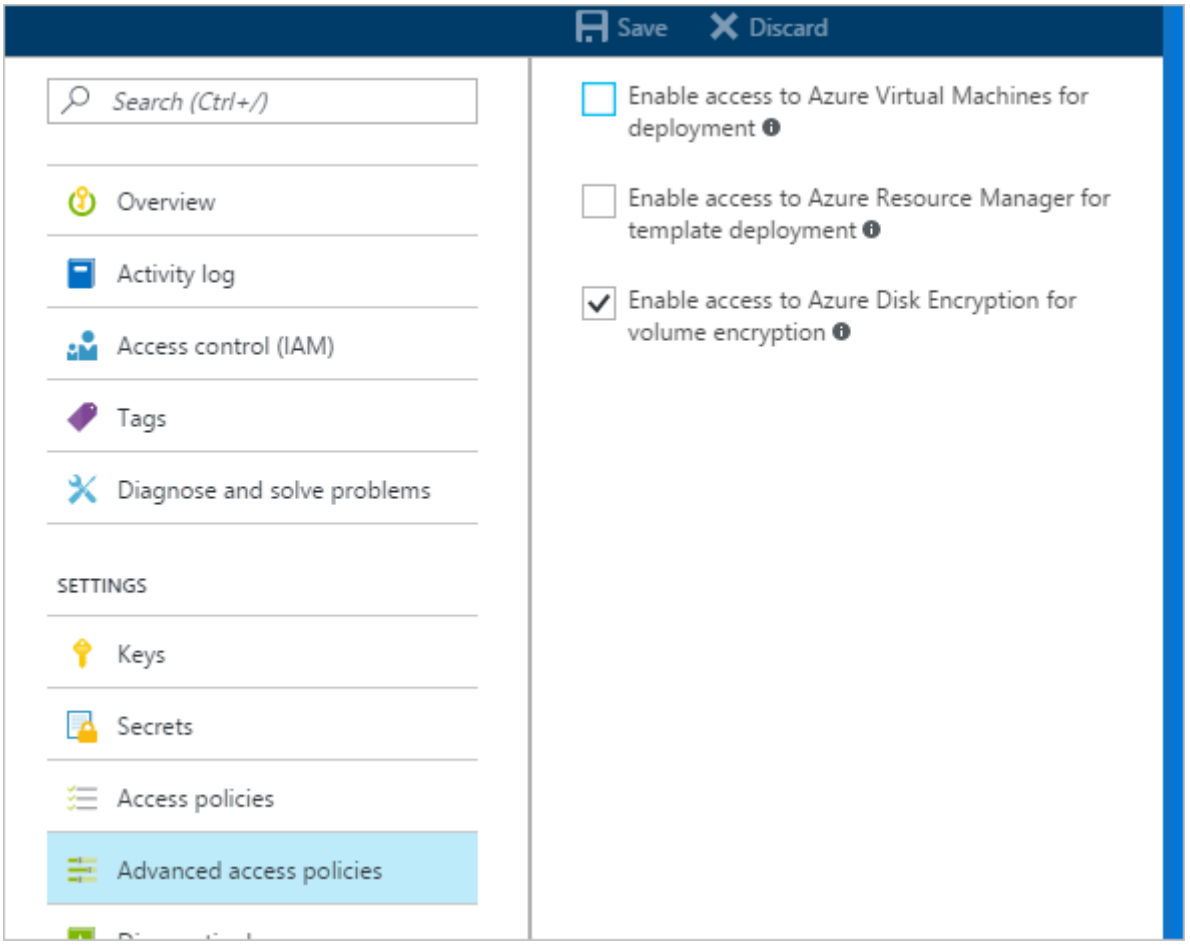
- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

Azure PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName "<your-unique-keyvault-name>" -
ResourceGroupName "MyResourceGroup" -EnabledForTemplateDeployment
```

Azure portal

1. Select your key vault and go to **Access Policies**.
2. Under "Enable Access to", select the box labeled **Azure Disk Encryption for volume encryption**.
3. Select **Azure Virtual Machines for deployment** and/or **Azure Resource Manager for template deployment**, if needed.
4. Click **Save**.



Azure Disk Encryption and auto-rotation

Although Azure Key Vault now has key auto-rotation, it isn't currently compatible with Azure Disk Encryption. Specifically, Azure Disk Encryption will continue to use the original encryption key, even after it has been auto-rotated.

Rotating an encryption key won't break Azure Disk Encryption, but disabling the "old" encryption key (in other words, the key Azure Disk Encryption is still using) will.

Set up a key encryption key (KEK)

> **Important**
>
> The account running to enable disk encryption over the key vault must have "reader" permissions.

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

You can generate a new KEK by using the Azure CLI az keyvault key create command, the Azure PowerShell Add-AzKeyVaultKey cmdlet, or the Azure portal. You must generate an RSA key type; Azure Disk Encryption doesn't currently support using Elliptic Curve keys.

You can instead import a KEK from your on-premises key management HSM. For more information, see Key Vault Documentation.

Your key vault KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:

- Example of a valid secret URL: `https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`
- Example of a valid KEK URL: `https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxxxxxxxxxx`

Azure CLI

Use the Azure CLI az keyvault key create command to generate a new KEK and store it in your key vault.

Azure CLI

```
az keyvault key create --name "myKEK" --vault-name "<your-unique-keyvault-name>" --kty RSA --size 4096
```

You may instead import a private key by using the Azure CLI az keyvault key import command:

In either case, you supply the name of your KEK to the Azure CLI az vm encryption enable --key-encryption-key parameter.

Azure CLI

```
az vm encryption enable -g "MyResourceGroup" --name "myVM" --disk-encryption-keyvault "<your-unique-keyvault-name>" --key-encryption-key "myKEK"
```

Azure PowerShell

Use the Azure PowerShell Add-AzKeyVaultKey cmdlet to generate a new KEK and store it in your key vault.

PowerShell

```
Add-AzKeyVaultKey -Name "myKEK" -VaultName "<your-unique-keyvault-name>" -Destination "HSM" -Size 4096
```

You may instead import a private key using the Azure PowerShell az keyvault key import command.

In either case, you will supply the ID of your KEK key Vault and the URL of your KEK to the Azure PowerShell Set-AzVMDiskEncryptionExtension -KeyEncryptionKeyVaultId and -KeyEncryptionKeyUrl parameters. This example assumes that you are using the same key vault for both the disk encryption key and the KEK.

PowerShell

```
$KeyVault = Get-AzKeyVault -VaultName "<your-unique-keyvault-name>" -ResourceGroupName "myResourceGroup"
$KEK = Get-AzKeyVaultKey -VaultName "<your-unique-keyvault-name>" -Name "myKEK"

Set-AzVMDiskEncryptionExtension -ResourceGroupName MyResourceGroup -VMName "MyVM" -DiskEncryptionKeyVaultUrl $KeyVault.VaultUri -DiskEncryptionKeyVaultId $KeyVault.ResourceId -KeyEncryptionKeyVaultId $KeyVault.ResourceId -KeyEncryptionKeyUrl $KEK.Id -SkipVmBackup -VolumeType All
```

Next steps

- Azure Disk Encryption prerequisites CLI script
- Azure Disk Encryption prerequisites PowerShell script
- Learn Azure Disk Encryption scenarios on Linux VMs
- Learn how to troubleshoot Azure Disk Encryption
- Read the Azure Disk Encryption sample scripts

Azure Disk Encryption sample scripts for Linux VMs

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

> **Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Please consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article provides sample scripts for preparing pre-encrypted VHDs and other tasks.

> **Note**
>
> All scripts refer to the latest, non-AAD version of ADE, except where noted.

Sample PowerShell scripts for Azure Disk Encryption

- **List all encrypted VMs in your subscription**

  You can find all ADE-encrypted VMs and the extension version, in all resource groups present in a subscription, using this PowerShell script.

  Alternatively, these cmdlets will show all ADE-encrypted VMs (but not the extension version):

  ```
  Azure PowerShell
  ```

```
$osVolEncrypted = {(Get-AzVMDiskEncryptionStatus -ResourceGroupName
$_.ResourceGroupName -VMName $_.Name).OsVolumeEncrypted}
$dataVolEncrypted= {(Get-AzVMDiskEncryptionStatus -ResourceGroupName
$_.ResourceGroupName -VMName $_.Name).DataVolumesEncrypted}
Get-AzVm | Format-Table @{Label="MachineName"; Expression={$_.Name}},
@{Label="OsVolumeEncrypted"; Expression=$osVolEncrypted},
@{Label="DataVolumesEncrypted"; Expression=$dataVolEncrypted}
```

- **List all encrypted VMSS instances in your subscription**

  You can find all ADE-encrypted VMSS instances and the extension version, in all resource groups present in a subscription, using this PowerShell script.

- **List all disk encryption secrets used for encrypting VMs in a key vault**

  Azure PowerShell

```
Get-AzKeyVaultSecret -VaultName $KeyVaultName | where
{$_.Tags.ContainsKey('DiskEncryptionKeyFileName')} | format-table
@{Label="MachineName"; Expression={$_.Tags['MachineName']}},
@{Label="VolumeLetter"; Expression={$_.Tags['VolumeLetter']}},
@{Label="EncryptionKeyURL"; Expression={$_.Id}}
```

Using the Azure Disk Encryption prerequisites PowerShell script

If you're already familiar with the prerequisites for Azure Disk Encryption, you can use the Azure Disk Encryption prerequisites PowerShell script. For an example of using this PowerShell script, see the Encrypt a VM Quickstart. You can remove the comments from a section of the script, starting at line 211, to encrypt all disks for existing VMs in an existing resource group.

The following table shows which parameters can be used in the PowerShell script:

Expand table

Parameter	Description	Mandatory?
$resourceGroupName	Name of the resource group to which the KeyVault belongs to. A new resource group with this name will be created if one doesn't exist.	True
$keyVaultName	Name of the KeyVault in which encryption keys are to be placed. A new vault with this name will be created if one doesn't exist.	True

Parameter	Description	Mandatory?
$location	Location of the KeyVault. Make sure the KeyVault and VMs to be encrypted are in the same location. Get a location list with `Get-AzLocation`.	True
$subscriptionId	Identifier of the Azure subscription to be used. You can get your Subscription ID with `Get-AzSubscription`.	True
$aadAppName	Name of the Microsoft Entra application that will be used to write secrets to KeyVault. A new application with this name will be created if one doesn't exist. If this app already exists, pass aadClientSecret parameter to the script.	False
$aadClientSecret	Client secret of the Microsoft Entra application that was created earlier.	False
$keyEncryptionKeyName	Name of optional key encryption key in KeyVault. A new key with this name will be created if one doesn't exist.	False

Encrypt or decrypt VMs without a Microsoft Entra app

- Enable disk encryption on an existing or running Linux VM
- Disable encryption on a running Linux VM
  - Disabling encryption is only allowed on Data volumes for Linux VMs.

Encrypt or decrypt VMs with a Microsoft Entra app (previous release)

- Enable disk encryption on an existing or running Linux VM
- Disable encryption on a running Linux VM
  - Disabling encryption is only allowed on Data volumes for Linux VMs.
- Create a new encrypted managed disk from a pre-encrypted VHD/storage blob
  - Creates a new encrypted managed disk provided a pre-encrypted VHD and its corresponding encryption settings

Encrypting an OS drive on a running Linux VM

Prerequisites for OS disk encryption

- The VM must be using a distribution compatible with OS disk encryption as listed in the Azure Disk Encryption supported operating systems
- The VM must be created from the Marketplace image in Azure Resource Manager.

- Azure VM with at least 4 GB of RAM (recommended size is 7 GB). See Memory requirements for further information.
- (For RHEL and CentOS) Disable SELinux. To disable SELinux, see "4.4.2. Disabling SELinux" in the SELinux User's and Administrator's Guide on the VM.
- After you disable SELinux, reboot the VM at least once.

Steps

1. Create a VM by using one of the distributions specified previously.

2. Configure the VM according to your needs. If you're going to encrypt all the (OS + data) drives, the data drives need to be specified and mountable from /etc/fstab.

   > **Note**
   >
   > Use UUID=... to specify data drives in /etc/fstab instead of specifying the block device name (for example, /dev/sdb1). During encryption, the order of drives changes on the VM. If your VM relies on a specific order of block devices, it will fail to mount them after encryption.

3. Sign out of the SSH sessions.

4. To encrypt the OS, specify volumeType as **All** or **OS** when you enable encryption.

   > **Note**
   >
   > All user-space processes that are not running as `systemd` services should be killed with a `SIGKILL`. Reboot the VM. When you enable OS disk encryption on a running VM, plan on VM downtime.

5. Periodically monitor the progress of encryption by using the instructions in the next section.

6. After Get-AzVmDiskEncryptionStatus shows "VMRestartPending", restart your VM either by signing in to it or by using the portal, PowerShell, or CLI.

   Azure PowerShell

   ```
   C:\> Get-AzVmDiskEncryptionStatus  -ResourceGroupName $ResourceGroupName -
   VMName $VMName
   -ExtensionName $ExtensionName
   ```

Output

```
OsVolumeEncrypted          : VMRestartPending
DataVolumesEncrypted       : NotMounted
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : OS disk successfully encrypted, reboot the VM
```

Before you reboot, we recommend that you save boot diagnostics of the VM.

Monitoring OS encryption progress

You can monitor OS encryption progress in three ways:

- Use the `Get-AzVmDiskEncryptionStatus` cmdlet and inspect the ProgressMessage field:

Azure PowerShell

```
Get-AzVMDiskEncryptionStatus -ResourceGroupName $_.ResourceGroupName -VMName
$_.Name
```

Output

```
OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : NotMounted
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : OS disk encryption started
```

After the VM reaches "OS disk encryption started", it takes about 40 to 50 minutes on a Premium-storage backed VM.

Because of issue #388 in WALinuxAgent, `OsVolumeEncrypted` and `DataVolumesEncrypted` show up as `Unknown` in some distributions. With WALinuxAgent version 2.1.5 and later, this issue is fixed automatically. If you see `Unknown` in the output, you can verify disk-encryption status by using the Azure Resource Explorer.

Go to Azure Resource Explorer, and then expand this hierarchy in the selection panel on left:

```
|-- subscriptions
   |-- [Your subscription]
        |-- resourceGroups
```

```
|-- [Your resource group]
     |-- providers
          |-- Microsoft.Compute
               |-- virtualMachines
                    |-- [Your virtual machine]
                         |-- InstanceView
```

In the InstanceView, scroll down to see the encryption status of your drives.



- Look at boot diagnostics. Messages from the ADE extension should be prefixed with `[AzureDiskEncryption]`.

- Sign in to the VM via SSH, and get the extension log from:

  /var/log/azure/Microsoft.Azure.Security.AzureDiskEncryptionForLinux

  We recommend that you don't sign-in to the VM while OS encryption is in progress. Copy the logs only when the other two methods have failed.

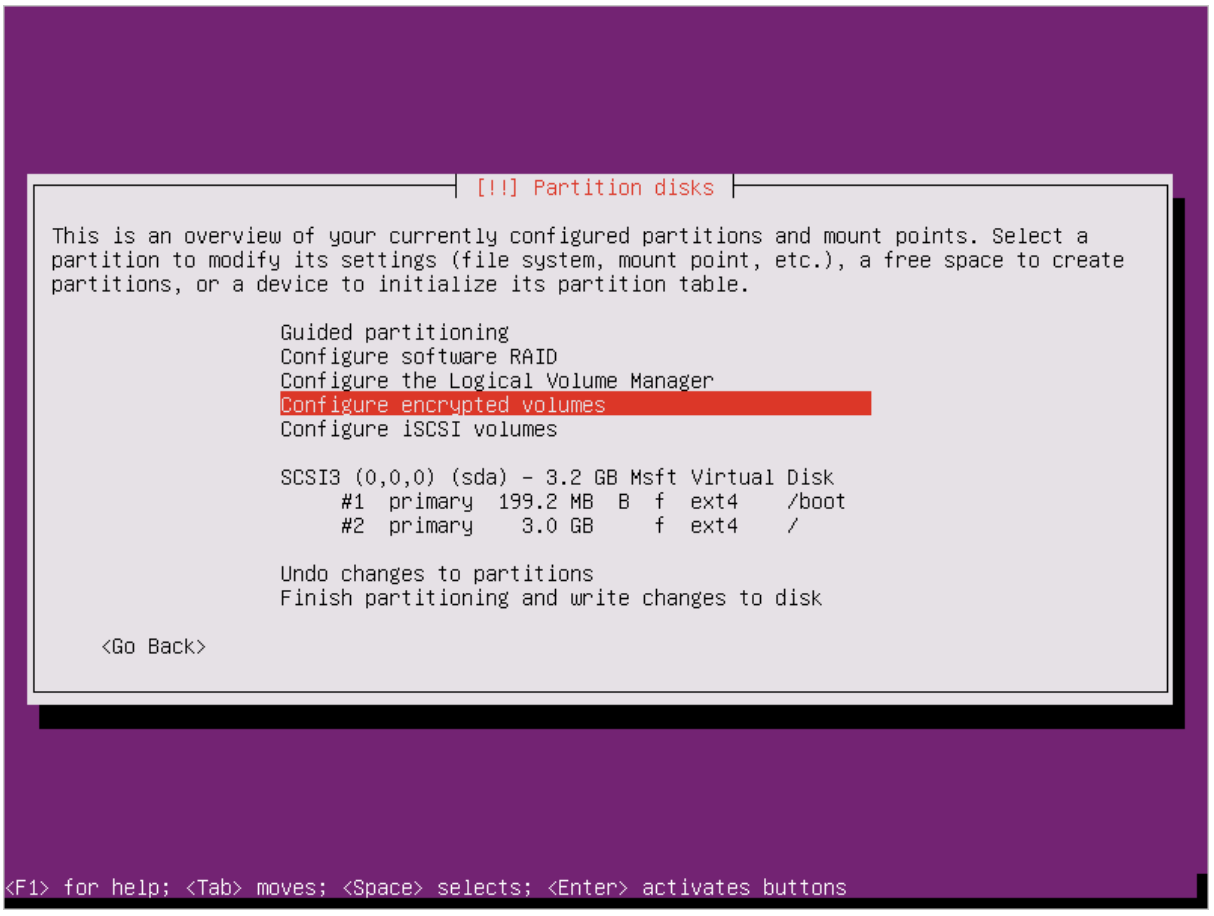
Prepare a pre-encrypted Linux VHD

The preparation for pre-encrypted VHDs can vary depending on the distribution. Examples on preparing Ubuntu, openSUSE, and CentOS 7 are available.

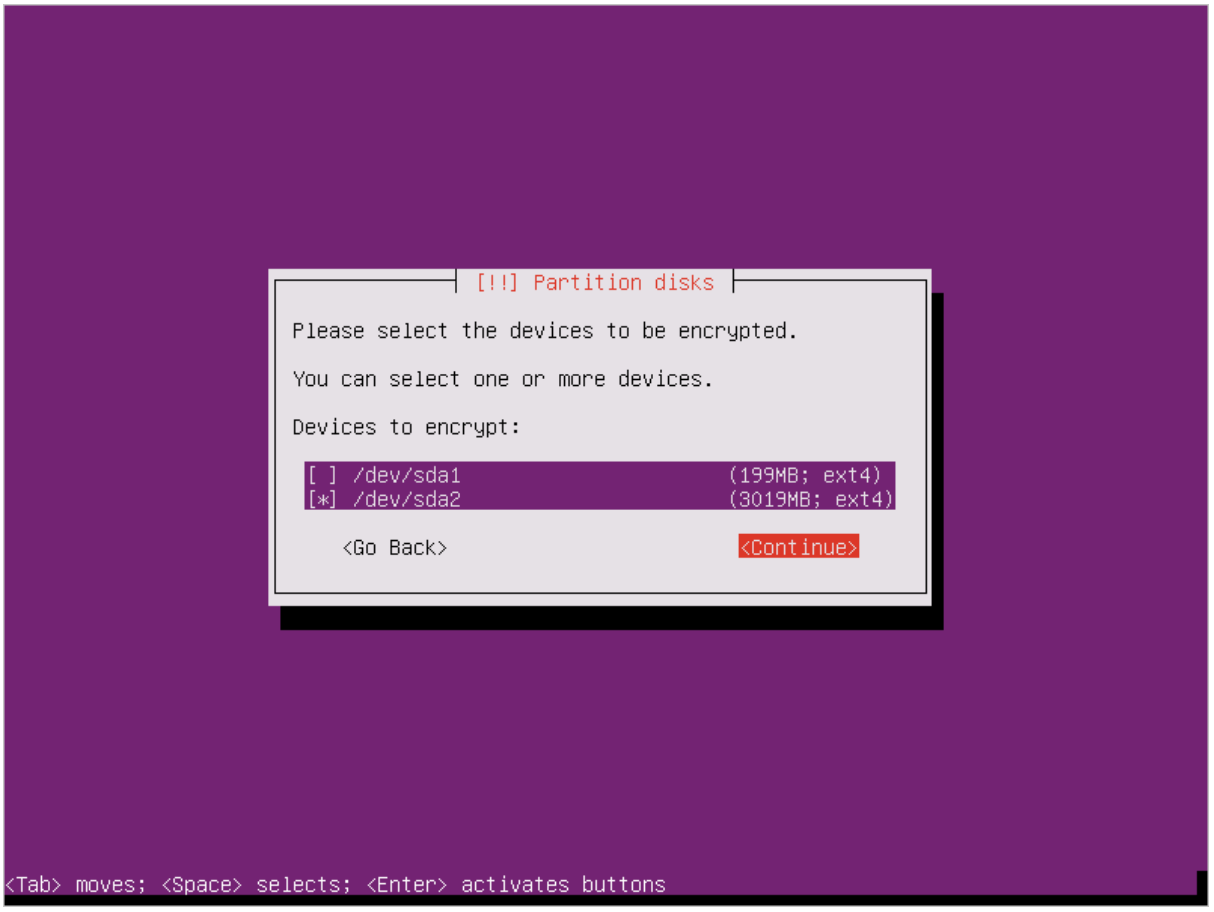
Ubuntu

Configure encryption during the distribution installation by doing the following steps:

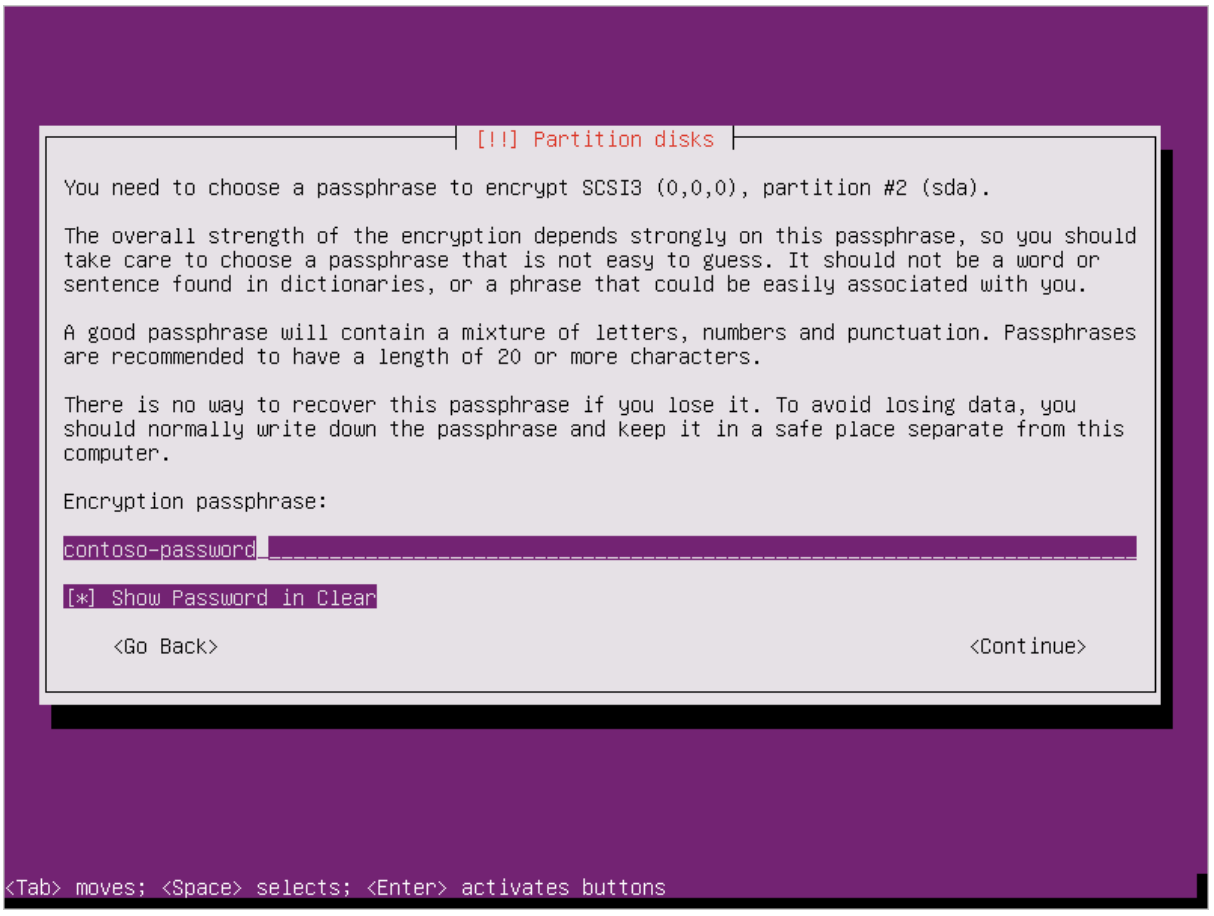
1. Select **Configure encrypted volumes** when you partition the disks.



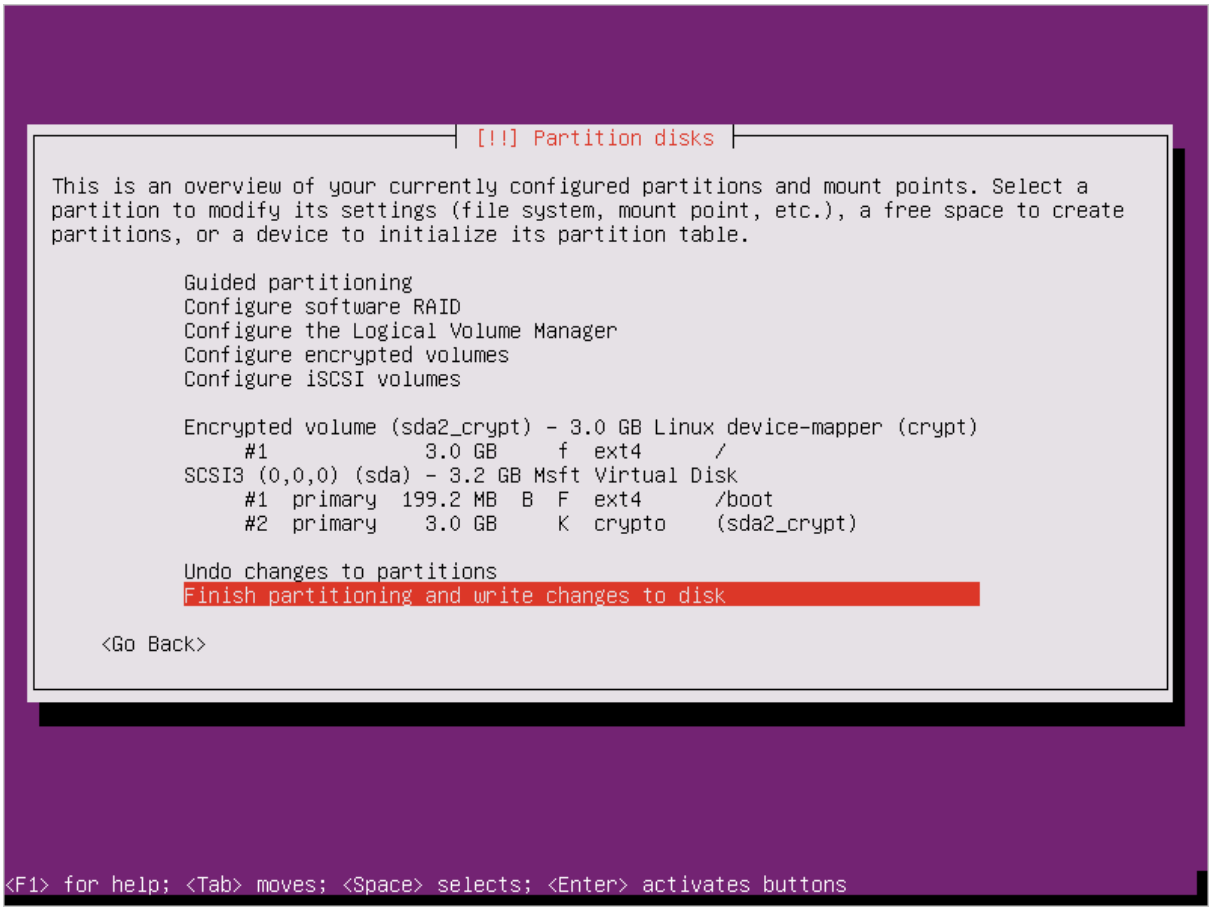
2. Create a separate boot drive, which must not be encrypted. Encrypt your root drive.



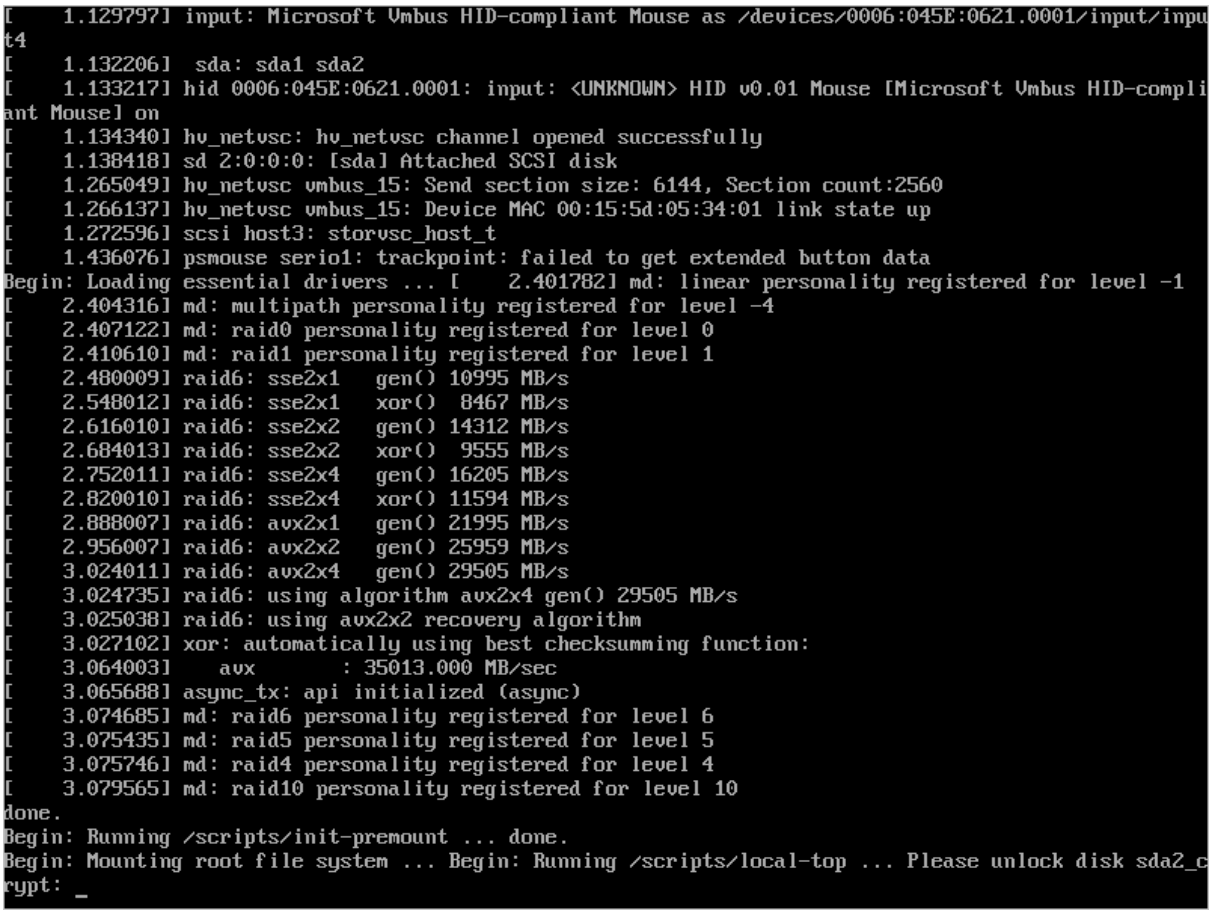
3. Provide a passphrase. This is the passphrase that you uploaded to the key vault.



4. Finish partitioning.



5. When you boot the VM and are asked for a passphrase, use the passphrase you provided in step 3.



6. Prepare the VM for uploading into Azure using these instructions. Don't run the last step (deprovisioning the VM) yet.

Configure encryption to work with Azure by doing the following steps:

1. Create a file under `/usr/local/sbin/azure_crypt_key.sh`, with the content in the following script. Pay attention to the KeyFileName, because it's the passphrase file name used by Azure.

Bash

```
#!/bin/sh
MountPoint=/tmp-keydisk-mount
KeyFileName=LinuxPassPhraseFileName
echo "Trying to get the key from disks ..." >&2
mkdir -p $MountPoint
modprobe vfat >/dev/null 2>&1
modprobe ntfs >/dev/null 2>&1
sleep 2
OPENED=0
cd /sys/block
for DEV in sd*; do

    echo "> Trying device: $DEV ..." >&2
    mount -t vfat -r /dev/${DEV}1 $MountPoint >/dev/null||
    mount -t ntfs -r /dev/${DEV}1 $MountPoint >/dev/null
    if [ -f $MountPoint/$KeyFileName ]; then
            cat $MountPoint/$KeyFileName
```

```
            umount $MountPoint 2>/dev/null
            OPENED=1
            break
    fi
    umount $MountPoint 2>/dev/null
done

  if [ $OPENED -eq 0 ]; then
    echo "FAILED to find suitable passphrase file ..." >&2
    echo -n "Try to enter your password: " >&2
    read -s -r A </dev/console
    echo -n "$A"
 else
    echo "Success loading keyfile!" >&2
fi
```

2. Change the crypt config in */etc/crypttab*. It should look like this:

```config
 xxx_crypt uuid=xxxxxxxxxxxxxxxxxxxx none
luks,discard,keyscript=/usr/local/sbin/azure_crypt_key.sh
```

3. Add executable permissions to the script:

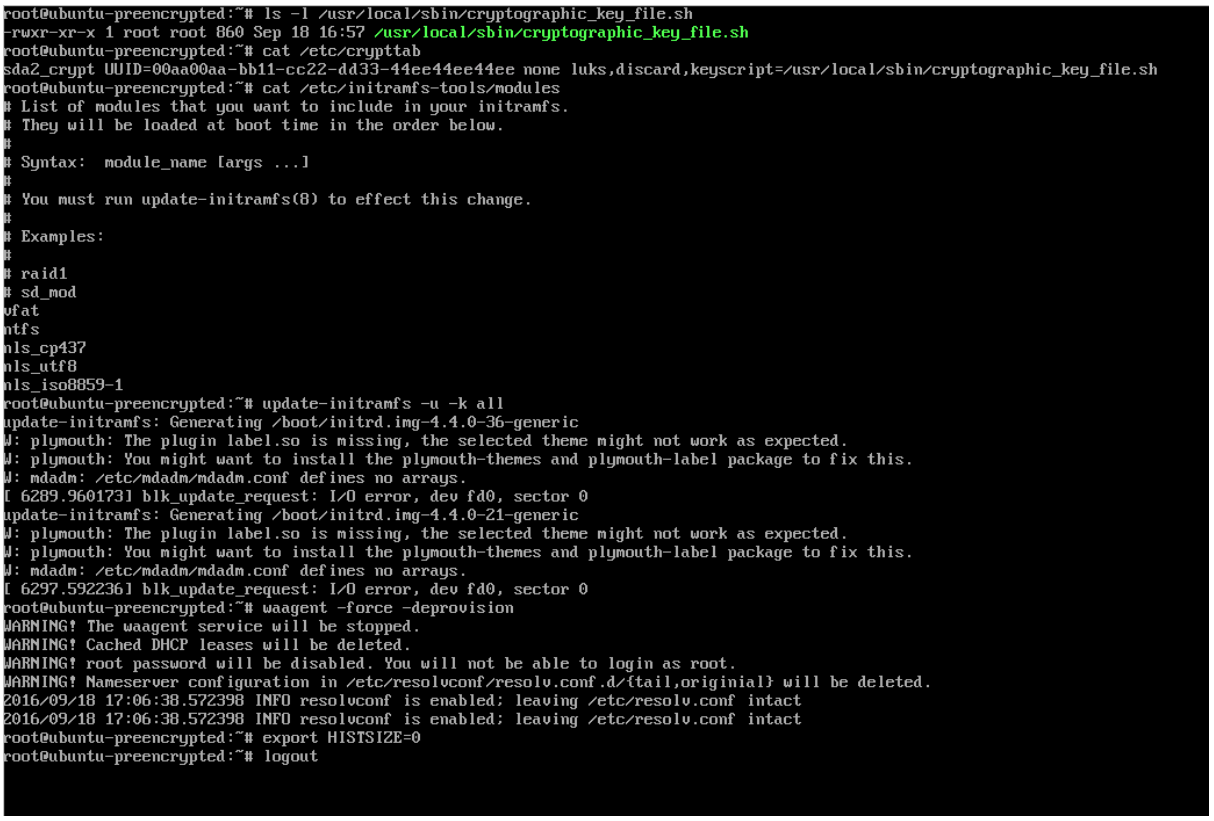
```Bash
 sudo chmod +x /usr/local/sbin/azure_crypt_key.sh
```

4. Edit `/etc/initramfs-tools/modules` by appending lines:

```config
 vfat
 ntfs
 nls_cp437
 nls_utf8
 nls_iso8859-1
```

5. Run `update-initramfs -u -k all` to update the initramfs to make the `keyscript` take effect.

6. Now you can deprovision the VM.



7. Continue to the next step and upload your VHD into Azure.

Upload encrypted VHD to an Azure storage account

After DM-Crypt encryption is enabled, the local encrypted VHD needs to be uploaded to your storage account.

PowerShell

```
    Add-AzVhd [-Destination] <Uri> [-LocalFilePath] <FileInfo> [[-
NumberOfUploaderThreads] <Int32> ] [[-BaseImageUriToPatch] <Uri> ] [[-OverWrite]]
[ <CommonParameters>]
```

Upload the secret for the pre-encrypted VM to your key vault

When encrypting using a Microsoft Entra app (previous release), the disk-encryption secret that you obtained previously must be uploaded as a secret in your key vault. The key vault needs to have disk encryption and permissions enabled for your Microsoft Entra client.

Azure PowerShell

```
 $AadClientId = "My-AAD-Client-Id"
 $AadClientSecret = "My-AAD-Client-Secret"

 $key vault = New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$ResourceGroupName -Location $Location

 Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$ResourceGroupName -ServicePrincipalName $AadClientId -PermissionsToKeys all -
PermissionsToSecrets all
 Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$ResourceGroupName -EnabledForDiskEncryption
```

Disk encryption secret not encrypted with a KEK

To set up the secret in your key vault, use Set-AzKeyVaultSecret. The passphrase is encoded as a base64 string and then uploaded to the key vault. In addition, make sure that the following tags are set when you create the secret in the key vault.

Azure PowerShell

```
 # This is the passphrase that was provided for encryption during the distribution
installation
 $passphrase = "contoso-password"

 $tags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP";
"DiskEncryptionKeyFileName" = "LinuxPassPhraseFileName"}
 $secretName = [guid]::NewGuid().ToString()
 $secretValue =
[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($passphrase))
 $secureSecretValue = ConvertTo-SecureString $secretValue -AsPlainText -Force

 $secret = Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $secretName -
SecretValue $secureSecretValue -tags $tags
 $secretUrl = $secret.Id
```

Use the `$secretUrl` in the next step for attaching the OS disk without using KEK.

Disk encryption secret encrypted with a KEK

Before you upload the secret to the key vault, you can optionally encrypt it by using a key encryption key. Use the wrap API to first encrypt the secret using the key encryption key. The output of this wrap operation is a base64 URL encoded string, which you can then upload as a secret by using the Set-AzKeyVaultSecret cmdlet.

Azure PowerShell

```
    # This is the passphrase that was provided for encryption during the
distribution installation
    $passphrase = "contoso-password"

    Add-AzKeyVaultKey -VaultName $KeyVaultName -Name "keyencryptionkey" -
Destination Software
    $KeyEncryptionKey = Get-AzKeyVaultKey -VaultName $KeyVault.OriginalVault.Name
-Name "keyencryptionkey"

    $apiversion = "2015-06-01"

    ###############################
    # Get Auth URI
    ###############################

    $uri = $KeyVault.VaultUri + "/keys"
    $headers = @{}

    $response = try { Invoke-RestMethod -Method GET -Uri $uri -Headers $headers }
catch { $_.Exception.Response }

    $authHeader = $response.Headers["www-authenticate"]
    $authUri = [regex]::match($authHeader, 'authorization="
(.*?)"').Groups[1].Value

    Write-Host "Got Auth URI successfully"

    ###############################
    # Get Auth Token
    ###############################

    $uri = $authUri + "/oauth2/token"
    $body = "grant_type=client_credentials"
    $body += "&client_id=" + $AadClientId
    $body += "&client_secret=" + [Uri]::EscapeDataString($AadClientSecret)
    $body += "&resource=" + [Uri]::EscapeDataString("https://vault.azure.net")
    $headers = @{}

    $response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body
$body

    $access_token = $response.access_token

    Write-Host "Got Auth Token successfully"

    ###############################
    # Get KEK info
    ###############################

    $uri = $KeyEncryptionKey.Id + "?api-version=" + $apiversion
    $headers = @{"Authorization" = "Bearer " + $access_token}

    $response = Invoke-RestMethod -Method GET -Uri $uri -Headers $headers
```

```
    $keyid = $response.key.kid

    Write-Host "Got KEK info successfully"

    ##############################
    # Encrypt passphrase using KEK
    ##############################

    $passphraseB64 =
[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Passphrase))
    $uri = $keyid + "/encrypt?api-version=" + $apiversion
    $headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" =
"application/json"}
    $bodyObj = @{"alg" = "RSA-OAEP"; "value" = $passphraseB64}
    $body = $bodyObj | ConvertTo-Json

    $response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body
$body

    $wrappedSecret = $response.value

    Write-Host "Encrypted passphrase successfully"

    ##############################
    # Store secret
    ##############################

    $secretName = [guid]::NewGuid().ToString()
    $uri = $KeyVault.VaultUri + "/secrets/" + $secretName + "?api-version=" +
$apiversion
    $secretAttributes = @{"enabled" = $true}
    $secretTags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP";
"DiskEncryptionKeyFileName" = "LinuxPassPhraseFileName"}
    $headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" =
"application/json"}
    $bodyObj = @{"value" = $wrappedSecret; "attributes" = $secretAttributes;
"tags" = $secretTags}
    $body = $bodyObj | ConvertTo-Json

    $response = Invoke-RestMethod -Method PUT -Uri $uri -Headers $headers -Body
$body

    Write-Host "Stored secret successfully"

    $secretUrl = $response.id
```

Use `$KeyEncryptionKey` and `$secretUrl` in the next step for attaching the OS disk using KEK.

Specify a secret URL when you attach an OS disk

Without using a KEK

While you're attaching the OS disk, you need to pass `$secretUrl`. The URL was generated in the "Disk-encryption secret not encrypted with a KEK" section.

powershazurepowershell-interactiveell

```
    Set-AzVMOSDisk `
            -VM $VirtualMachine `
            -Name $OSDiskName `
            -SourceImageUri $VhdUri `
            -VhdUri $OSDiskUri `
            -Linux `
            -CreateOption FromImage `
            -DiskEncryptionKeyVaultId $KeyVault.ResourceId `
            -DiskEncryptionKeyUrl $SecretUrl
```

Using a KEK

When you attach the OS disk, pass `$KeyEncryptionKey` and `$secretUrl`. The URL was generated in the "Disk encryption secret encrypted with a KEK" section.

Azure PowerShell

```
    Set-AzVMOSDisk `
            -VM $VirtualMachine `
            -Name $OSDiskName `
            -SourceImageUri $CopiedTemplateBlobUri `
            -VhdUri $OSDiskUri `
            -Linux `
            -CreateOption FromImage `
            -DiskEncryptionKeyVaultId $KeyVault.ResourceId `
            -DiskEncryptionKeyUrl $SecretUrl `
            -KeyEncryptionKeyVaultId $KeyVault.ResourceId `
            -KeyEncryptionKeyURL $KeyEncryptionKey.Id
```

Azure Disk Encryption on an isolated network

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

> **Caution**
>
> This article references CentOS, a Linux distribution that is End Of Life (EOL) status. Consider your use and plan accordingly. For more information, see the CentOS End Of Life guidance.

Applies to: ✔ Linux VMs ✔ Flexible scale sets.

When connectivity is restricted by a firewall, proxy requirement, or network security group (NSG) settings, the ability of the extension to perform needed tasks might be disrupted. This disruption can result in status messages such as "Extension status not available on the VM."

Package management

Azure Disk Encryption depends on many components, which are typically installed as part of ADE enablement if not already present. When behind a firewall or otherwise isolated from the Internet, these packages must be preinstalled or available locally.

Here are the packages necessary for each distribution. For a full list of supported distros and volume types, see supported VMs and operating systems.

- **Ubuntu 14.04, 16.04, 18.04**: lsscsi, psmisc, at, cryptsetup-bin, python-parted, python-six, procps, grub-pc-bin
- **CentOS 7.2 - 7.9, 8.1, 8.2**: lsscsi, psmisc, lvm2, uuid, at, patch, cryptsetup, cryptsetup-reencrypt, pyparted, procps-ng, util-linux
- **CentOS 6.8**: lsscsi, psmisc, lvm2, uuid, at, cryptsetup-reencrypt, parted, python-six

- **RedHat 7.2 - 7.9, 8.1, 8.2**: lsscsi, psmisc, lvm2, uuid, at, patch, cryptsetup, cryptsetup-reencrypt, procps-ng, util-linux
- **RedHat 6.8**: lsscsi, psmisc, lvm2, uuid, at, patch, cryptsetup-reencrypt
- **openSUSE 42.3, SLES 12-SP4, 12-SP3**: lsscsi, cryptsetup

On Red Hat, when a proxy is required, you must make sure that the subscription-manager and yum are set up properly. For more information, see How to troubleshoot subscription-manager and yum problems.

When packages are installed manually, they must also be manually upgraded as new versions are released.

Network security groups

Any network security group settings that are applied must still allow the endpoint to meet the documented network configuration prerequisites for disk encryption. See Azure Disk Encryption: Networking requirements

Azure Disk Encryption with Microsoft Entra ID (previous version)

If using Azure Disk Encryption with Microsoft Entra ID (previous version), the Microsoft Authentication Library must be installed manually for all distros (in addition to the packages appropriate for the distro).

When encryption is being enabled with Microsoft Entra credentials, the target VM must allow connectivity to both Microsoft Entra endpoints and Key Vault endpoints. Current Microsoft Entra authentication endpoints are maintained in sections 56 and 59 of the Microsoft 365 URLs and IP address ranges documentation. Key Vault instructions are provided in the documentation on how to Access Azure Key Vault behind a firewall.

Azure Instance Metadata Service

The virtual machine must be able to access the Azure Instance Metadata service endpoint, which uses a well-known nonroutable IP address (`169.254.169.254`) that can be accessed only from within the VM. Proxy configurations that alter local HTTP traffic to this address (for example, adding an X-Forwarded-For header) aren't supported.

Next steps

- See more steps for Azure disk encryption troubleshooting
- Azure data encryption at rest

Verify encryption status for Linux

Summarize this article for me

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The scope of this article is to validate the encryption status of a virtual machine by using different methods: the Azure portal, PowerShell, the Azure CLI, or the operating system of the virtual machine (VM).

You can validate the encryption status during or after the encryption, by either:

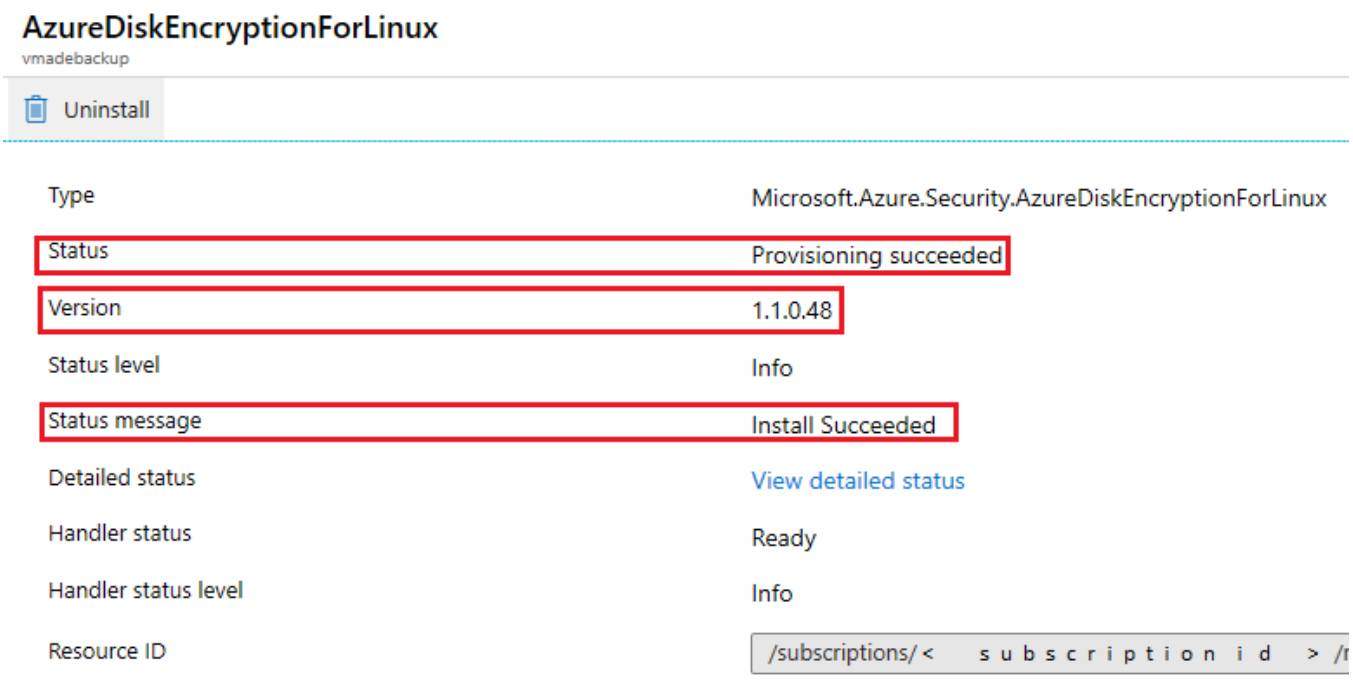
- Checking the disks attached to a particular VM.
- Querying the encryption settings on each disk, whether the disk is attached or unattached.

This scenario applies for Azure Disk Encryption dual-pass and single-pass extensions. Linux distributions are the only environment for this scenario.

> **Note**
>
> We're using variables throughout the article. Replace the values accordingly.

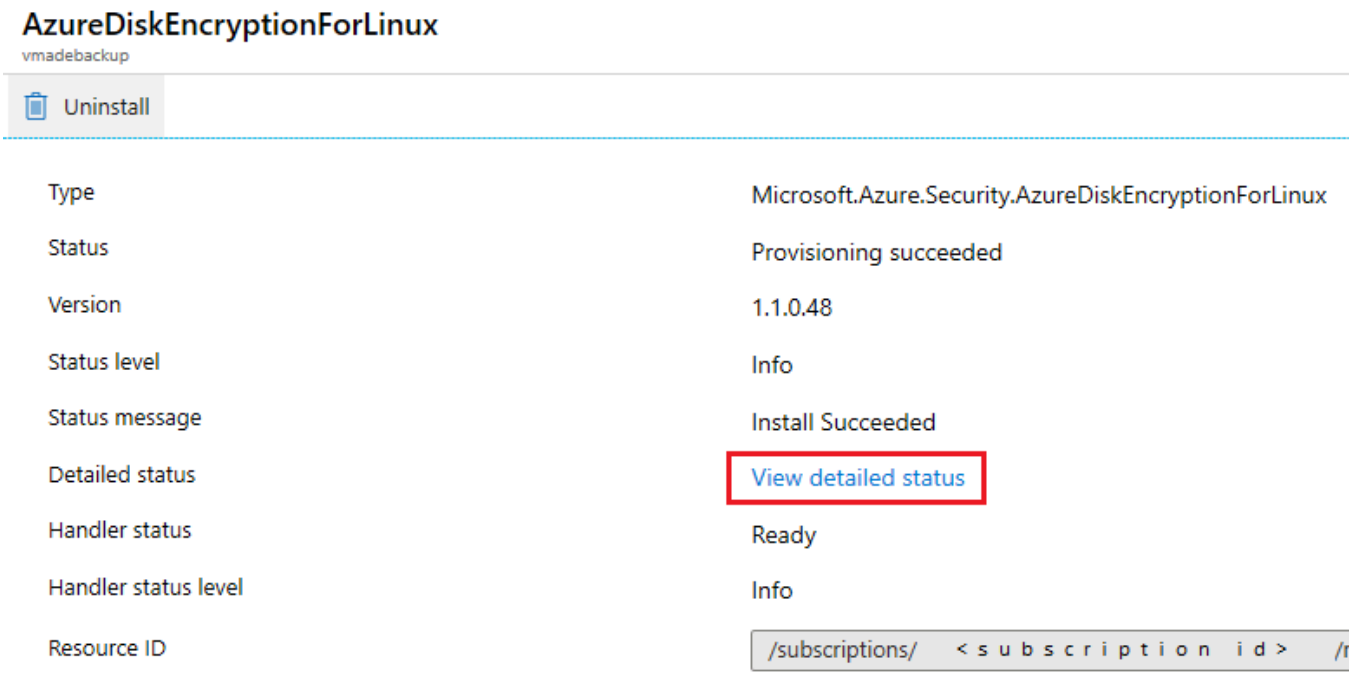
Portal

In the Azure portal, inside the **Extensions** section, select the Azure Disk Encryption extension in the list. The information for **Status message** indicates the current encryption status:



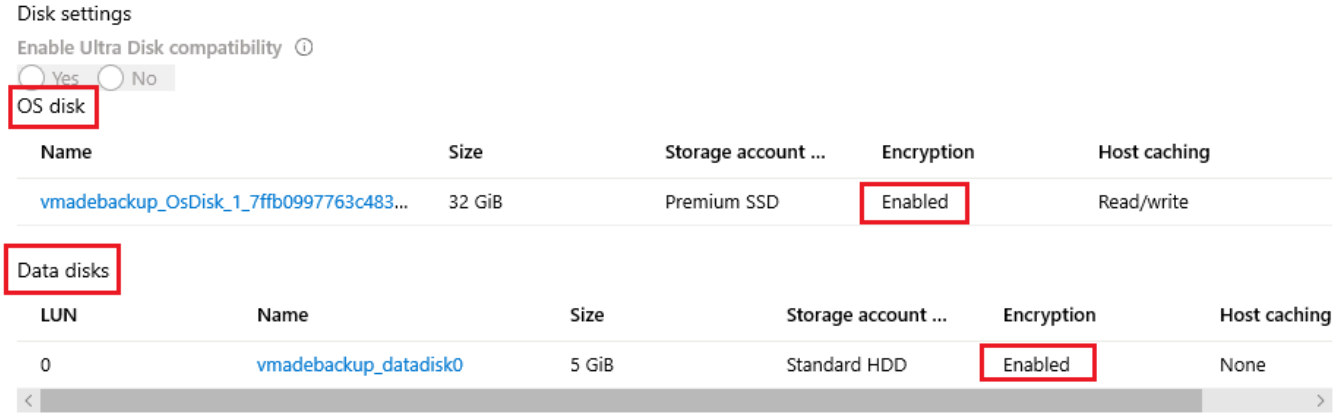
In the list of extensions, you'll see the corresponding Azure Disk Encryption extension version. Version 0.x corresponds to Azure Disk Encryption dual pass, and version 1.x corresponds to Azure Disk Encryption single pass.

You can get more details by selecting the extension and then selecting **View detailed status**. The detailed status of the encryption process appears in JSON format.





Another way to validate the encryption status is by looking at the **Disk settings** section.



> **Note**
>
> This status means the disks have encryption settings stamped, not that they were actually encrypted at the OS level.
>
> By design, the disks are stamped first and encrypted later. If the encryption process fails, the disks may end up stamped but not encrypted.
>
> To confirm if the disks are truly encrypted, you can double check the encryption of each disk at the OS level.

PowerShell

You can validate the *general* encryption status of an encrypted VM by using the following PowerShell commands:

Azure PowerShell

```
$VMNAME="VMNAME"
$RGNAME="RGNAME"
```

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName  ${RGNAME} -VMName ${VMNAME}
```

```
PS Azure:\> Get-AzVmDiskEncryptionStatus -ResourceGroupName ${RGNAME} -VMName ${VMNAME}

OsVolumeEncrypted          : NotEncrypted
DataVolumesEncrypted       : Encrypted
OsVolumeEncryptionSettings :
ProgressMessage            : Provisioning succeeded
```

You can capture the encryption settings from each disk by using the following PowerShell commands.

Single pass

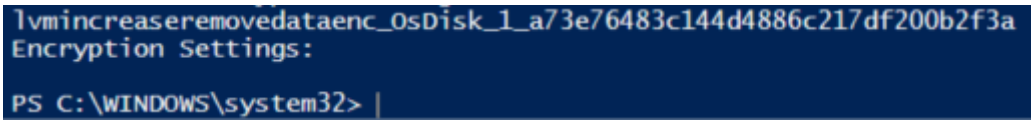
In a single pass, the encryption settings are stamped on each of the disks (OS and data). You can capture the encryption settings for an OS disk in a single pass as follows:

Azure PowerShell

```
$RGNAME = "RGNAME"
$VMNAME = "VMNAME"

$VM = Get-AzVM -Name ${VMNAME} -ResourceGroupName ${RGNAME}
 $Sourcedisk = Get-AzDisk -ResourceGroupName ${RGNAME} -DiskName
$VM.StorageProfile.OsDisk.Name
 Write-Host
"================================================================================
================================================================================"
 Write-Host "Encryption Settings:"
 Write-Host
"================================================================================
================================================================================"
 Write-Host "Enabled:" $Sourcedisk.EncryptionSettingsCollection.Enabled
 Write-Host "Version:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettingsVersion
 Write-Host "Source Vault:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Sourc
eVault.Id
 Write-Host "Secret URL:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Secre
tUrl
 Write-Host "Key URL:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.KeyEncryptionKey.KeyUrl
 Write-Host
"================================================================================
================================================================================"
```



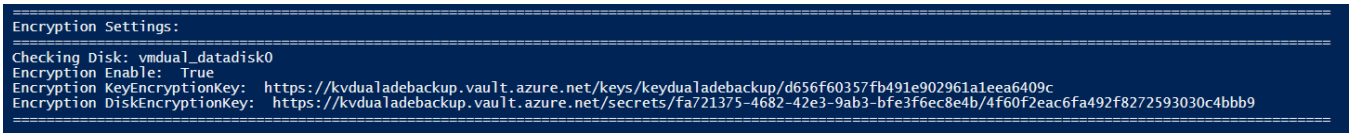
If the disk doesn't have encryption settings stamped, the output will be empty:



Use the following commands to capture encryption settings for data disks:

Azure PowerShell

```
$RGNAME = "RGNAME"
$VMNAME = "VMNAME"

$VM = Get-AzVM -Name ${VMNAME} -ResourceGroupName ${RGNAME}
 clear
 foreach ($i in $VM.StorageProfile.DataDisks|ForEach-Object{$_.Name})
 {
 Write-Host
"================================================================================
================================================================================"
 Write-Host "Encryption Settings:"
 Write-Host
"================================================================================
================================================================================"
 Write-Host "Checking Disk:" $i
 $Disk=(Get-AzDisk -ResourceGroupName ${RGNAME} -DiskName $i)
 Write-Host "Encryption Enable: " $Sourcedisk.EncryptionSettingsCollection.Enabled
 Write-Host "Encryption KeyEncryptionKey: "
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.KeyEncryptionKey.KeyUrl
;
 Write-Host "Encryption DiskEncryptionKey: "
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Secre
tUrl;
 Write-Host
"================================================================================
================================================================================"
 }
```



Dual pass

In a dual pass, the encryption settings are stamped in the VM model and not on each individual disk.

To verify that the encryption settings were stamped in a dual pass, use the following commands:

Azure PowerShell

```
$RGNAME = "RGNAME"
$VMNAME = "VMNAME"

$vm = Get-AzVm -ResourceGroupName ${RGNAME} -Name ${VMNAME};
$Sourcedisk = Get-AzDisk -ResourceGroupName ${RGNAME} -DiskName
$VM.StorageProfile.OsDisk.Name
clear
Write-Host
"================================================================================
================================================================================"
Write-Host "Encryption Settings:"
Write-Host
"================================================================================
================================================================================"
Write-Host "Enabled:" $Sourcedisk.EncryptionSettingsCollection.Enabled
Write-Host "Version:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettingsVersion
Write-Host "Source Vault:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Sourc
eVault.Id
Write-Host "Secret URL:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Secre
tUrl
Write-Host "Key URL:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.KeyEncryptionKey.KeyUrl
Write-Host
"================================================================================
================================================================================"
```

Unattached disks

Check the encryption settings for disks that aren't attached to a VM.

Managed disks

Azure PowerShell

```
$Sourcedisk = Get-AzDisk -ResourceGroupName ${RGNAME} -DiskName ${TARGETDISKNAME}
Write-Host
"================================================================================
================================================================================"
Write-Host "Encryption Settings:"
Write-Host
"================================================================================
================================================================================"
```

```
Write-Host "Enabled:" $Sourcedisk.EncryptionSettingsCollection.Enabled
Write-Host "Version:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettingsVersion
Write-Host "Source Vault:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Sourc
eVault.Id
Write-Host "Secret URL:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.DiskEncryptionKey.Secre
tUrl
Write-Host "Key URL:"
$Sourcedisk.EncryptionSettingsCollection.EncryptionSettings.KeyEncryptionKey.KeyUrl
Write-Host
"================================================================================
================================================================================"
```

Azure CLI

You can validate the *general* encryption status of an encrypted VM by using the following Azure CLI commands:

Azure CLI

```
VMNAME="VMNAME"
RGNAME="RGNAME"
az vm encryption show --name ${VMNAME} --resource-group ${RGNAME} --query
"substatus"
```

```
$ az vm encryption show --name ${VMNAME} --resource-group ${RGNAME} --query "substatus" -o table
Code                                                                    Level    DisplayStatus          Message
----------------------------------------------------------------------  -------  ---------------------  ----------------------------------------
ComponentStatus/Microsoft.Azure.Security.AzureDiskEncryptionForLinux/succeeded  Info     Provisioning succeeded  {"os": "Encrypted", "data": "Encrypted"}
```

Single pass

You can validate the encryption settings for each disk by using the following Azure CLI commands:

Azure CLI

```
az vm encryption show -g ${RGNAME} -n ${VMNAME} --query "disks[*].[name,
statuses[*].displayStatus]"  -o table
```

```
Column1                                                                 Column2
----------------------------------------------------------------------  --------------------------------
lvmincreaseremovedataenc_OsDisk_1_a73e76483c144d4886c217df200b2f3a  ['Disk is not encrypted']
lvmincreaseremovedataenc_datadisk1                                  ['Encryption is enabled on disk']
lvmincreaseremovedataenc_datadisk2                                  ['Encryption is enabled on disk']
lvmincreaseremovedataenc_datadisk3                                  ['Encryption is enabled on disk']
lvmincreaseremovedataenc_datadisk4                                  ['Encryption is enabled on disk']
```

> **ⓘ Important**
>
> If the disk doesn't have encryption settings stamped, you'll see the text **Disk is not encrypted**.

Use the following commands to get detailed status and encryption settings.

OS disk:

Azure CLI

```
RGNAME="RGNAME"
VMNAME="VNAME"

disk=`az vm show -g ${RGNAME} -n ${VMNAME} --query storageProfile.osDisk.name -o
tsv`
for disk in $disk; do \
echo
"================================================================================
================================================================================"
echo -ne "Disk Name: "; az disk show -g ${RGNAME} -n ${disk} --query name -o tsv; \
echo -ne "Encryption Enabled: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.enabled -o tsv; \
echo -ne "Version: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --query
encryptionSettingsCollection.encryptionSettingsVersion -o tsv; \
echo -ne "Disk Encryption Key: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.encryptionSettings[].diskEncryptionKey.secretUrl -o
tsv; \
echo -ne "key Encryption Key: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.encryptionSettings[].keyEncryptionKey.keyUrl -o tsv; \
echo
"================================================================================
================================================================================"
done
```

```
================================================================================
Disk Name: vmadebackup_OsDisk_1_7ffb0997763c483881d53c54e58f4898
Encryption Enabled: true
Disk Encryption Key: https://kvadebackup.vault.azure.net/secrets/5dfe7d9d-c77e-478f-aa2d-321f62c394d9/6588ab5e10f54423b0f35ec6a3a64f1d
key Encryption Key: https://kvadebackup.vault.azure.net/keys/keyadebackup/1551be9b670947428b04b6aa0a31a040
================================================================================
```

Data disks:

Azure CLI

```
RGNAME="RGNAME"
VMNAME="VMNAME"
az vm encryption show --name ${VMNAME} --resource-group ${RGNAME} --query
"substatus"

for disk in `az vm show -g ${RGNAME} -n ${VMNAME} --query
```

```
storageProfile.dataDisks[].name -o tsv`; do \
echo
"================================================================================
================================================================================"; \
echo -ne "Disk Name: "; az disk show -g ${RGNAME} -n ${disk} --query name -o tsv; \
echo -ne "Encryption Enabled: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.enabled -o tsv; \
echo -ne "Version: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --query
encryptionSettingsCollection.encryptionSettingsVersion -o tsv; \
echo -ne "Disk Encryption Key: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.encryptionSettings[].diskEncryptionKey.secretUrl -o
tsv; \
echo -ne "key Encryption Key: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.encryptionSettings[].keyEncryptionKey.keyUrl -o tsv; \
echo
"================================================================================
================================================================================"
done
```

```
================================================================================================================================================================
Disk Name: vmadebackup_datadisk0
Encryption Enabled: true
Disk Encryption Key: https://kvadebackup.vault.azure.net/secrets/5dfe7d9d-c77e-478f-aa2d-321f62c394d9/4d0c7805d33e47dd81838aabccfff85a
key Encryption Key: https://kvadebackup.vault.azure.net/keys/keyadebackup/1551be9b670947428b04b6aa0a31a040
================================================================================================================================================================
```

Dual pass

Azure CLI

```
az vm encryption show --name ${VMNAME} --resource-group ${RGNAME} -o table
```

```
OsDisk     DataDisk    OsType    ProgressMessage
---------  ----------  --------  ------------------------------------------------------------------------------------------------------------------------
Encrypted  Encrypted   Linux     https://kvdualadebackup.vault.azure.net/secrets/fa721375-4682-42e3-9ab3-bfe3f6ec8e4b/4f60f2eac6fa492f8272593030c4bbb9
```

You can also check the encryption settings on the VM Model Storage profile of the OS disk:

Azure CLI

```
disk=`az vm show -g ${RGNAME} -n ${VMNAME} --query storageProfile.osDisk.name -o
tsv`
for disk in $disk; do \
echo
"================================================================================
================================================================================"; \
echo -ne "Disk Name: "; az disk show -g ${RGNAME} -n ${disk} --query name -o tsv; \
echo -ne "Encryption Enabled: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.enabled -o tsv; \
echo -ne "Version: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --query
encryptionSettingsCollection.encryptionSettingsVersion -o tsv; \
echo -ne "Disk Encryption Key: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.encryptionSettings[].diskEncryptionKey.secretUrl -o
tsv; \
```

```
echo -ne "key Encryption Key: "; az disk show -g ${RGNAME} -n ${disk} --query
encryptionSettingsCollection.encryptionSettings[].keyEncryptionKey.keyUrl -o tsv; \
echo
"================================================================================
================================================================================"
done
```

```
================================================================================
Disk Name: vmdual_OsDisk_1_16c83fa21fe649a8846d395a9104bd2b
Encryption Enabled: true
Disk Encryption Key: https://kvdualadebackup.vault.azure.net/secrets/fa721375-4682-42e3-9ab3-bfe3f6ec8e4b/4f60f2eac6fa492f8272593030c4bbb9
key Encryption Key: https://kvdualadebackup.vault.azure.net/keys/keydualadebackup/d656f60357fb491e902961a1eea6409c
================================================================================
```

Unattached disks

Check the encryption settings for disks that aren't attached to a VM.

Managed disks

Azure CLI

```
RGNAME="RGNAME"
TARGETDISKNAME="DISKNAME"
echo
"================================================================================
================================================================================"
echo -ne "Disk Name: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --query name
-o tsv; \
echo -ne "Encryption Enabled: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --
query encryptionSettingsCollection.enabled -o tsv; \
echo -ne "Version: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --query
encryptionSettingsCollection.encryptionSettingsVersion -o tsv; \
echo -ne "Disk Encryption Key: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --
query encryptionSettingsCollection.encryptionSettings[].diskEncryptionKey.secretUrl
-o tsv; \
echo -ne "key Encryption Key: "; az disk show -g ${RGNAME} -n ${TARGETDISKNAME} --
query encryptionSettingsCollection.encryptionSettings[].keyEncryptionKey.keyUrl -o
tsv; \
echo
"================================================================================
================================================================================"
```

Unmanaged disks

On March 31st, 2026, unmanaged disks are being retired. If you're using unmanaged disks, migrate to managed disks before that date. For details on the retirement, see Migrate your Azure unmanaged disks by March 31st, 2026.

Unmanaged disks are VHD files that are stored as page blobs in Azure storage accounts.

To get the details for a specific disk, you need to provide:

- The ID of the storage account that contains the disk.
- A connection string for that particular storage account.
- The name of the container that stores the disk.
- The disk name.

This command lists all the IDs for all your storage accounts:

Azure CLI

```
az storage account list --query [].[id] -o tsv
```

The storage account IDs are listed in the following form:

/subscriptions/<subscription id>/resourceGroups/<resource group name>/providers/Microsoft.Storage/storageAccounts/<storage account name>

Select the appropriate ID and store it on a variable:

Azure CLI

```
id="/subscriptions/<subscription id>/resourceGroups/<resource group name>/providers/Microsoft.Storage/storageAccounts/<storage account name>"
```

This command gets the connection string for one particular storage account and stores it on a variable:

Azure CLI

```
ConnectionString=$(az storage account show-connection-string --ids $id --query connectionString -o tsv)
```

The following command lists all the containers under a storage account:

Azure CLI

```
az storage container list --connection-string $ConnectionString --query [].[name] -o tsv
```

The container used for disks is normally named "vhds."

Store the container name on a variable:

Azure CLI

```
ContainerName="name of the container"
```

Use this command to list all the blobs on a particular container:

Azure CLI

```
az storage blob list -c ${ContainerName} --connection-string $ConnectionString --query [].[name] -o tsv
```

Choose the disk that you want to query and store its name on a variable:

Azure CLI

```
DiskName="diskname.vhd"
```

Query the disk encryption settings:

Azure CLI

```
az storage blob show -c ${ContainerName} --connection-string ${ConnectionString} -n ${DiskName} --query metadata.DiskEncryptionSettings
```

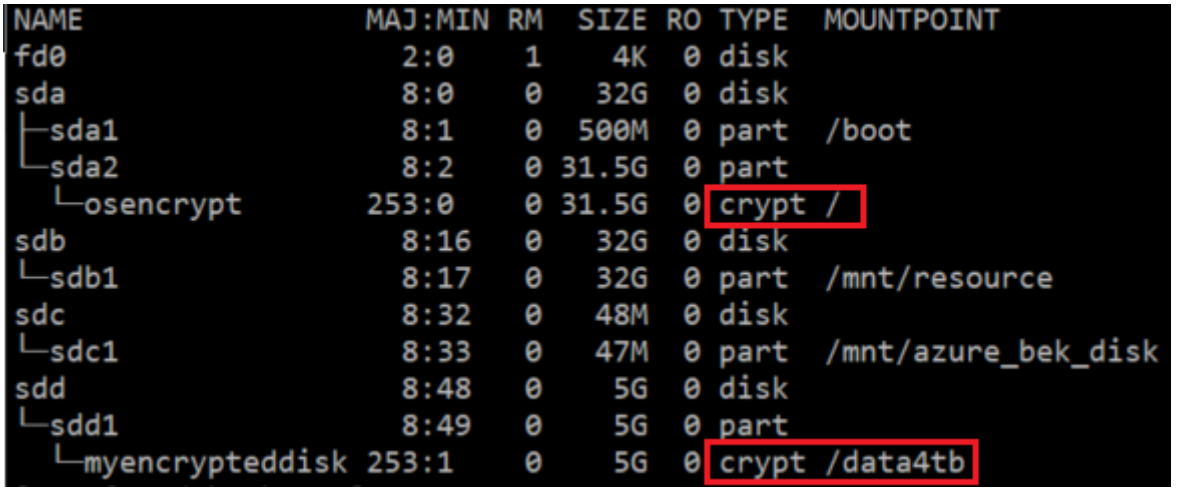
Operating system

Validate if the data disk partitions are encrypted (and the OS disk isn't).

When a partition or disk is encrypted, it's displayed as a **crypt** type. When it's not encrypted, it's displayed as a **part/disk** type.

Bash

```
sudo lsblk
```

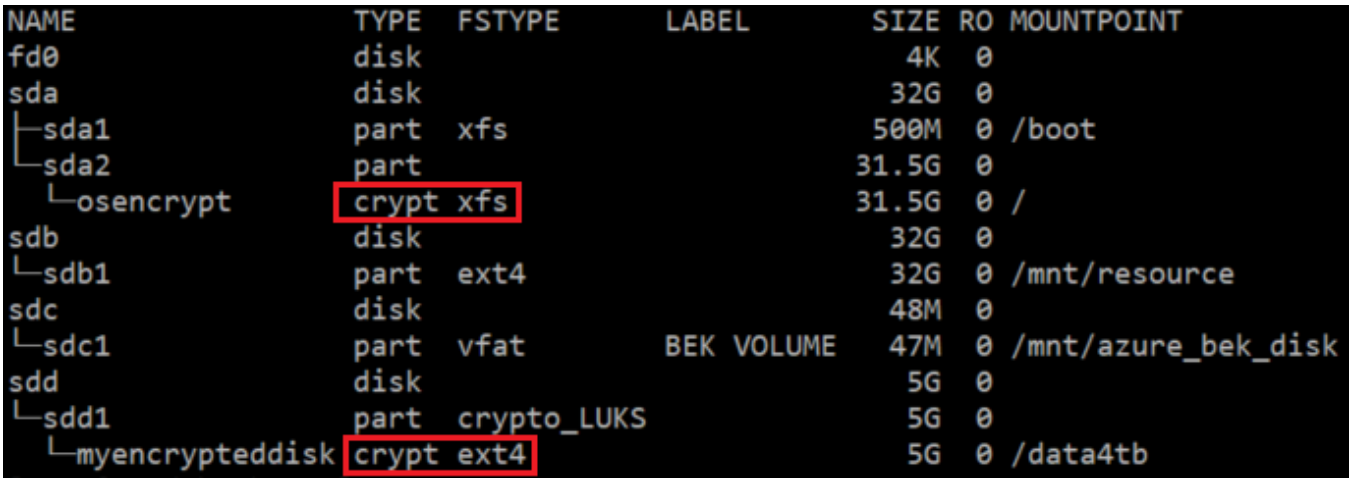


You can get more details by using the following **lsblk** variant.

You'll see a **crypt** type layer that is mounted by the extension. The following example shows logical volumes and normal disks having **crypto_LUKS FSTYPE**.

Bash

```
sudo lsblk -o NAME,TYPE,FSTYPE,LABEL,SIZE,RO,MOUNTPOINT
```



As an extra step, you can validate if the data disk has any keys loaded:

Bash

```
sudo cryptsetup luksDump /dev/VGNAME/LVNAME
```

Bash

```
sudo cryptsetup luksDump /dev/sdd1
```

And you can check which **dm** devices are listed as **crypt**:

Bash

```
sudo dmsetup ls --target crypt
```

Next steps

- Azure Disk Encryption troubleshooting

Last updated on 02/22/2026

Configure LVM and RAID on encrypted devices

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article is a step-by-step process for how to perform Logical Volume Management (LVM) and RAID on encrypted devices. The process applies to the following environments:

- Linux distributions
  - RHEL 7.6+
  - Ubuntu 18.04+
  - SUSE 12+
- Azure Disk Encryption single-pass extension
- Azure Disk Encryption dual-pass extension

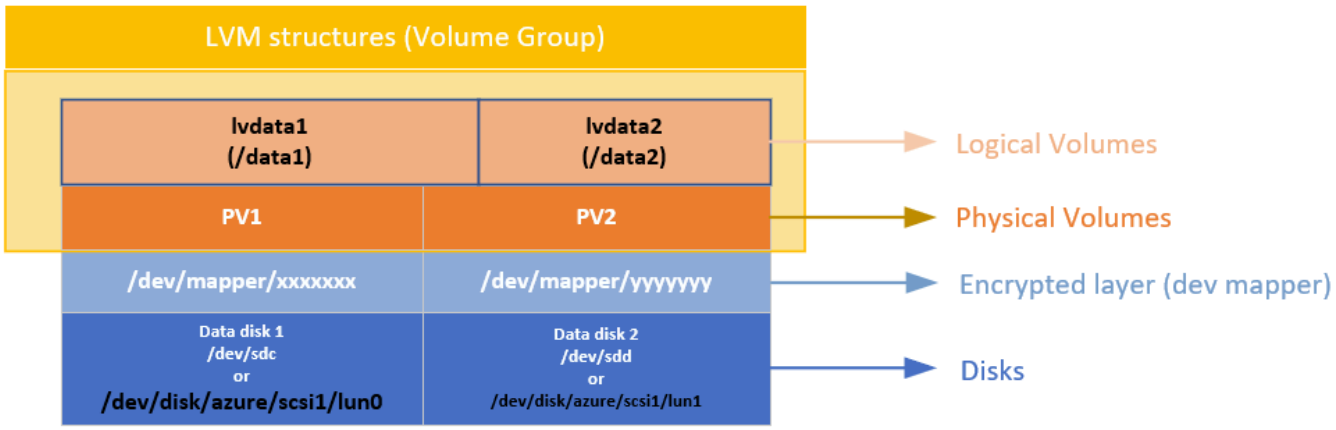
Scenarios

The procedures in this article support the following scenarios:

- Configure LVM on top of encrypted devices (LVM-on-crypt)
- Configure RAID on top of encrypted devices (RAID-on-crypt)

After the underlying device or devices are encrypted, then you can create the LVM or RAID structures on top of that encrypted layer.

The physical volumes (PVs) are created on top of the encrypted layer. The physical volumes are used to create the volume group. You create the volumes and add the required entries on /etc/fstab.



In a similar way, the RAID device is created on top of the encrypted layer on the disks. A file system is created on top of the RAID device and added to /etc/fstab as a regular device.

Considerations

We recommend that you use LVM-on-crypt. RAID is an option when LVM can't be used because of specific application or environment limitations.

You'll use the **EncryptFormatAll** option. For more information about this option, see Use the EncryptFormatAll feature for data disks on Linux VMs.

Although you can use this method when you're also encrypting the OS, we're just encrypting data drives here.

The procedures assume that you already reviewed the prerequisites in Azure Disk Encryption scenarios on Linux VMs and in Quickstart: Create and encrypt a Linux VM with the Azure CLI.

The Azure Disk Encryption dual-pass version is on a deprecation path and should no longer be used on new encryptions.

General steps

When you're using the "on-crypt" configurations, use the process outlined in the following procedures.

> ⓘ **Note**
>
> We're using variables throughout the article. Replace the values accordingly.

Deploy a VM

The following commands are optional, but we recommend that you apply them on a newly deployed virtual machine (VM).

PowerShell:

PowerShell

```
New-AzVm -ResourceGroupName ${RGNAME} `
-Name ${VMNAME} `
-Location ${LOCATION} `
-Size ${VMSIZE} `
-Image ${OSIMAGE} `
-Credential ${creds} `
-Verbose
```

Azure CLI:

Azure CLI

```
az vm create \
-n ${VMNAME} \
-g ${RGNAME} \
--image ${OSIMAGE} \
--admin-username ${username} \
--admin-password ${password} \
-l ${LOCATION} \
--size ${VMSIZE} \
-o table
```

Attach disks to the VM

Repeat the following commands for `$N` number of new disks that you want to attach to the VM.

PowerShell:

PowerShell

```
$storageType = 'Standard_LRS'
$dataDiskName = ${VMNAME} + '_datadisk0'
$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $LOCATION -
CreateOption Empty -DiskSizeGB 5
$dataDisk1 = New-AzDisk -DiskName $dataDiskName -Disk $diskConfig -
ResourceGroupName ${RGNAME}
$vm = Get-AzVM -Name ${VMNAME} -ResourceGroupName ${RGNAME}
$vm = Add-AzVMDataDisk -VM $vm -Name $dataDiskName -CreateOption Attach -
```

```
ManagedDiskId $dataDisk1.Id -Lun 0
Update-AzVM -VM ${VM} -ResourceGroupName ${RGNAME}
```

Azure CLI:

Azure CLI

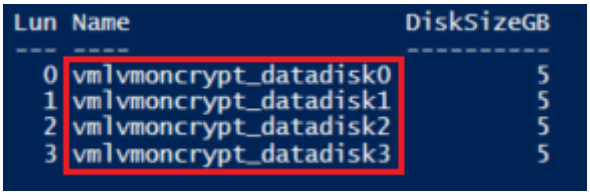
```
az vm disk attach \
-g ${RGNAME} \
--vm-name ${VMNAME} \
--name ${VMNAME}datadisk1 \
--size-gb 5 \
--new \
-o table
```

Verify that the disks are attached to the VM

PowerShell:

PowerShell

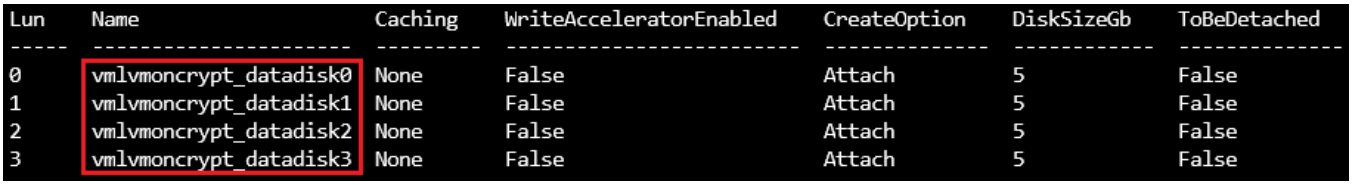
```
$VM = Get-AzVM -ResourceGroupName ${RGNAME} -Name ${VMNAME}
$VM.StorageProfile.DataDisks | Select-Object Lun,Name,DiskSizeGB
```



Azure CLI:

Azure CLI

```
az vm show -g ${RGNAME} -n ${VMNAME} --query storageProfile.dataDisks -o table
```



Portal:

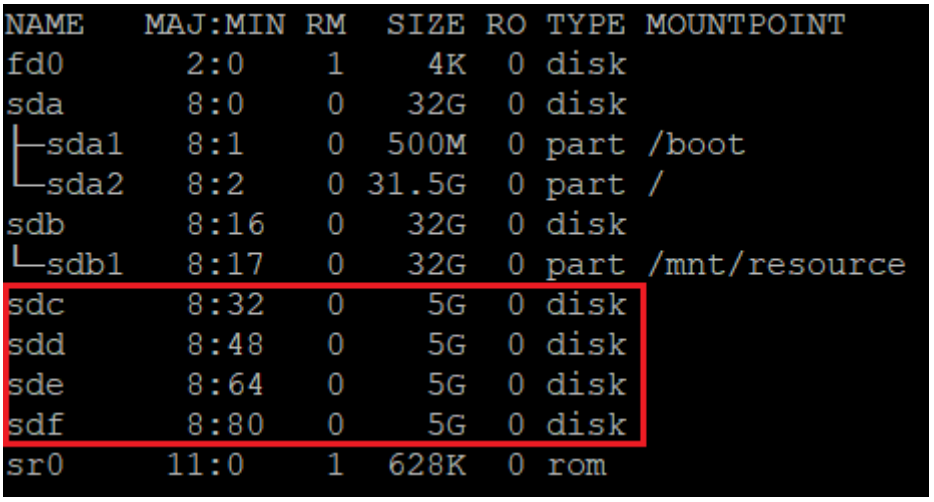
Data disks

LUN	Name	Size	Storage account ...	Encryption	Host caching
0	vmlvmoncrypt_datadisk0	5 GiB	Standard HDD	Not enabled	None
1	vmlvmoncrypt_datadisk1	5 GiB	Standard HDD	Not enabled	None
2	vmlvmoncrypt_datadisk2	5 GiB	Standard HDD	Not enabled	None
3	vmlvmoncrypt_datadisk3	5 GiB	Standard HDD	Not enabled	None

OS:

Bash

```
lsblk
```



Configure the disks to be encrypted

This configuration is done at the operating system level. The corresponding disks are configured for a traditional encryption through Azure Disk Encryption:

- File systems are created on top of the disks.
- Temporary mount points are created to mount the file systems.
- File systems are configured on /etc/fstab to be mounted at boot time.

Check the device letter assigned to the new disks. In this example, we're using four data disks.

Bash

```
lsblk
```

```
NAME    MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
fd0       2:0    1     4K  0 disk
sda       8:0    0    32G  0 disk
├─sda1    8:1    0   500M  0 part /boot
└─sda2    8:2    0 31.5G  0 part /
sdb       8:16   0    32G  0 disk
└─sdb1    8:17   0    32G  0 part /mnt/resource
sdc       8:32   0     5G  0 disk
sdd       8:48   0     5G  0 disk
sde       8:64   0     5G  0 disk
sdf       8:80   0     5G  0 disk
sr0      11:0    1   628K  0 rom
```

Create a file system on top of each disk

This command iterates the creation of an ext4 file system on each disk defined on the "in" part of the "for" cycle.

Bash

```
for disk in c d e f; do echo mkfs.ext4 -F /dev/sd${disk}; done |bash
```

```
[root@vmlvmoncrypt ~]# for disk in c d e; do echo mkfs.ext4 -F /dev/sd${disk}; d
one |bash
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310720 blocks
65536 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Find the universally unique identifier (UUID) of the file systems that you recently created, create a temporary folder, add the corresponding entries on /etc/fstab, and mount all the file systems.

This command also iterates on each disk defined on the "in" part of the "for" cycle:

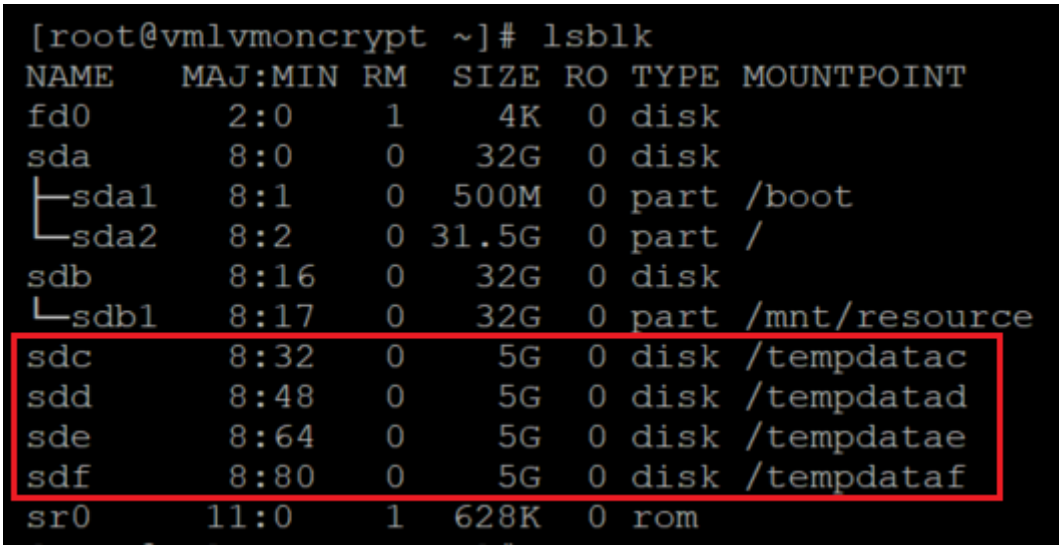
Bash

```
for disk in c d e f; do diskuuid="$(blkid -s UUID -o value /dev/sd${disk})"; \
mkdir /tempdata${disk}; \
echo "UUID=${diskuuid} /tempdata${disk} ext4 defaults,nofail 0 0" >> /etc/fstab; \
mount -a; \
done
```

Verify that the disks are mounted properly

Bash

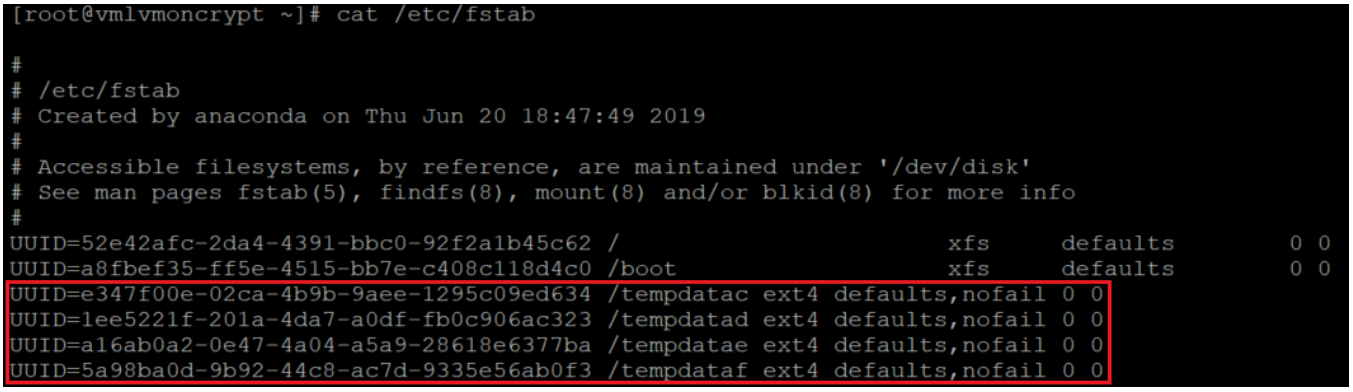
```
lsblk
```



Also verify that the disks are configured:

Bash

```
cat /etc/fstab
```



Encrypt the data disks

PowerShell using a key encryption key (KEK):

PowerShell

```
$sequenceVersion = [Guid]::NewGuid()
Set-AzVMDiskEncryptionExtension -ResourceGroupName $RGNAME `
-VMName ${VMNAME} `
-DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl `
-DiskEncryptionKeyVaultId $KeyVaultResourceId `
-KeyEncryptionKeyUrl $keyEncryptionKeyUrl `
-KeyEncryptionKeyVaultId $KeyVaultResourceId `
-VolumeType 'DATA' `
-EncryptFormatAll `
-SequenceVersion $sequenceVersion `
-skipVmBackup;
```

Azure CLI using a KEK:

Azure CLI

```
az vm encryption enable \
--resource-group ${RGNAME} \
--name ${VMNAME} \
--disk-encryption-keyvault ${KEYVAULTNAME} \
--key-encryption-key ${KEYNAME} \
--key-encryption-keyvault ${KEYVAULTNAME} \
--volume-type "DATA" \
--encrypt-format-all \
-o table
```

Verify the encryption status

Continue to the next step only when all the disks are encrypted.

PowerShell:

PowerShell

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName ${RGNAME} -VMName ${VMNAME}
```

```
OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : EncryptionInProgress
OsVolumeEncryptionSettings : Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : Encryption succeeded for data volumes
```

Azure CLI:

Azure CLI

```
az vm encryption show -n ${VMNAME} -g ${RGNAME} -o table
```

```
user@Azure>az vm encryption show -n ${VMNAME} -g ${RGNAME} -o table
Status                 Message
---------------------  ------------------------------------
Provisioning succeeded Encryption succeeded for data volumes
```

Portal:

Data disks

LUN	Name	Size	Storage account ...	Encryption	Host caching
0	vmlvmoncrypt_datadisk0	5 GiB	Standard HDD	Enabled	None
1	vmlvmoncrypt_datadisk1	5 GiB	Standard HDD	Enabled	None
2	vmlvmoncrypt_datadisk2	5 GiB	Standard HDD	Enabled	None
3	vmlvmoncrypt_datadisk3	5 GiB	Standard HDD	Enabled	None

OS level:

Bash

```
lsblk
```

```
[root@vmlvmoncrypt ~]# lsblk
NAME                                  MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
fd0                                     2:0    1     4K  0 disk
sda                                     8:0    0    32G  0 disk
├─sda1                                  8:1    0   500M  0 part  /boot
└─sda2                                  8:2    0 31.5G  0 part  /
sdb                                     8:16   0    32G  0 disk
└─sdb1                                  8:17   0    32G  0 part
  └─resourceencrypt                   253:4    0    32G  0 crypt /mnt/resource
sdc                                     8:32   0    48M  0 disk
└─sdc1                                  8:33   0    47M  0 part  /mnt/azure_bek_disk
sdd                                     8:48   0     5G  0 disk
└─c49ff535-1df9-45ad-9dad-f0846509f052 253:0    0     5G  0 crypt /tempdatac
sde                                     8:64   0     5G  0 disk
└─6712ad6f-65ce-487b-aa52-462f381611a1 253:1    0     5G  0 crypt /tempdatad
sdf                                     8:80   0     5G  0 disk
└─ea607dfd-c396-48d6-bc54-603cf741bc2a 253:2    0     5G  0 crypt /tempdatae
sdg                                     8:96   0     5G  0 disk
└─4159c60a-a546-455b-985f-92865d51158c 253:3    0     5G  0 crypt /tempdataf
sr0                                    11:0    1  1024M  0 rom
```

The extension will add the file systems to /var/lib/azure_disk_encryption_config/azure_crypt_mount (an old encryption) or to /etc/crypttab (new encryptions).

> ⓘ **Note**
>
> Do not modify any of these files.

This file will take care of activating these disks during the boot process so that LVM or RAID can use them later.

Don't worry about the mount points on this file. Azure Disk Encryption will lose the ability to get the disks mounted as a normal file system after we create a physical volume or a RAID device on top of those encrypted devices. (This will remove the file system format that we used during the preparation process.)

Remove the temporary folders and temporary fstab entries

You unmount the file systems on the disks that will be used as part of LVM.

Bash

```
for disk in c d e f; do umount /tempdata${disk}; done
```

And remove the /etc/fstab entries:

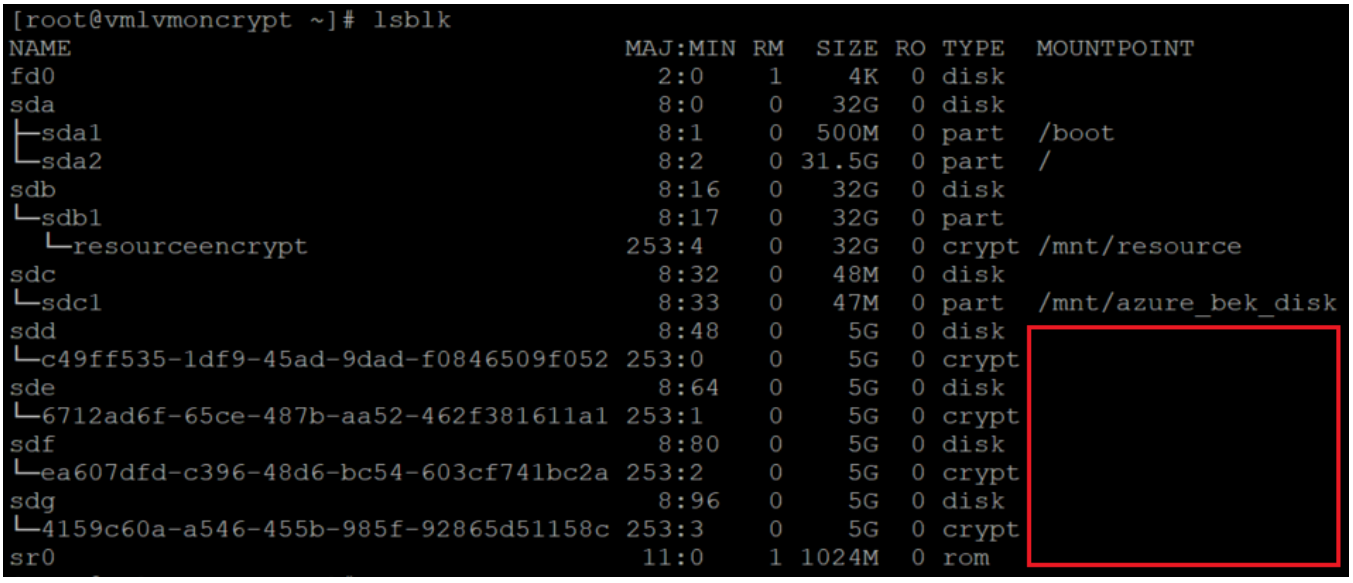
Bash

```
vi /etc/fstab
```

Verify that the disks are not mounted and that the entries on /etc/fstab were removed

Bash

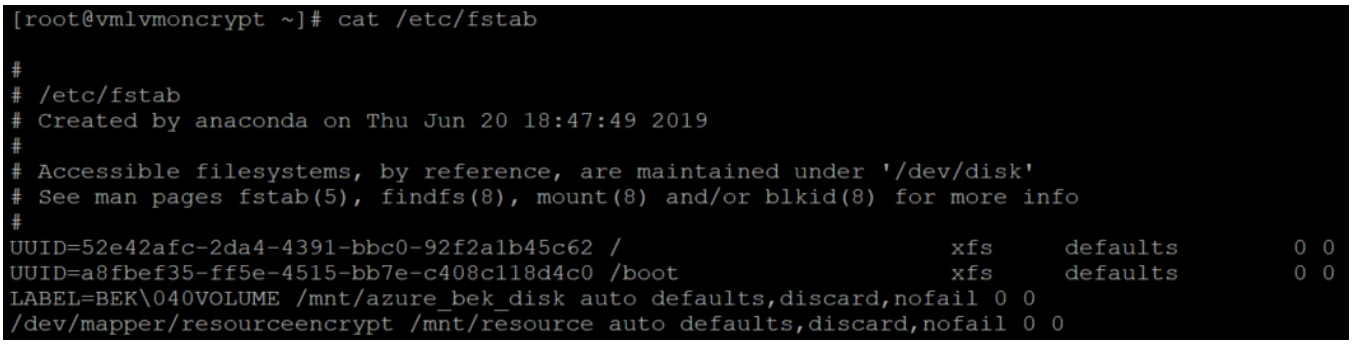
```
lsblk
```



And verify that the disks are configured:

Bash

```
cat /etc/fstab
```



Steps for LVM-on-crypt

Now that the underlying disks are encrypted, you can create the LVM structures.

Instead of using the device name, use the /dev/mapper paths for each of the disks to create a physical volume (on the crypt layer on top of the disk, not on the disk itself).

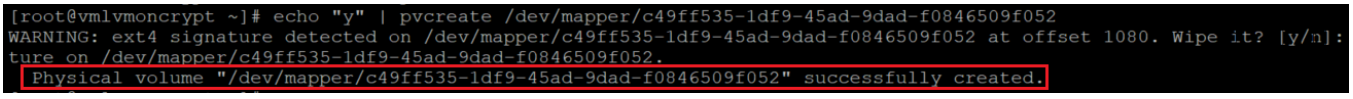
Configure LVM on top of the encrypted layers

Create the physical volumes

You'll get a warning that asks if it's OK to wipe out the file system signature. Continue by entering **y**, or use **echo "y"** as shown:

Bash

```
echo "y" | pvcreate /dev/mapper/c49ff535-1df9-45ad-9dad-f0846509f052
echo "y" | pvcreate /dev/mapper/6712ad6f-65ce-487b-aa52-462f381611a1
echo "y" | pvcreate /dev/mapper/ea607dfd-c396-48d6-bc54-603cf741bc2a
echo "y" | pvcreate /dev/mapper/4159c60a-a546-455b-985f-92865d51158c
```



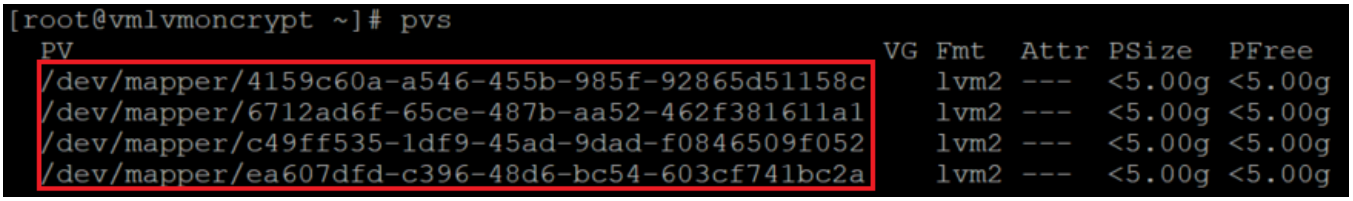
> ⓘ Note

The /dev/mapper/device names here need to be replaced for your actual values based on the output of **lsblk**.

Verify the information for physical volumes

Bash

```
pvs
```



Create the volume group

Create the volume group by using the same devices already initialized:

Bash

```
vgcreate vgdata /dev/mapper/
```

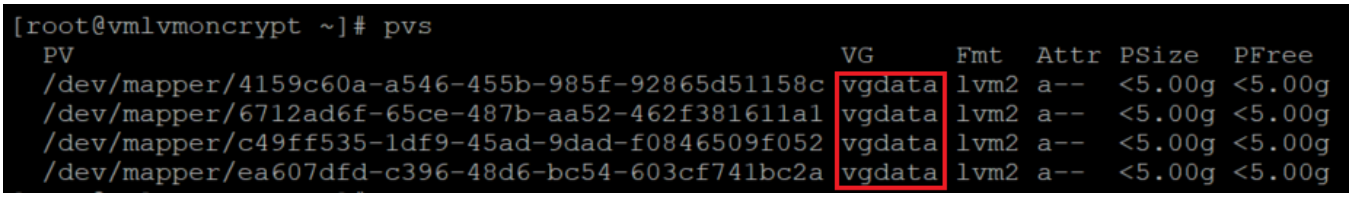
Check the information for the volume group

Bash

```
vgdisplay -v vgdata
```

Bash

```
pvs
```



Create logical volumes

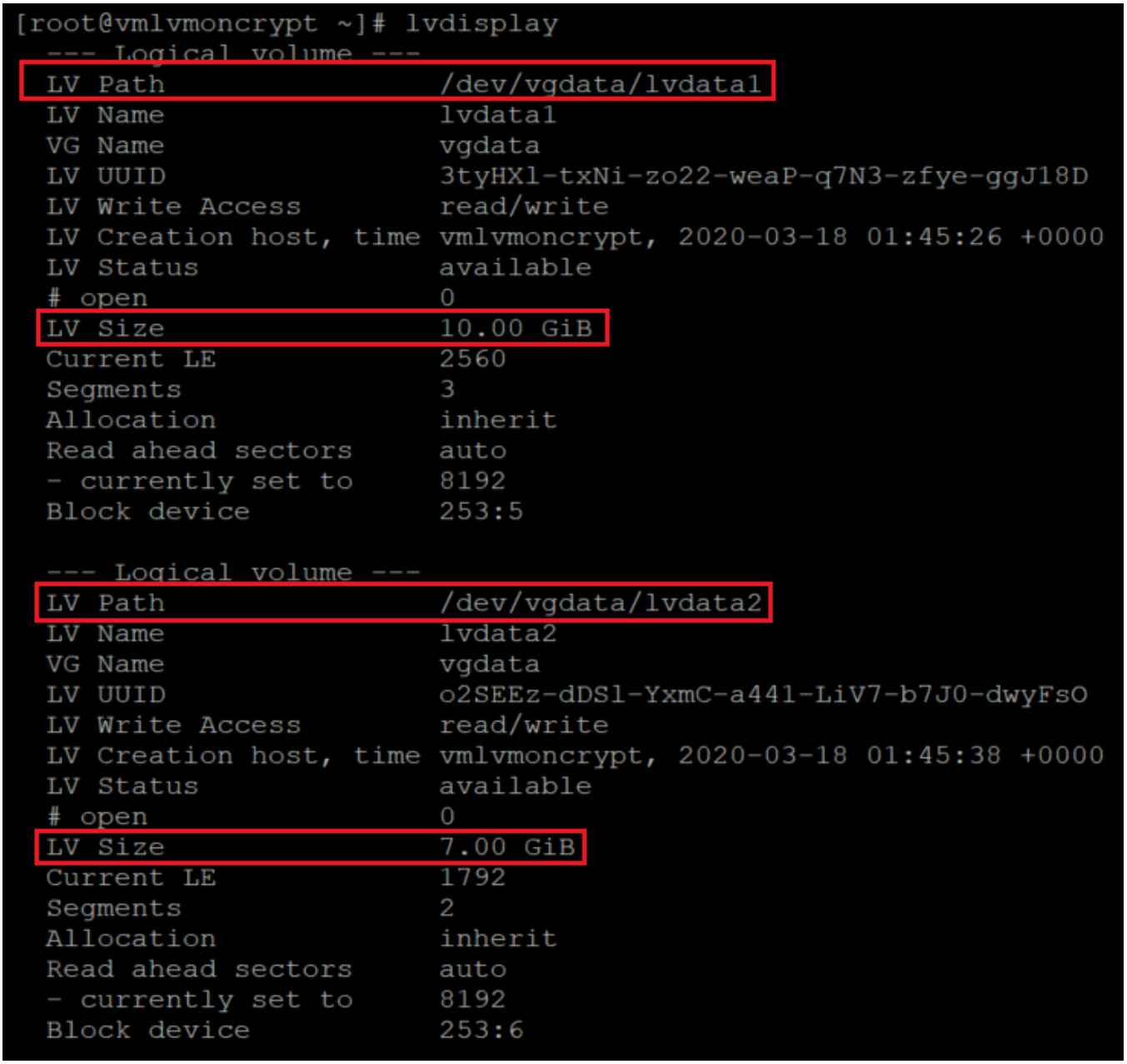
Bash

```
lvcreate -L 10G -n lvdata1 vgdata
lvcreate -L 7G -n lvdata2 vgdata
```

Check the created logical volumes

Bash

```
lvdisplay
lvdisplay vgdata/lvdata1
lvdisplay vgdata/lvdata2
```



Create file systems on top of the structures for logical volumes

Bash

```
echo "yes" | mkfs.ext4 /dev/vgdata/lvdata1
echo "yes" | mkfs.ext4 /dev/vgdata/lvdata2
```

Create the mount points for the new file systems

Bash

```
mkdir /data0
mkdir /data1
```

Add the new file systems to /etc/fstab and mount them

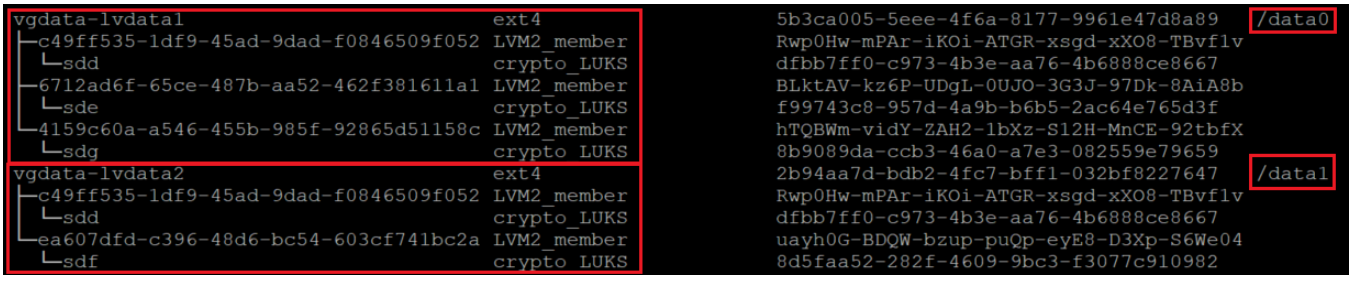
Bash

```
echo "/dev/mapper/vgdata-lvdata1 /data0 ext4 defaults,nofail 0 0" >>/etc/fstab
echo "/dev/mapper/vgdata-lvdata2 /data1 ext4 defaults,nofail 0 0" >>/etc/fstab
mount -a
```

Verify that the new file systems are mounted

Bash

```
lsblk -fs
df -h
```



On this variation of **lsblk**, we're listing the devices showing the dependencies in reverse order. This option helps to identify the devices grouped by the logical volume instead of the original /dev/sd[disk] device names.

It's important to make sure that the **nofail** option is added to the mount point options of the LVM volumes created on top of a device encrypted through Azure Disk Encryption. It prevents the OS from getting stuck during the boot process (or in maintenance mode).

If you don't use the **nofail** option:

- The OS will never get into the stage where Azure Disk Encryption is started and the data disks are unlocked and mounted.
- The encrypted disks will be unlocked at the end of the boot process. The LVM volumes and file systems will be automatically mounted until Azure Disk Encryption unlocks them.

You can test rebooting the VM and validate that the file systems are also automatically getting mounted after boot time. This process might take several minutes, depending on the number and sizes of file systems.

Reboot the VM and verify after reboot

Bash

```
shutdown -r now
```

Bash

```
lsblk
df -h
```

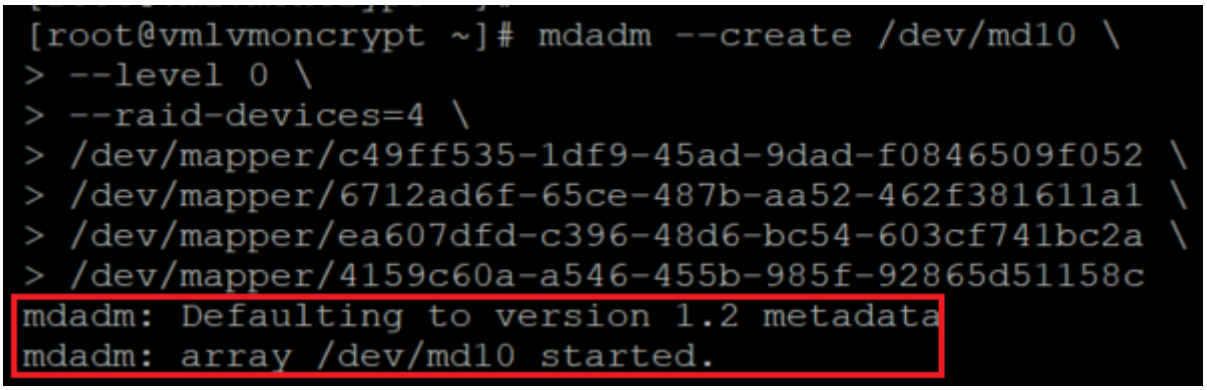
Steps for RAID-on-crypt

Now that the underlying disks are encrypted, you can continue to create the RAID structures. The process is the same as the one for LVM, but instead of using the device name, use the /dev/mapper paths for each disk.

Configure RAID on top of the encrypted layer of the disks

Bash

```
mdadm --create /dev/md10 \
--level 0 \
--raid-devices=4 \
/dev/mapper/c49ff535-1df9-45ad-9dad-f0846509f052 \
/dev/mapper/6712ad6f-65ce-487b-aa52-462f381611a1 \
/dev/mapper/ea607dfd-c396-48d6-bc54-603cf741bc2a \
/dev/mapper/4159c60a-a546-455b-985f-92865d51158c
```

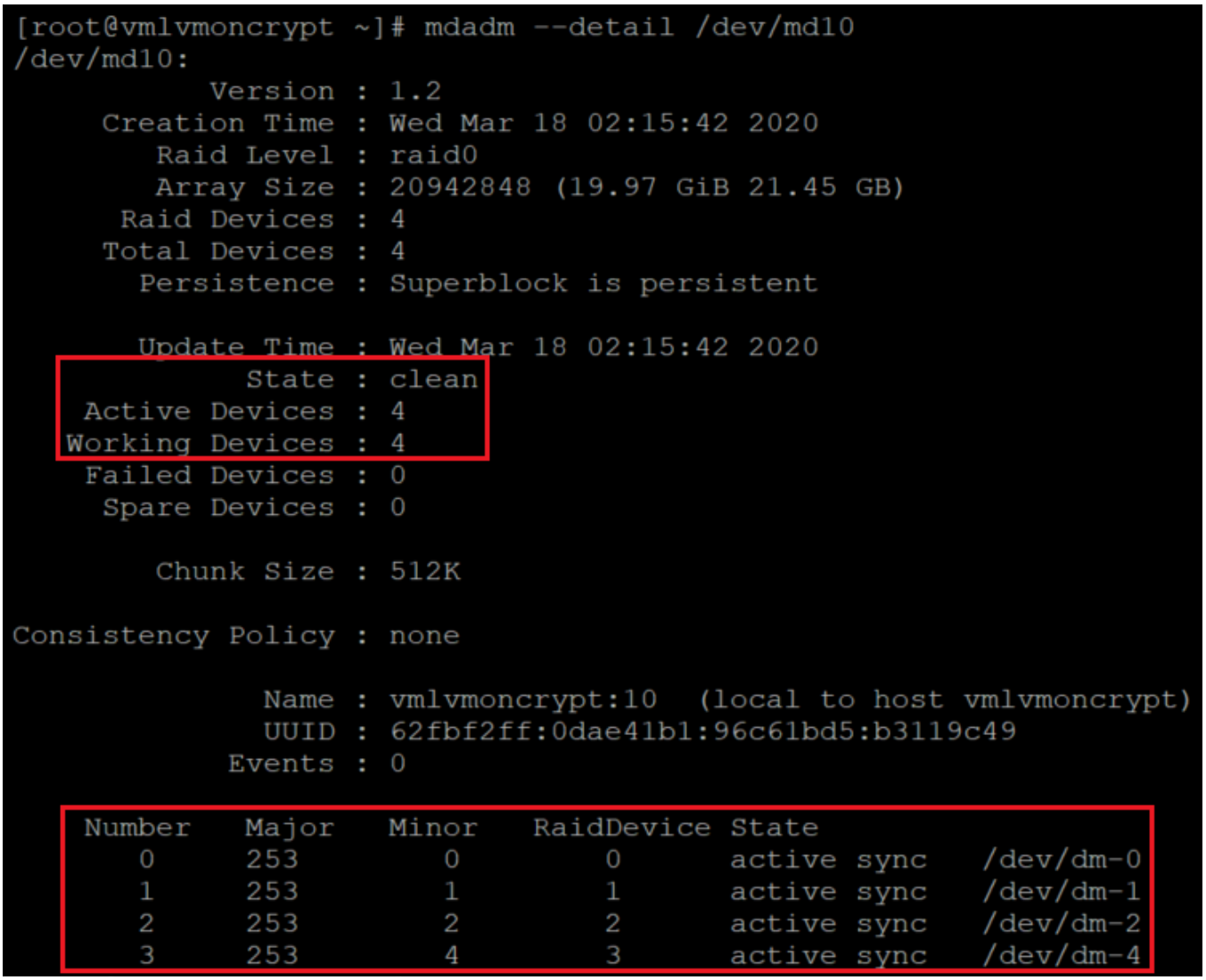


> ⓘ Note
>
> The /dev/mapper/device names here need to be replaced with your actual values, based on the output of **lsblk**.

Check/monitor RAID creation

Bash

```
watch -n1 cat /proc/mdstat
mdadm --examine /dev/mapper/[]
mdadm --detail /dev/md10
```



Create a file system on top of the new RAID device

Bash

```
mkfs.ext4 /dev/md10
```

Create a new mount point for the file system, add the new file system to /etc/fstab, and mount it:

> ⓘ **Note**
>
> This cycle iterates only on one device for this particular example, is built this way to be used for multiple md devices if needed.

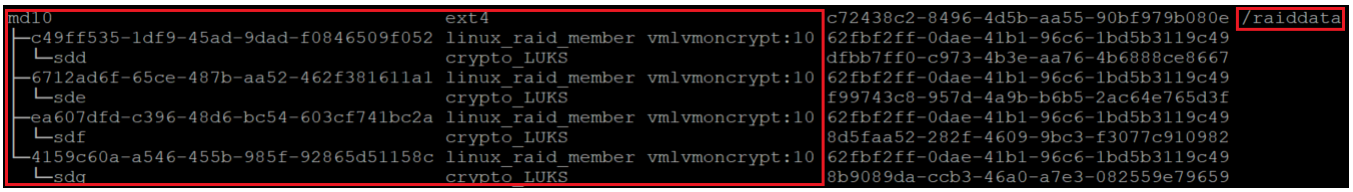
Bash

```
for device in md10; do diskuuid="$(blkid -s UUID -o value /dev/${device})"; \
mkdir /raiddata; \
echo "UUID=${diskuuid} /raiddata ext4 defaults,nofail 0 0" >> /etc/fstab; \
```

```
mount -a; \
done
```

Verify that the new file system is mounted:

```Bash
lsblk -fs
df -h
```



It's important to make sure that the **nofail** option is added to the mount point options of the RAID volumes created on top of a device encrypted through Azure Disk Encryption. It prevents the OS from getting stuck during the boot process (or in maintenance mode).

If you don't use the **nofail** option:

- The OS will never get into the stage where Azure Disk Encryption is started and the data disks are unlocked and mounted.
- The encrypted disks will be unlocked at the end of the boot process. The RAID volumes and file systems will be automatically mounted until Azure Disk Encryption unlocks them.

You can test rebooting the VM and validate that the file systems are also automatically getting mounted after boot time. This process might take several minutes, depending on the number and sizes of file systems.

```Bash
shutdown -r now
```

And when you can log in:

```Bash
lsblk
df -h
```

Next steps

- Resize logical volume management devices encrypted with Azure Disk Encryption
- Azure Disk Encryption troubleshooting

How to resize logical volume management devices that use Azure Disk Encryption

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

In this article, you'll learn how to resize data disks that use Azure Disk Encryption. To resize the disks, you'll use logical volume management (LVM) on Linux. The steps apply to multiple scenarios.

You can use this resizing process in the following environments:

- Linux distributions:
  - Red Hat Enterprise Linux (RHEL) 7 or later
  - Ubuntu 18.04 or later
  - SUSE 12 or later
- Azure Disk Encryption versions:
  - Single-pass extension
  - Dual-pass extension

Prerequisites

This article assumes that you have:

- An existing LVM configuration. For more information, see Configure LVM on a Linux VM.
- Disks that are already encrypted by Azure Disk Encryption. For more information, see Configure LVM and RAID on encrypted devices.
- Experience using Linux and LVM.
- Experience using */dev/disk/scsi1/* paths for data disks on Azure. For more information, see Troubleshoot Linux VM device name problems.

Scenarios

The procedures in this article apply to the following scenarios:

- Traditional LVM and LVM-on-crypt configurations
- Traditional LVM encryption
- LVM-on-crypt
- Data disks only. OS disk resizing is not supported.

Traditional LVM and LVM-on-crypt configurations

Traditional LVM and LVM-on-crypt configurations extend a logical volume (LV) when the volume group (VG) has available space.

Traditional LVM encryption

In traditional LVM encryption, LVs are encrypted. The whole disk isn't encrypted.

By using traditional LVM encryption, you can:

- Extend the LV when you add a new physical volume (PV).
- Extend the LV when you resize an existing PV.

LVM-on-crypt

The recommended method for disk encryption is LVM-on-encrypt. This method encrypts the entire disk, not just the LV.

By using LVM-on-crypt, you can:

- Extend the LV when you add a new PV.
- Extend the LV when you resize an existing PV.

> ⓘ **Note**
>
> We don't recommend mixing traditional LVM encryption and LVM-on-crypt on the same VM.

The following sections provide examples of how to use LVM and LVM-on-crypt. The examples use preexisting values for disks, PVs, VGs, LVs, file systems, universally unique identifiers (UUIDs), and mount points. Replace these values with your own values to fit your environment.

Extend an LV when the VG has available space

The traditional way to resize LVs is to extend an LV when the VG has space available. You can use this method for nonencrypted disks, traditional LVM-encrypted volumes, and LVM-on-crypt configurations.

1. Verify the current size of the file system that you want to increase:

   ```
   Bash

   df -h /mountpoint
   ```

   ```
   [root@azurevm]# df -h /datalv02
   Filesystem                                      Size  Used Avail Use% Mounted on
   /dev/mapper/78b259a1-956b-4f3a-8e6b-c05bf5040bc5  974M  958M     0 100% /datalv02
   ```

2. Verify that the VG has enough space to increase the LV:

   ```
   Bash

   sudo vgs
   ```

   ```
   [root@azurevm]#  vgs
     VG     #PV #LV #SN Attr   VSize   VFree
     datavg   3   4   0 wz--n- <14.99g <4.99g
   ```

   You can also use `vgdisplay`:

   ```
   Bash

   sudo vgdisplay vgname
   ```

```
[root@azurevm]# vgdisplay datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  5
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                4
  Open LV               4
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               <14.99 GiB
  PE Size               4.00 MiB
  Total PE              3837
  Alloc PE / Size       2560 / 10.00 GiB
  Free  PE / Size       1277 / <4.99 GiB
  VG UUID               jvc1Kz-FXLd-dNjC-14q7-k85I-Pp8z-bMqFLY
```

3. Identify which LV needs to be resized:

Bash

```
sudo lsblk
```

```
[root@azurevm]#  lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part  /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   46M  0 part  /mnt/azure_bek_disk
sdd                                       8:48   0    5G  0 disk
├─datavg-datalv01                       253:0    0    1G  0 lvm
│ └─55b4afd2-a160-426b-9bd7-588af6c46e9b 253:4    0 1022M  0 crypt /datalv01
├─datavg-datalv02                       253:1    0    1G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0 1022M  0 crypt /datalv02
└─datavg-datalv03                       253:2    0    1G  0 lvm
  └─70abfc58-b0fd-441b-8b77-f86c9249af5e 253:6    0 1022M  0 crypt /datalv03
sde                                       8:64   0    5G  0 disk
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sdf                                       8:80   0    5G  0 disk
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sr0                                      11:0    1 1024M  0 rom
```

For LVM-on-crypt, the difference is that this output shows that the encrypted layer is at the disk level.

```
[root@vmresizelv ~]# lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part
  └─resourceencrypt                     253:0    0   32G  0 crypt /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   47M  0 part  /mnt/azure_bek_disk
sde                                       8:64   0    5G  0 disk
└─e19fcd77-e974-4e5c-a874-e78e4b6d2f48  253:1    0    5G  0 crypt
  ├─datavg-datalv01                     253:4    0    1G  0 lvm   /datalv01
  ├─datavg-datalv02                     253:5    0    6G  0 lvm   /datalv02
  ├─datavg-datalv03                     253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                     253:7    0    3G  0 lvm   /datalv04
sdf                                       8:80   0    5G  0 disk
└─49de5df0-1c65-48bf-809c-588805eda921  253:2    0    5G  0 crypt
  ├─datavg-datalv02                     253:5    0    6G  0 lvm   /datalv02
  └─datavg-datalv04                     253:7    0    3G  0 lvm   /datalv04
sdg                                       8:96   0    5G  0 disk
└─8a33198f-fa7e-491f-a9c6-e9d0a1d25d57  253:3    0    5G  0 crypt
  ├─datavg-datalv03                     253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                     253:7    0    3G  0 lvm   /datalv04
```

4. Check the LV size:

Bash

```
sudo lvdisplay lvname
```

```
[root@azurevm]#  lvdisplay /dev/datavg/datalv02
  --- Logical volume ---
  LV Path                /dev/datavg/datalv02
  LV Name                datalv02
  VG Name                datavg
  LV UUID                Isacy4-ukYq-UKCB-tG48-tjfc-LW1r-kWTyfy
  LV Write Access        read/write
  LV Creation host, time vmresizelv, 2020-09-18 20:13:14 +0000
  LV Status              available
  # open                 1
  LV Size                1.00 GiB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:1
```

5. Increase the LV size by using `-r` to resize the file system online:

Bash

```
sudo lvextend -r -L +2G /dev/vgname/lvname
```

```
[root@azurevm]# lvextend -r -L +2G /dev/datavg/datalv02
  Size of logical volume datavg/datalv02 changed from 1.00 GiB (256 extents) to 3.00 GiB (768 extents).
  Logical volume datavg/datalv02 successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/78b259a1-956b-4f3a-8e6b-c05bf5040bc5 is mounted on /datalv02; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mapper/78b259a1-956b-4f3a-8e6b-c05bf5040bc5 is now 785920 blocks long.
```

6. Verify the new sizes for the LV and the file system:

```
Bash

df -h /mountpoint
```

```
[root@azurevm]# df -h /datalv02/
Filesystem                                      Size  Used Avail Use% Mounted on
/dev/mapper/78b259a1-956b-4f3a-8e6b-c05bf5040bc5  3.0G  959M  1.9G  34% /datalv02
```

The size output indicates that the LV and file system were successfully resized.

You can check the LV information again to confirm the changes at the level of the LV:

```
Bash

sudo lvdisplay lvname
```

```
[root@azurevm]#  lvdisplay /dev/datavg/datalv02
  --- Logical volume ---
  LV Path                /dev/datavg/datalv02
  LV Name                datalv02
  VG Name                datavg
  LV UUID                Isacy4-ukYq-UKCB-tG48-tjfc-LW1r-kWTyfy
  LV Write Access        read/write
  LV Creation host, time vmresizelv, 2020-09-18 20:13:14 +0000
  LV Status              available
  # open                 1
  LV Size                3.00 GiB
  Current LE             768
  Segments               3
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:1
```

Extend a traditional LVM volume by adding a new PV

When you need to add a new disk to increase the VG size, extend your traditional LVM volume by adding a new PV.

1. Verify the current size of the file system that you want to increase:

```
Bash
```

```
df -h /mountpoint
```

```
[root@azurevm]# df -h /datalv01
Filesystem                                      Size  Used Avail Use% Mounted on
/dev/mapper/55b4afd2-a160-426b-9bd7-588af6c46e9b  974M  958M     0 100% /datalv01
```

2. Verify the current PV configuration:

Bash

```
sudo pvs
```

```
[root@azurevm]# pvs
  PV         VG     Fmt  Attr PSize  PFree
  /dev/sdd   datavg lvm2 a--  <5.00g     0
  /dev/sde   datavg lvm2 a--  <5.00g     0
  /dev/sdf   datavg lvm2 a--  <5.00g     0
```

3. Check the current VG information:

Bash

```
sudo vgs
```

```
[root@azurevm]# vgs
  VG     #PV #LV #SN Attr   VSize   VFree
  datavg   3   4   0 wz--n- <14.99g    0
```

4. Check the current disk list. Identify data disks by checking the devices in */dev/disk/azure/scsi1/*.

Bash

```
sudo ls -l /dev/disk/azure/scsi1/
```

```
[root@azurevm]# ls -l /dev/disk/azure/scsi1/
total 0
lrwxrwxrwx. 1 root root 12 Sep 18 21:43 lun0 -> ../../../sdd
lrwxrwxrwx. 1 root root 12 Sep 18 21:43 lun1 -> ../../../sde
lrwxrwxrwx. 1 root root 12 Sep 18 21:43 lun2 -> ../../../sdf
```

5. Check the output of `lsblk`:

Bash

```
sudo lsbk
```

```
[root@azurevm]#  lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part  /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   46M  0 part  /mnt/azure_bek_disk
sdd                                       8:48   0    5G  0 disk
├─datavg-datalv01                       253:0    0    1G  0 lvm
│ └─55b4afd2-a160-426b-9bd7-588af6c46e9b 253:4    0 1022M  0 crypt /datalv01
├─datavg-datalv02                       253:1    0    6G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0    6G  0 crypt /datalv02
└─datavg-datalv03                       253:2    0    1G  0 lvm
  └─70abfc58-b0fd-441b-8b77-f86c9249af5e 253:6    0 1022M  0 crypt /datalv03
sde                                       8:64   0    5G  0 disk
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sdf                                       8:80   0    5G  0 disk
├─datavg-datalv02                       253:1    0    6G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0    6G  0 crypt /datalv02
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sr0                                      11:0    1 1024M  0 rom
```

6. Attach the new disk to the VM by following the instructions in Attach a data disk to a Linux VM.

7. Check the disk list, and notice the new disk.

Bash

```
sudo ls -l /dev/disk/azure/scsi1/
```

```
[root@azurevm]#  ls -l /dev/disk/azure/scsi1/
total 0
lrwxrwxrwx. 1 root root 12 Sep 18 21:43 lun0 -> ../../../sdd
lrwxrwxrwx. 1 root root 12 Sep 18 21:43 lun1 -> ../../../sde
lrwxrwxrwx. 1 root root 12 Sep 18 21:43 lun2 -> ../../../sdf
lrwxrwxrwx. 1 root root 12 Sep 18 22:20 lun3 -> ../../../sdg
```

Bash

```
sudo lsblk
```

```
[root@azurevm]# lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part  /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   46M  0 part  /mnt/azure_bek_disk
sdd                                       8:48   0    5G  0 disk
├─datavg-datalv01                       253:0    0    1G  0 lvm
│ └─55b4afd2-a160-426b-9bd7-588af6c46e9b 253:4    0 1022M  0 crypt /datalv01
├─datavg-datalv02                       253:1    0    6G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0    6G  0 crypt /datalv02
└─datavg-datalv03                       253:2    0    1G  0 lvm
  └─70abfc58-b0fd-441b-8b77-f86c9249af5e 253:6    0 1022M  0 crypt /datalv03
sde                                       8:64   0    5G  0 disk
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sdf                                       8:80   0    5G  0 disk
├─datavg-datalv02                       253:1    0    6G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0    6G  0 crypt /datalv02
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sdg                                       8:96   0    5G  0 disk
sr0                                      11:0    1 1024M  0 rom
```

8. Create a new PV on top of the new data disk:

```
Bash

sudo pvcreate /dev/newdisk
```

```
[root@azurevm]#  pvcreate /dev/sdg
  Physical volume "/dev/sdg" successfully created.
[root@azurevm]#
```

This method uses the whole disk as a PV without a partition. Alternatively, you can use `fdisk` to create a partition and then use that partition for `pvcreate`.

9. Verify that the PV was added to the PV list:

```
Bash

sudo pvs
```

```
[root@azurevm]#  pvs
  PV         VG     Fmt  Attr PSize  PFree
  /dev/sdd   datavg lvm2 a--  <5.00g     0
  /dev/sde   datavg lvm2 a--  <5.00g     0
  /dev/sdf   datavg lvm2 a--  <5.00g     0
  /dev/sdg          lvm2 ---   5.00g 5.00g
```

10. Extend the VG by adding the new PV to it:

```
```

Bash

```
sudo vgextend vgname /dev/newdisk
```

```
[root@azurevm]#  vgextend datavg /dev/sdg
  Volume group "datavg" successfully extended
```

11. Check the new VG size:

Bash

```
sudo vgs
```

```
[root@azurevm]#  vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  datavg    4   4   0 wz--n- 19.98g <5.00g
```

13. Use `lsblk` to identify the LV that needs to be resized:

Bash

```
sudo lsblk
```

```
[root@azurevm]#  lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part  /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   46M  0 part  /mnt/azure_bek_disk
sdd                                       8:48   0    5G  0 disk
├─datavg-datalv01                       253:0    0    1G  0 lvm
│ └─55b4afd2-a160-426b-9bd7-588af6c46e9b 253:4    0 1022M  0 crypt /datalv01
├─datavg-datalv02                       253:1    0    6G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0    6G  0 crypt /datalv02
└─datavg-datalv03                       253:2    0    1G  0 lvm
  └─70abfc58-b0fd-441b-8b77-f86c9249af5e 253:6    0 1022M  0 crypt /datalv03
sde                                       8:64   0    5G  0 disk
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sdf                                       8:80   0    5G  0 disk
├─datavg-datalv02                       253:1    0    6G  0 lvm
│ └─78b259a1-956b-4f3a-8e6b-c05bf5040bc5 253:5    0    6G  0 crypt /datalv02
└─datavg-datalv04                       253:3    0    7G  0 lvm
  └─e5dee6bc-0f22-4579-b6c0-6103a9aba86c 253:7    0    7G  0 crypt /datalv04
sdg                                       8:96   0    5G  0 disk
```

14. Extend the LV size by using `-r` to increase the file system online:

Bash

```
sudo lvextend -r -L +2G /dev/vgname/lvname
```

```
[root@azurevm]#  lvextend -r -L +2G /dev/datavg/datalv01
  Size of logical volume datavg/datalv01 changed from 1.00 GiB (256 extents) to 3.00 GiB (768 extents).
  Logical volume datavg/datalv01 successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/55b4afd2-a160-426b-9bd7-588af6c46e9b is mounted on /datalv01; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mapper/55b4afd2-a160-426b-9bd7-588af6c46e9b is now 785920 blocks long.
```

15. Verify the new sizes of the LV and file system:

```
Bash

df -h /mountpoint
```

```
[root@azurevm]# df -h /datalv01
Filesystem                                      Size  Used Avail Use% Mounted on
/dev/mapper/55b4afd2-a160-426b-9bd7-588af6c46e9b  3.0G  959M  1.9G  34% /datalv01
```

> **ⓘ Important**
>
> When Azure Data Encryption is used on traditional LVM configurations, the encrypted layer is created at the LV level, not at the disk level.
>
> At this point, the encrypted layer is expanded to the new disk. The actual data disk has no encryption settings at the platform level, so its encryption status isn't updated.
>
> These are some of the reasons why LVM-on-crypt is the recommended approach.

16. Check the encryption information from the portal:

LUN ⓘ	Disk name	Storage type	Size (GiB)	Max IOPS	Max throughput (...	Encryption ⓘ
0	vmresizelv_datadisk0	Standard HDD	5	500	60	SSE with PMK & ADE
1	vmresizelv_datadisk1	Standard HDD	5	500	60	SSE with PMK & ADE
2	vmresizelv_datadisk2	Standard HDD	5	500	60	SSE with PMK & ADE
3	vmresizelv_datadisk3	Standard HDD	5	500	60	SSE with PMK

To update the encryption settings on the disk, add a new LV and enable the extension on the VM.

17. Add a new LV, create a file system on it, and add it to `/etc/fstab`.

18. Set the encryption extension again. This time you'll stamp the encryption settings on the new data disk at the platform level. Here's a CLI example:

```
azurecliazurecli

az vm encryption enable -g ${RGNAME} --name ${VMNAME} --disk-encryption-
keyvault "<your-unique-keyvault-name>"
```

19. Check the encryption information from the portal:



LUN	Disk name	Storage type	Size (GiB)	Max IOPS	Max throughput (MBps)	Encryption
0	vmresizelv_datadisk0	Standard HDD	5	500	60	SSE with PMK & ADE
1	vmresizelv_datadisk1	Standard HDD	5	500	60	SSE with PMK & ADE
2	vmresizelv_datadisk2	Standard HDD	5	500	60	SSE with PMK & ADE
3	vmresizelv_datadisk3	Standard HDD	5	500	60	SSE with PMK & ADE

After the encryption settings are updated, you can delete the new LV. Also delete the entry from the `/etc/fstab` and `/etc/crypttab` that you created.

```
[root@azurevm]#  cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Jun 20 18:47:49 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=52e42afc-2da4-4391-bbc0-92f2a1b45c62 /                       xfs     defaults        0 0
UUID=a8fbef35-ff5e-4515-bb7e-c408c118d4c0 /boot                   xfs     defaults        0 0
/dev/mapper/55b4afd2-a160-426b-9bd7-588af6c46e9b /datalv01 ext4 defaults,nofail 0 0
/dev/mapper/78b259a1-956b-4f3a-8e6b-c05bf5040bc5 /datalv02 ext4 defaults,nofail 0 0
/dev/mapper/70abfc58-b0fd-441b-8b77-f86c9249af5e /datalv03 ext4 defaults,nofail 0 0
/dev/mapper/e5dee6bc-0f22-4579-b6c0-6103a9aba86c /datalv04 ext4 defaults,nofail 0 0
/dev/mapper/9157c7b5-8bad-4a84-84e2-0ca2bffcfce4 /datalv05 ext4 defaults,nofail 0 0
LABEL=BEK\040VOLUME /mnt/azure_bek_disk auto defaults,discard,nofail 0 0
[root@azurevm]#
[root@azurevm]#
[root@azurevm]#  cat /etc/crypttab

55b4afd2-a160-426b-9bd7-588af6c46e9b /dev/mapper/datavg-datalv01 /mnt/azure_bek_disk/LinuxPassPhraseFileName_1_0 luks,nofail

78b259a1-956b-4f3a-8e6b-c05bf5040bc5 /dev/mapper/datavg-datalv02 /mnt/azure_bek_disk/LinuxPassPhraseFileName_1_0 luks,nofail

70abfc58-b0fd-441b-8b77-f86c9249af5e /dev/mapper/datavg-datalv03 /mnt/azure_bek_disk/LinuxPassPhraseFileName_1_0 luks,nofail

e5dee6bc-0f22-4579-b6c0-6103a9aba86c /dev/mapper/datavg-datalv04 /mnt/azure_bek_disk/LinuxPassPhraseFileName_1_2 luks,nofail

9157c7b5-8bad-4a84-84e2-0ca2bffcfce4 /dev/mapper/datavg-datalv05 /mnt/azure_bek_disk/LinuxPassPhraseFileName_1_3 luks,nofail
[root@azurevm]#
```

Follow these steps to finish cleaning up:

1. Unmount the LV:

   ```
   Bash

   sudo umount /mountpoint
   ```

2. Close the encrypted layer of the volume:

   ```
   Bash

   sudo cryptsetup luksClose /dev/vgname/lvname
   ```

3. Delete the LV:

   ```
   Bash

   sudo lvremove /dev/vgname/lvname
   ```

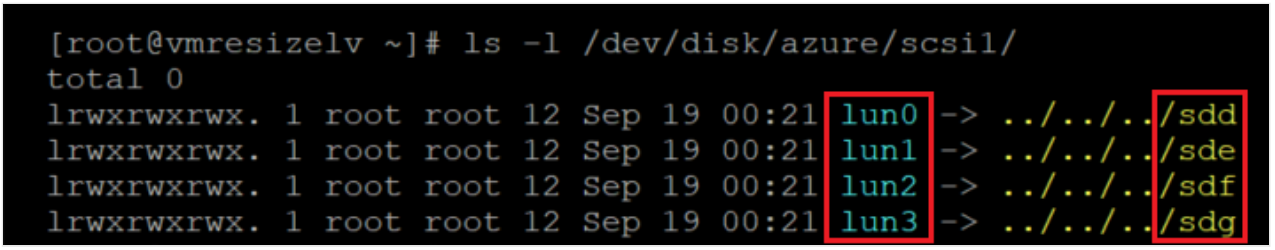
Extend a traditional LVM volume by resizing an existing PV

Im some scenarios, your limitations might require you to resize an existing disk. Here's how:

1. Identify your encrypted disks:

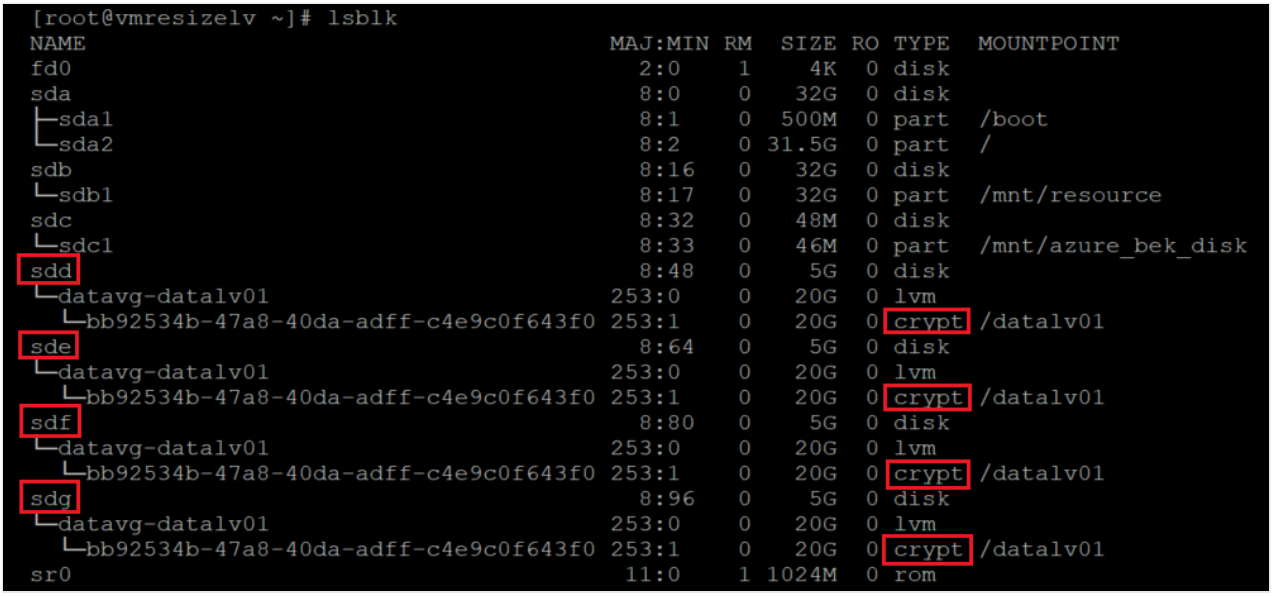
Bash

```
sudo ls -l /dev/disk/azure/scsi1/
```



Bash

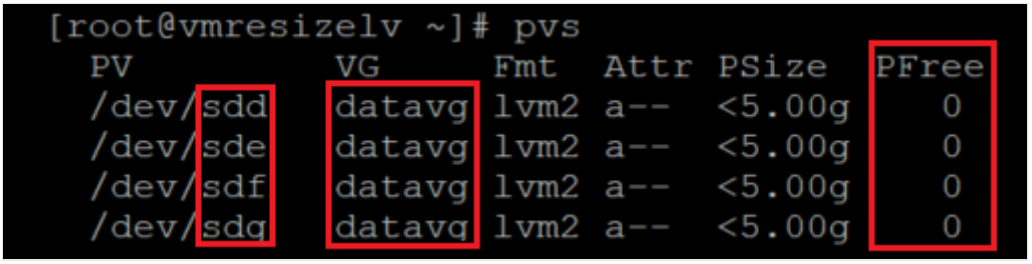
```
sudo lsblk -fs
```



2. Check the PV information:

Bash

```
sudo pvs
```



The results in the image show that all of the space on all of the PVs is currently used.

3. Check the VG information:

```
Bash

sudo vgs
sudo vgdisplay -v vgname
```

```
[root@vmresizelv ~]# vgs
  VG     #PV #LV #SN Attr   VSize  VFree
  datavg   4   1   0 wz--n- 19.98g    0
[root@vmresizelv ~]# vgdisplay datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        4
  Metadata Sequence No  16
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                4
  Act PV                4
  VG Size               19.98 GiB
  PE Size               4.00 MiB
  Total PE              5116
  Alloc PE / Size       5116 / 19.98 GiB
  Free  PE / Size       0 / 0
  VG UUID               jvc1Kz-FXLd-dNjC-14q7-k85I-Pp8z-bMqFLY
```

4. Check the disk sizes. You can use `fdisk` or `lsblk` to list the drive sizes.

```
Bash

for disk in `sudo ls -l /dev/disk/azure/scsi1/* | awk -F/ '{print $NF}'` ; do
echo "sudo fdisk -l /dev/${disk} | grep ^Disk "; done | bash

sudo lsblk -o "NAME,SIZE"
```

```
Disk /dev/sdd: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sde: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdf: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdg: 5368 MB, 5368709120 bytes, 10485760 sectors
```

Here we identified which PVs are associated with which LVs by using `lsblk -fs`. You can identify the associations by running `lvdisplay`.

Bash

```
sudo lvdisplay --maps VG/LV
sudo lvdisplay --maps datavg/datalv1
```

```
[root@vmresizelv ~]# lvdisplay --maps datavg/datalv01
  --- Logical volume ---
  LV Path                /dev/datavg/datalv01
  LV Name                datalv01
  VG Name                datavg
  LV UUID                oiTd6A-PTqY-AbzV-Kd0J-0891-dlMP-jBUJUC
  LV Write Access        read/write
  LV Creation host, time vmresizelv, 2020-09-19 00:21:25 +0000
  LV Status              available
  # open                 1
  LV Size                19.98 GiB
  Current LE             5116
  Segments               4
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:0

  --- Segments ---
  Logical extents 0 to 1278:
    Type                linear
    Physical volume     /dev/sdd
    Physical extents    0 to 1278

  Logical extents 1279 to 2557:
    Type                linear
    Physical volume     /dev/sde
    Physical extents    0 to 1278

  Logical extents 2558 to 3836:
    Type                linear
    Physical volume     /dev/sdf
    Physical extents    0 to 1278

  Logical extents 3837 to 5115:
    Type                linear
    Physical volume     /dev/sdg
    Physical extents    0 to 1278
```

In this case, all four data drives are part of the same VG and a single LV. Your configuration might differ.

5. Check the current file system utilization:

Bash

```
df -h /datalvm*
```

```
[root@vmresizelv ~]# df -h /datalv01/
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/bb92534b-47a8-40da-adff-c4e9c0f643f0   20G   20G     0 100% /datalv01
```

6. Resize the data disks by following the instructions in Expand an Azure managed disk. You can use the portal, the CLI, or PowerShell.

> **ⓘ Important**
>
> Some data disks on Linux VMs can be resized without Deallocating the VM, please check [Expand virtual hard disks on a Linux VM](/azure/virtual-machines/linux/expand-disks? tabs=ubuntu#expand-an-azure-managed-disk) in order to verify your disks meet the requirements.

7. Start the VM and check the new sizes by using `fdisk`.

Bash

```
for disk in `sudo ls -l /dev/disk/azure/scsi1/* | awk -F/ '{print $NF}'` ; do
echo "sudo fdisk -l /dev/${disk} | grep ^Disk "; done | bash

sudo lsblk -o "NAME,SIZE"
```

```
Disk /dev/sdd: 21.5 GB, 21474836480 bytes, 41943040 sectors
Disk /dev/sde: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdf: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdg: 5368 MB, 5368709120 bytes, 10485760 sectors
```

In this case, `/dev/sdd` was resized from 5 G to 20 G.

8. Check the current PV size:

Bash

```
sudo pvdisplay /dev/resizeddisk
```

```
[root@vmresizelv ~]# pvdisplay /dev/sdd
  --- Physical volume ---
  PV Name               /dev/sdd
  VG Name               datavg
  PV Size               5.00 GiB / not usable 4.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              1279
  Free PE               0
  Allocated PE          1279
  PV UUID               gy2E19-BHhh-lHyl-MGJg-P6tu-iSeh-vYpO8U
```

Even though the disk was resized, the PV still has the previous size.

9. Resize the PV:

```
Bash

sudo pvresize /dev/resizeddisk
```

```
[root@vmresizelv ~]# pvresize /dev/sdd
  Physical volume "/dev/sdd" changed
  1 physical volume(s) resized / 0 physical volume(s) not resized
```

10. Check the PV size:

```
Bash

sudo pvdisplay /dev/resizeddisk
```

```
[root@vmresizelv ~]# pvdisplay /dev/sdd
  --- Physical volume ---
  PV Name               /dev/sdd
  VG Name               datavg
  PV Size               <20.00 GiB / not usable 3.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              5119
  Free PE               3840
  Allocated PE          1279
  PV UUID               gy2E19-BHhh-lHyl-MGJg-P6tu-iSeh-vYpO8U
```

Apply the same procedure for all of the disks that you want to resize.

11. Check the VG information.

```
Bash

sudo vgdisplay vgname
```

```
[root@vmresizelv ~]# vgdisplay datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        4
  Metadata Sequence No  17
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                4
  Act PV                4
  VG Size               34.98 GiB
  PE Size               4.00 MiB
  Total PE              8956
  Alloc PE / Size       5116 / 19.98 GiB
  Free  PE / Size       3840 / 15.00 GiB
  VG UUID               jvclKz-FXLd-dNjC-14q7-k85I-Pp8z-bMqFLY
```

The VG now has enough space to be allocated to the LVs.

12. Resize the LV:

```
Bash

sudo lvresize -r -L +5G vgname/lvname
sudo lvresize -r -l +100%FREE /dev/datavg/datalv01
```

```
[root@vmresizelv ~]# lvresize -r -L +5G /dev/datavg/datalv01
  Size of logical volume datavg/datalv01 changed from 19.98 GiB (5116 extents) to 24.98 GiB (6396 extents).
  Logical volume datavg/datalv01 successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/bb92534b-47a8-40da-adff-c4e9c0f643f0 is mounted on /datalv01; on-line resizing required
old_desc_blocks = 3, new_desc_blocks = 4
The filesystem on /dev/mapper/bb92534b-47a8-40da-adff-c4e9c0f643f0 is now 6548992 blocks long.
```

13. Check the size of the file system:

```
Bash

df -h /datalvm2
```

```
[root@vmresizelv ~]# df -h /datalv01/
Filesystem                                      Size  Used Avail Use% Mounted on
/dev/mapper/bb92534b-47a8-40da-adff-c4e9c0f643f0   35G   20G   14G  60% /datalv01
```

Extend an LVM-on-crypt volume by adding a new PV

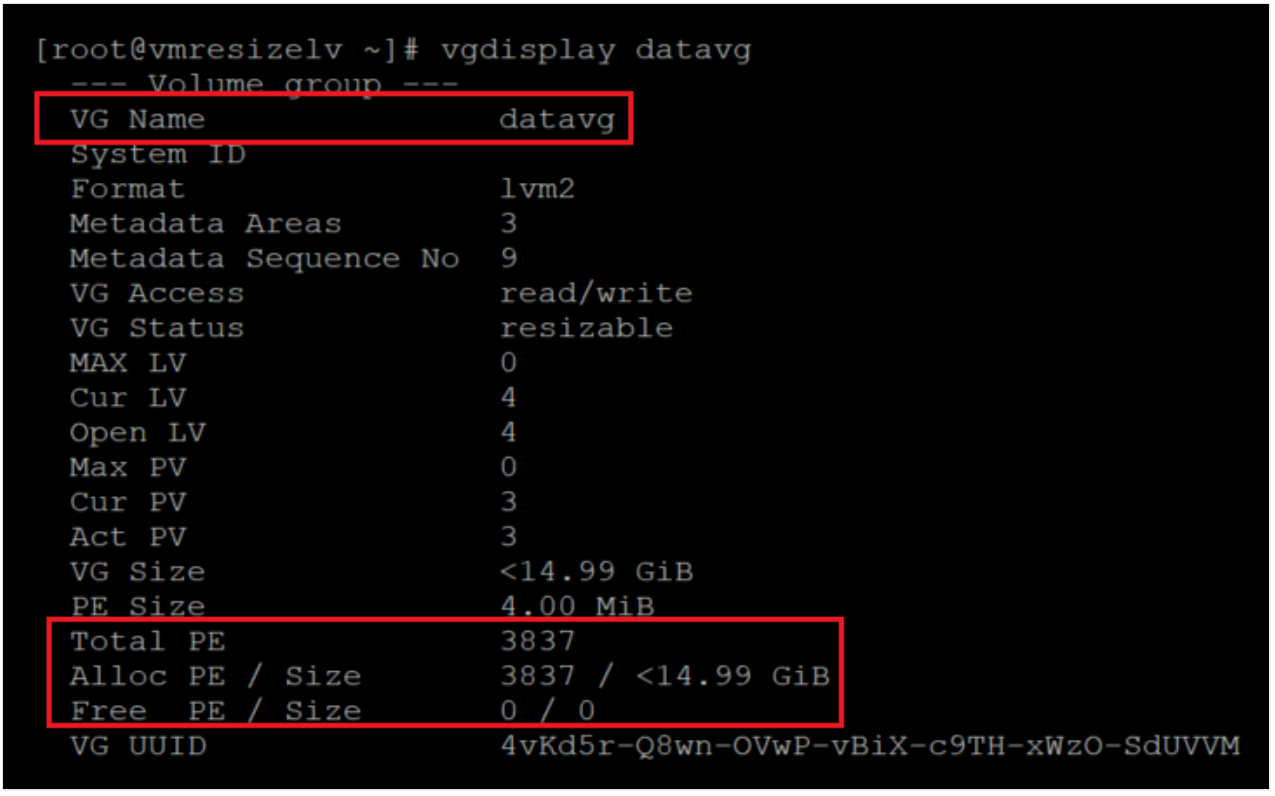
You can also extend an LVM-on-crypt volume by adding a new PV. This method closely follows the steps in Configure LVM and RAID on encrypted devices. See the sections that explain how

to add a new disk and set it up in an LVM-on-crypt configuration.

You can use this method to add space to an existing LV. Or you can create new VGs or LVs.

1. Verify the current size of your VG:

```
Bash

sudo vgdisplay vgname
```



2. Verify the size of the file system and LV that you want to expand:

```
Bash

sudo lvdisplay /dev/vgname/lvname
```

```
[root@vmresizelv ~]# lvdisplay /dev/datavg/datalv01
  --- Logical volume ---
  LV Path                /dev/datavg/datalv01
  LV Name                datalv01
  VG Name                datavg
  LV UUID                K8zZjy-brpp-Lx74-atfR-lTVu-h26Q-urRi7z
  LV Write Access        read/write
  LV Creation host, time vmresizelv, 2020-09-21 14:44:27 +0000
  LV Status              available
  # open                 1
  LV Size                1.00 GiB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:4
```

Bash

```
df -h mountpoint
```

```
[root@vmresizelv ~]# df -h /datalv01
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/datavg-datalv01  976M  960M     0 100% /datalv01
```

3. Add a new data disk to the VM and identify it.

   Before you add the new disk, check the disks:

Bash

```
sudo fdisk -l | egrep ^"Disk /"
```

```
[root@vmresizelv ~]# fdisk -l | egrep ^"Disk /"
Disk /dev/sdc: 50 MB, 50331648 bytes, 98304 sectors
Disk /dev/sdf: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdg: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sde: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdb: 34.4 GB, 34359738368 bytes, 67108864 sectors
Disk /dev/sda: 34.4 GB, 34359738368 bytes, 67108864 sectors
Disk /dev/mapper/resourceencrypt: 34.4 GB, 34356527104 bytes, 67102592 sectors
Disk /dev/mapper/e19fcd77-e974-4e5c-a874-e78e4b6d2f48: 5366 MB, 5366611968 bytes, 10481664 sectors
Disk /dev/mapper/49de5df0-1c65-48bf-809c-588805eda921: 5366 MB, 5366611968 bytes, 10481664 sectors
Disk /dev/mapper/8a33198f-fa7e-491f-a9c6-e9d0a1d25d57: 5366 MB, 5366611968 bytes, 10481664 sectors
Disk /dev/mapper/datavg-datalv01: 1073 MB, 1073741824 bytes, 2097152 sectors
Disk /dev/mapper/datavg-datalv02: 6442 MB, 6442450944 bytes, 12582912 sectors
Disk /dev/mapper/datavg-datalv03: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/mapper/datavg-datalv04: 4282 MB, 4282384384 bytes, 8364032 sectors
```

   Here's another way to check the disks before you add the new disk:

Bash

```
sudo lsblk
```

```
[root@vmresizelv ~]# lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part
  └─resourceencrypt                     253:0    0   32G  0 crypt /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   47M  0 part  /mnt/azure_bek_disk
sde                                       8:64   0    5G  0 disk
└─e19fcd77-e974-4e5c-a874-e78e4b6d2f48  253:1    0    5G  0 crypt
  ├─datavg-datalv01                     253:4    0    1G  0 lvm   /datalv01
  ├─datavg-datalv02                     253:5    0    6G  0 lvm   /datalv02
  ├─datavg-datalv03                     253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                     253:7    0    3G  0 lvm   /datalv04
sdf                                       8:80   0    5G  0 disk
└─49de5df0-1c65-48bf-809c-588805eda921  253:2    0    5G  0 crypt
  ├─datavg-datalv02                     253:5    0    6G  0 lvm   /datalv02
  └─datavg-datalv04                     253:7    0    3G  0 lvm   /datalv04
sdg                                       8:96   0    5G  0 disk
└─8a33198f-fa7e-491f-a9c6-e9d0a1d25d57  253:3    0    5G  0 crypt
  ├─datavg-datalv03                     253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                     253:7    0    3G  0 lvm   /datalv04
```

To add the new disk, you can use PowerShell, the Azure CLI, or the Azure portal. For more information, see Attach a data disk to a Linux VM.

The kernel name scheme applies to the newly added device. A new drive is normally assigned the next available letter. In this case, the added disk is `sdd`.

4. Check the disks to make sure the new disk has been added:

Bash

```
sudo fdisk -l | egrep ^"Disk /"
```

```
[root@vmresizelv ~]# fdisk -l | egrep ^"Disk /"
Disk /dev/sdc: 50 MB, 50331648 bytes, 98304 sectors
Disk /dev/sdf: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdg: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sde: 5368 MB, 5368709120 bytes, 10485760 sectors
Disk /dev/sdb: 34.4 GB, 34359738368 bytes, 67108864 sectors
Disk /dev/sda: 34.4 GB, 34359738368 bytes, 67108864 sectors
Disk /dev/mapper/resourceencrypt: 34.4 GB, 34356527104 bytes, 67102592 sectors
Disk /dev/mapper/e19fcd77-e974-4e5c-a874-e78e4b6d2f48: 5366 MB, 5366611968 bytes, 10481664 sectors
Disk /dev/mapper/49de5df0-1c65-48bf-809c-588805eda921: 5366 MB, 5366611968 bytes, 10481664 sectors
Disk /dev/mapper/8a33198f-fa7e-491f-a9c6-e9d0a1d25d57: 5366 MB, 5366611968 bytes, 10481664 sectors
Disk /dev/mapper/datavg-datalv01: 1073 MB, 1073741824 bytes, 2097152 sectors
Disk /dev/mapper/datavg-datalv02: 6442 MB, 6442450944 bytes, 12582912 sectors
Disk /dev/mapper/datavg-datalv03: 4294 MB, 4294967296 bytes, 8388608 sectors
Disk /dev/mapper/datavg-datalv04: 4282 MB, 4282384384 bytes, 8364032 sectors
Disk /dev/sdd: 5368 MB, 5368709120 bytes, 10485760 sectors
```

Bash

```
sudo lsblk
```

```
[root@vmresizelv ~]# lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                       2:0    1    4K  0 disk
sda                                       8:0    0   32G  0 disk
├─sda1                                    8:1    0  500M  0 part  /boot
└─sda2                                    8:2    0 31.5G  0 part  /
sdb                                       8:16   0   32G  0 disk
└─sdb1                                    8:17   0   32G  0 part
  └─resourceencrypt                     253:0    0   32G  0 crypt /mnt/resource
sdc                                       8:32   0   48M  0 disk
└─sdc1                                    8:33   0   47M  0 part  /mnt/azure_bek_disk
sdd                                       8:48   0    5G  0 disk
sde                                       8:64   0    5G  0 disk
└─e19fcd77-e974-4e5c-a874-e78e4b6d2f48  253:1    0    5G  0 crypt
  ├─datavg-datalv01                     253:4    0    1G  0 lvm   /datalv01
  ├─datavg-datalv02                     253:5    0    6G  0 lvm   /datalv02
  ├─datavg-datalv03                     253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                     253:7    0    4G  0 lvm   /datalv04
sdf                                       8:80   0    5G  0 disk
└─49de5df0-1c65-48bf-809c-588805eda921  253:2    0    5G  0 crypt
  ├─datavg-datalv02                     253:5    0    6G  0 lvm   /datalv02
  └─datavg-datalv04                     253:7    0    4G  0 lvm   /datalv04
sdg                                       8:96   0    5G  0 disk
└─8a33198f-fa7e-491f-a9c6-e9d0a1d25d57  253:3    0    5G  0 crypt
  ├─datavg-datalv03                     253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                     253:7    0    4G  0 lvm   /datalv04
```

5. Create a file system on top of the recently added disk. Match the disk to the linked devices on `/dev/disk/azure/scsi1/`.

Bash

```
sudo ls -la /dev/disk/azure/scsi1/
```

```
[root@vmresizelv ~]# ls -la /dev/disk/azure/scsi1/*
lrwxrwxrwx. 1 root root 12 Sep 21 14:32 /dev/disk/azure/scsi1/lun1 -> ../../../sde
lrwxrwxrwx. 1 root root 12 Sep 21 14:32 /dev/disk/azure/scsi1/lun2 -> ../../../sdf
lrwxrwxrwx. 1 root root 12 Sep 21 14:32 /dev/disk/azure/scsi1/lun3 -> ../../../sdg
lrwxrwxrwx. 1 root root 12 Sep 21 16:39 /dev/disk/azure/scsi1/lun4 -> ../../../sdd
```

Bash

```
sudo mkfs.ext4 /dev/disk/azure/scsi1/${disk}
```

```
[root@vmresizelv ~]# mkfs.ext4 /dev/disk/azure/scsi1/lun4
mke2fs 1.42.9 (28-Dec-2013)
/dev/disk/azure/scsi1/lun4 is entire device, not just one partition!
Proceed anyway? (y,n) y
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310720 blocks
65536 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

6. Create a temporary mount point for the new added disk:

```
Bash

newmount=/data4
sudo mkdir ${newmount}
```

7. Add the recently created file system to `/etc/fstab`.

```
Bash

sudo blkid /dev/disk/azure/scsi1/lun4| awk -F\" '{print "UUID="$2"
'${newmount}' "$4" defaults,nofail 0 0"}' >> /etc/fstab
```

8. Mount the newly created file system:

```
Bash

sudo mount -a
```

9. Verify that the new file system is mounted:

```
Bash

df -h
```

```
[root@vmresizelv ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/sda2                    32G  2.7G   29G   9% /
devtmpfs                    7.9G     0  7.9G   0% /dev
tmpfs                       7.9G     0  7.9G   0% /dev/shm
tmpfs                       7.9G   18M  7.9G   1% /run
tmpfs                       7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sda1                   497M  100M  397M  21% /boot
tmpfs                       1.6G     0  1.6G   0% /run/user/1000
/dev/sdc1                    43M  4.5K   43M   1% /mnt/azure_bek_disk
/dev/mapper/resourceencrypt  32G   49M   30G   1% /mnt/resource
tmpfs                       1.6G     0  1.6G   0% /run/user/995
/dev/mapper/datavg-datalv01 976M  960M     0 100% /datalv01
/dev/mapper/datavg-datalv02 5.9G  5.9G     0 100% /datalv02
/dev/mapper/datavg-datalv03 4.0G  3.9G     0 100% /datalv03
/dev/mapper/datavg-datalv04 3.9G  4.0M  3.8G   1% /datalv04
/dev/sdd                    4.8G   20M  4.6G   1% /data4
```

Bash

```
sudo lsblk
```

```
[root@vmresizelv ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
fd0                                   2:0    1    4K  0 disk
sda                                   8:0    0   32G  0 disk
├─sda1                                8:1    0  500M  0 part  /boot
└─sda2                                8:2    0 31.5G  0 part  /
sdb                                   8:16   0   32G  0 disk
└─sdb1                                8:17   0   32G  0 part
  └─resourceencrypt                 253:0    0   32G  0 crypt /mnt/resource
sdc                                   8:32   0   48M  0 disk
└─sdc1                                8:33   0   47M  0 part  /mnt/azure_bek_disk
sdd                                   8:48   0    5G  0 disk  /data4
sde                                   8:64   0    5G  0 disk
└─e19fcd77-e974-4e5c-a874-e78e4b6d2f48 253:1    0    5G  0 crypt
  ├─datavg-datalv01                 253:4    0    1G  0 lvm   /datalv01
  ├─datavg-datalv02                 253:5    0    6G  0 lvm   /datalv02
  ├─datavg-datalv03                 253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                 253:7    0    4G  0 lvm   /datalv04
sdf                                   8:80   0    5G  0 disk
└─49de5df0-1c65-48bf-809c-588805eda921 253:2    0    5G  0 crypt
  ├─datavg-datalv02                 253:5    0    6G  0 lvm   /datalv02
  └─datavg-datalv04                 253:7    0    4G  0 lvm   /datalv04
sdg                                   8:96   0    5G  0 disk
└─8a33198f-fa7e-491f-a9c6-e9d0a1d25d57 253:3    0    5G  0 crypt
  ├─datavg-datalv03                 253:6    0    4G  0 lvm   /datalv03
  └─datavg-datalv04                 253:7    0    4G  0 lvm   /datalv04
```

10. Restart the encryption that you previously started for data drives.

> **Tip**
>
> For LVM-on-crypt, we recommend that you use `EncryptFormatAll`. Otherwise, you might see a double encryption while you set additional disks.
>
> For more information, see Configure LVM and RAID on encrypted devices.

Here's an example:

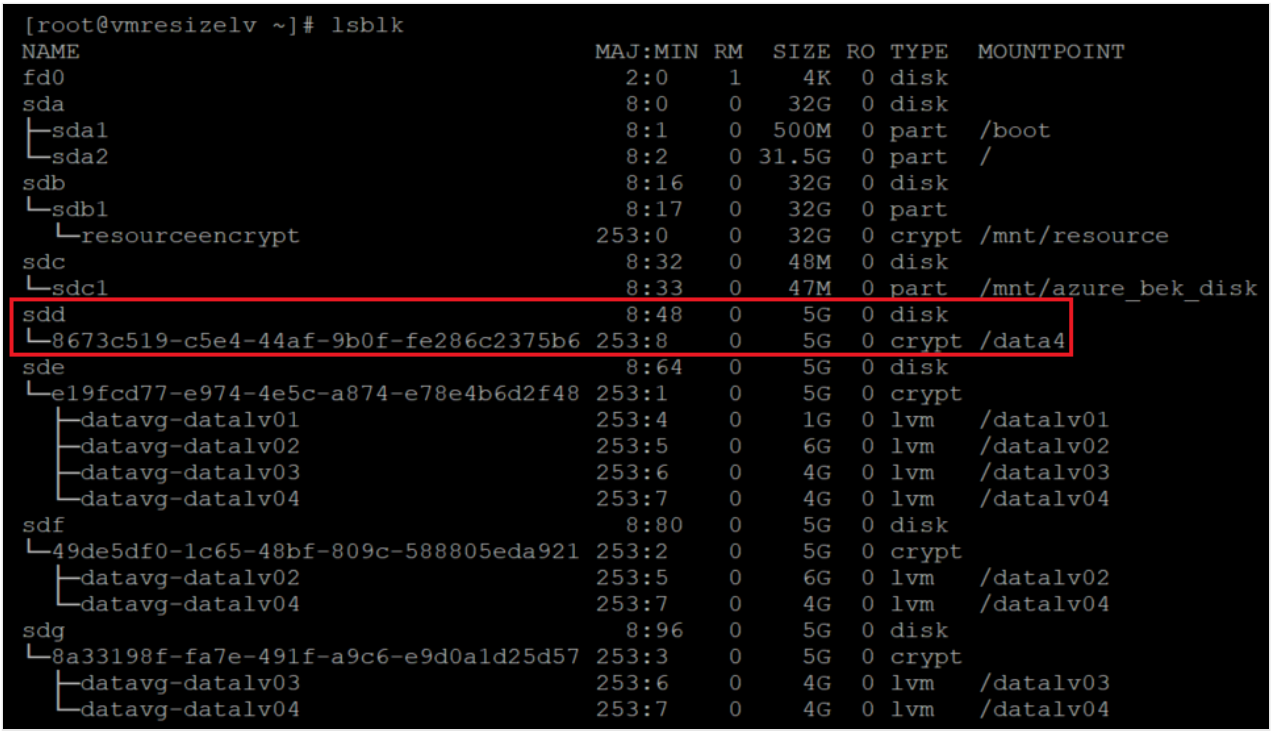
Azure CLI

```
az vm encryption enable \
--resource-group ${RGNAME} \
--name ${VMNAME} \
--disk-encryption-keyvault ${KEYVAULTNAME} \
--key-encryption-key ${KEYNAME} \
--key-encryption-keyvault ${KEYVAULTNAME} \
--volume-type "DATA" \
--encrypt-format-all \
-o table
```

When the encryption finishes, you see a crypt layer on the newly added disk:

Bash

```
sudo lsblk
```



11. Unmount the encrypted layer of the new disk:

Bash

```
sudo umount ${newmount}
```

12. Check the current PV information:

Bash

```
sudo pvs
```

```
[root@vmresizelv ~]# pvs
  PV                                            VG     Fmt  Attr PSize  PFree
  /dev/mapper/49de5df0-1c65-48bf-809c-588805eda921 datavg lvm2 a--  <5.00g    0
  /dev/mapper/8a33198f-fa7e-491f-a9c6-e9d0a1d25d57 datavg lvm2 a--  <5.00g    0
  /dev/mapper/e19fcd77-e974-4e5c-a874-e78e4b6d2f48 datavg lvm2 a--  <5.00g    0
```

13. Create a PV on top of the encrypted layer of the disk. Take the device name from the previous `lsblk` command. Add a `/dev/` mapper in front of the device name to create the PV:

```
Bash

sudo pvcreate /dev/mapper/mapperdevicename
```

```
[root@vmresizelv ~]# pvcreate /dev/mapper/8673c519-c5e4-44af-9b0f-fe286c2375b6
WARNING: ext4 signature detected on /dev/mapper/8673c519-c5e4-44af-9b0f-fe286c2375b6 at offset 1080. Wipe it? [y/n]: y
  Wiping ext4 signature on /dev/mapper/8673c519-c5e4-44af-9b0f-fe286c2375b6.
  Physical volume "/dev/mapper/8673c519-c5e4-44af-9b0f-fe286c2375b6" successfully created.
```

You see a warning about wiping the current `ext4 fs` signature. This warning is expected. Answer this question with `y`.

14. Verify that the new PV was added to the LVM configuration:

```
Bash

sudo pvs
```

```
[root@vmresizelv ~]# pvs
  PV                                            VG     Fmt  Attr PSize  PFree
  /dev/mapper/49de5df0-1c65-48bf-809c-588805eda921 datavg lvm2 a--  <5.00g    0
  /dev/mapper/8673c519-c5e4-44af-9b0f-fe286c2375b6       lvm2 ---  <5.00g <5.00g
  /dev/mapper/8a33198f-fa7e-491f-a9c6-e9d0a1d25d57 datavg lvm2 a--  <5.00g    0
  /dev/mapper/e19fcd77-e974-4e5c-a874-e78e4b6d2f48 datavg lvm2 a--  <5.00g    0
```

15. Add the new PV to the VG that you need to increase.

```
Bash

sudo vgextend vgname /dev/mapper/nameofhenewpv
```

```
[root@vmresizelv ~]# vgextend datavg /dev/mapper/8673c519-c5e4-44af-9b0f-fe286c2375b6
  Volume group "datavg" successfully extended
```

16. Verify the new size and free space of the VG:

```
Bash
```

```
sudo vgdisplay vgname
```

```
[root@vmresizelv ~]# vgdisplay datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        4
  Metadata Sequence No  10
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                4
  Open LV               4
  Max PV                0
  Cur PV                4
  Act PV                4
  VG Size               19.98 GiB
  PE Size               4.00 MiB
  Total PE              5116
  Alloc PE / Size       3837 / <14.99 GiB
  Free  PE / Size       1279 / <5.00 GiB
  VG UUID               4vKd5r-Q8wn-OVwP-vBiX-c9TH-xWzO-SdUVVM
```

Note the increase of the `Total PE` count and the `Free PE / Size`.

17. Increase the size of the LV and the file system. Use the `-r` option on `lvextend`. In this example, we're adding the total available space in the VG to the given LV.

Bash

```
sudo lvextend -r -l +100%FREE /dev/vgname/lvname
```

```
[root@vmresizelv ~]# lvextend -r -l +100%FREE datavg/datalv01
  Size of logical volume datavg/datalv01 changed from 1.00 GiB (256 extents) to <6.00 GiB (1535 extents).
  Logical volume datavg/datalv01 successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/datavg-datalv01 is mounted on /datalv01; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mapper/datavg-datalv01 is now 1571840 blocks long.
```

Follow the next steps to verify your changes.

1. Verify the size of the LV:

Bash

```
sudo lvdisplay /dev/vgname/lvname
```

```
[root@vmresizelv ~]# lvdisplay /dev/datavg/datalv01
  --- Logical volume ---
  LV Path                /dev/datavg/datalv01
  LV Name                datalv01
  VG Name                datavg
  LV UUID                K8zZjy-brpp-Lx74-atfR-lTVu-h26Q-urRi7z
  LV Write Access        read/write
  LV Creation host, time vmresizelv, 2020-09-21 14:44:27 +0000
  LV Status              available
  # open                 1
  LV Size                <6.00 GiB
  Current LE             1535
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:4
```

2. Verify the new size of the file system:

Bash

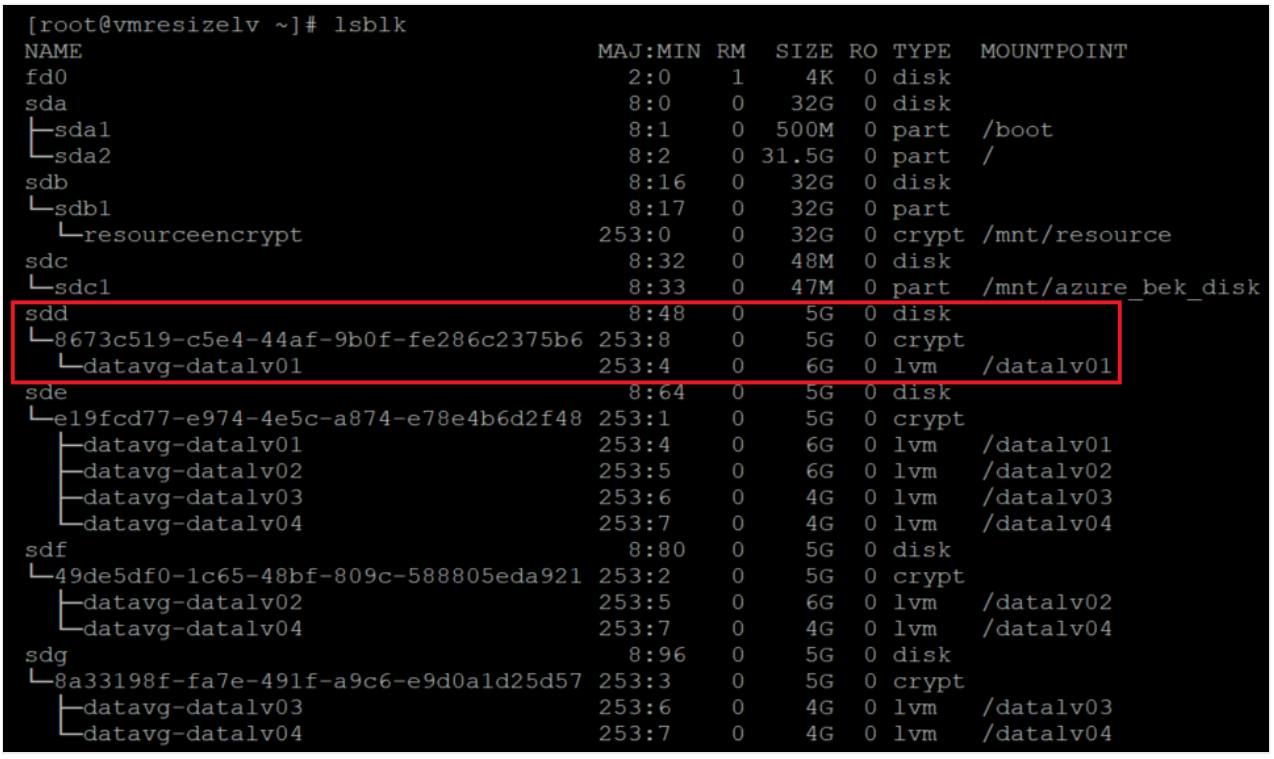
```
df -h /mountpoint
```

```
[root@vmresizelv ~]# df -h /datalv01/
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/datavg-datalv01  5.9G  962M  4.7G  17% /datalv01
```

3. Verify that the LVM layer is on top of the encrypted layer:

Bash

```
sudo lsblk
```

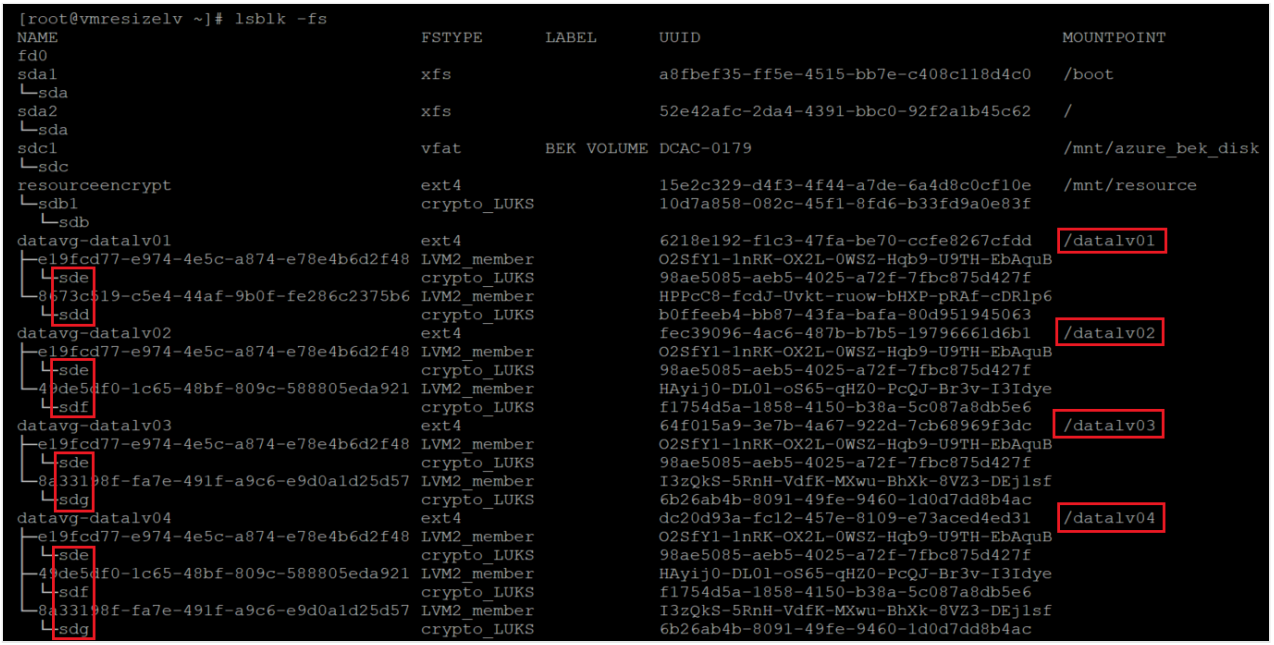


If you use `lsblk` without options, then you see the mount points multiple times. The command sorts by device and LVs.

You might want to use `lsblk -fs`. In this command, `-fs` reverses the sort order so that the mount points are shown once. The disks are shown multiple times.

Bash

```
sudo lsblk -fs
```

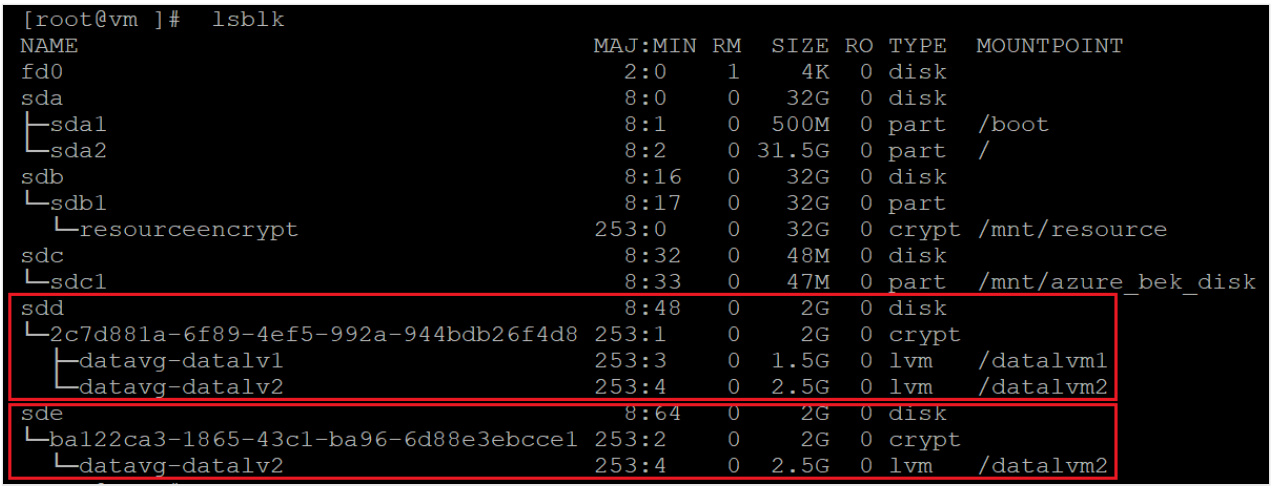


Extend an LVM on a crypt volume by resizing an existing PV

1. Identify your encrypted disks:

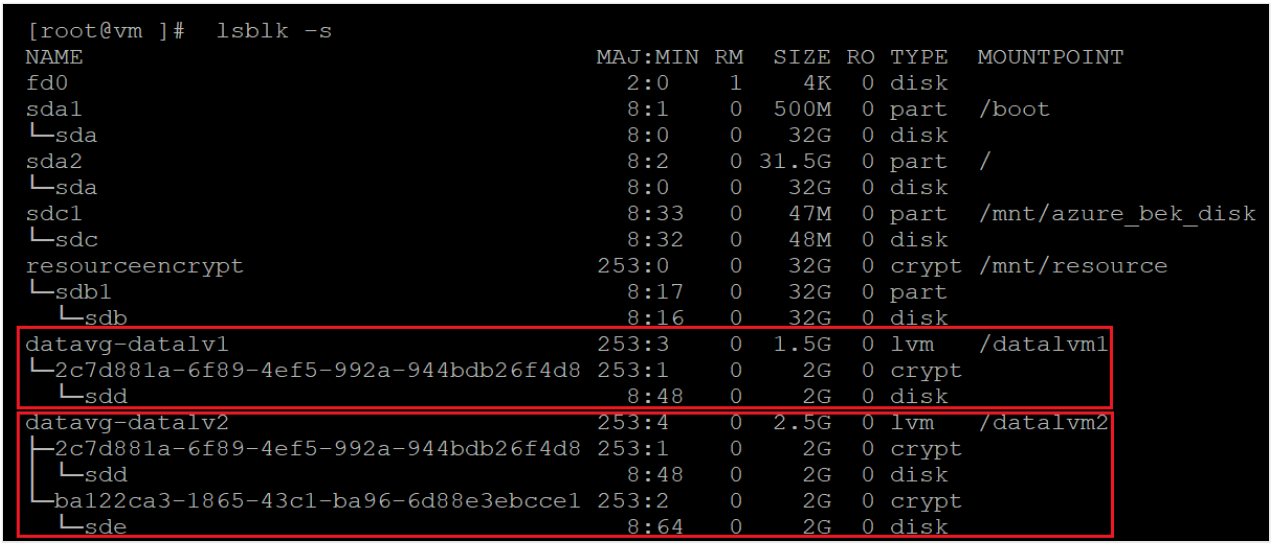
Bash

```
sudo lsblk
```



Bash

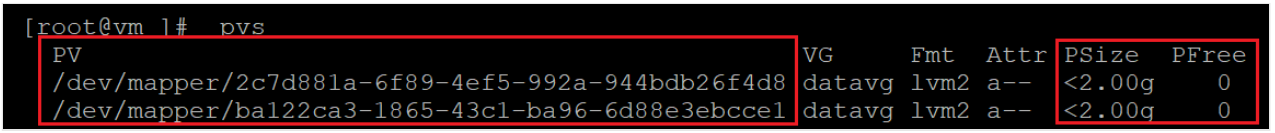
```
sudo lsblk -s
```



2. Check your PV information:

Bash

```
sudo pvs
```



3. Check your VG information:

```
Bash

sudo vgs
```

```
[root@vm ]#  vgdisplay  datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.99 GiB
  PE Size               4.00 MiB
  Total PE              1022
  Alloc PE / Size       1022 / 3.99 GiB
  Free  PE / Size       0 / 0
  VG UUID               24nwHl-raFN-igfO-0wk9-WTvc-sSTO-cXVYUX
```

4. Check your LV information:

```
Bash

sudo lvs
```

```
[root@vm ]#  lvs
  LV      VG     Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  datalv1 datavg -wi-ao---- 1.50g
  datalv2 datavg -wi-ao---- 2.49g
```

5. Check the file system utilization:

```
Bash

df -h /mountpoint(s)
```

```
[root@vm ]#  df -h  /datalvm*
Filesystem                   Size  Used Avail Use% Mounted on
/dev/mapper/datavg-datalv1   1.5G  1.5G     0 100% /datalvm1
/dev/mapper/datavg-datalv2   2.4G  2.4G     0 100% /datalvm2
```

6. Check the sizes of your disks:

Bash

```
sudo fdisk
sudo fdisk -l | egrep ^"Disk /"
sudo lsblk
```

```
[root@vm ]#  fdisk -l /dev/sdd

Disk /dev/sdd: 2147 MB, 2147483648 bytes, 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

[root@vm ]#  fdisk -l /dev/sde

Disk /dev/sde: 2147 MB, 2147483648 bytes, 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

7. Resize the data disk. You can use the portal, CLI, or PowerShell. For more information, see the disk-resize section in Expand virtual hard disks on a Linux VM.

> **ⓘ Important**
>
> You can't resize virtual disks while the VM is running. Deallocate your VM for this step.

8. Check your disks sizes:

Bash

```
sudo fdisk
sudo fdisk -l | egrep ^"Disk /"
sudo lsblk
```

```
[root@vmresizelvm ~]# fdisk -l /dev/sdd

Disk /dev/sdd: 4294 MB, 4294967296 bytes, 8388608 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

[root@vmresizelvm ~]# fdisk -l /dev/sde

Disk /dev/sde: 4294 MB, 4294967296 bytes, 8388608 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

In this case, both disks were resized from 2 GB to 4 GB. But the size of the file system, LV, and PV remain the same.

9. Check the current PV size. Remember that on LVM-on-crypt, the PV is the `/dev/mapper/` device, not the `/dev/sd*` device.

Bash

```
sudo pvdisplay /dev/mapper/devicemappername
```

```
[root@vmresizelvm ~]# pvdisplay /dev/mapper/2c7d881a-6f89-4ef5-992a-944bdb26f4d8
  --- Physical volume ---
  PV Name               /dev/mapper/2c7d881a-6f89-4ef5-992a-944bdb26f4d8
  VG Name               datavg
  PV Size               <2.00 GiB / not usable 2.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              511
  Free PE               0
  Allocated PE          511
  PV UUID               30iBnM-fZgv-1sXe-FJo2-052x-wjxu-8Fb6Mw
```

10. Resize the PV:

Bash

```
sudo pvresize /dev/mapper/devicemappername
```

```
[root@vmresizelvm ~]# pvresize /dev/mapper/2c7d881a-6f89-4ef5-992a-944bdb26f4d8
  Physical volume "/dev/mapper/2c7d881a-6f89-4ef5-992a-944bdb26f4d8" changed
  1 physical volume(s) resized / 0 physical volume(s) not resized
```

11. Check the new PV size:

Bash

```
sudo pvdisplay /dev/mapper/devicemappername
```

```
[root@vmresizelvm ~]# pvdisplay /dev/mapper/2c7d881a-6f89-4ef5-992a-944bdb26f4d8
  --- Physical volume ---
  PV Name               /dev/mapper/2c7d881a-6f89-4ef5-992a-944bdb26f4d8
  VG Name               datavg
  PV Size               <4.00 GiB / not usable 0
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              1023
  Free PE               512
  Allocated PE          511
  PV UUID               30iBnM-fZgv-1sXe-FJo2-052x-wjxu-8Fb6Mw
```

12. Resize the encrypted layer on the PV:

```
Bash

sudo cryptsetup resize /dev/mapper/devicemappername
```

Apply the same procedure for all of the disks that you want to resize.

13. Check your VG information:

```
Bash

sudo vgdisplay vgname
```

```
[root@vmresizelvm ~]# vgdisplay datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               5.99 GiB
  PE Size               4.00 MiB
  Total PE              1534
  Alloc PE / Size       1022 / 3.99 GiB
  Free  PE / Size       512 / 2.00 GiB
  VG UUID               24nwHl-raFN-igfO-0wk9-WTvc-sSTO-cXVYUX
```

The VG now has enough space to be allocated to the LVs.

14. Check the LV information:

```
Bash

sudo lvdisplay vgname/lvname
```

```
[root@vmresizelvm ~]# lvdisplay datavg/datalv2
  --- Logical volume ---
  LV Path                /dev/datavg/datalv2
  LV Name                datalv2
  VG Name                datavg
  LV UUID                iK3HYF-wYPX-XAso-0WGB-Zite-D1aT-dFae2o
  LV Write Access        read/write
  LV Creation host, time vmresizelvm, 2020-05-18 18:33:54 +0000
  LV Status              available
  # open                 1
  LV Size                2.49 GiB
  Current LE             638
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3
```

15. Check the file system utilization:

```
Bash

df -h /mountpoint
```

```
[root@vmresizelvm ~]# df -h /datalvm2/
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/datavg-datalv2  2.4G  2.4G     0 100% /datalvm2
```

16. Resize the LV:

```
Bash

sudo lvresize -r -L +2G /dev/vgname/lvname
```

```
[root@vmresizelvm]#  lvresize -r -L +2G /dev/datavg/datalv2
  Size of logical volume datavg/datalv2 changed from 2.50 GiB (640 extents) to 4.50 GiB (1152 extents).
  Logical volume datavg/datalv2 successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/datavg-datalv2 is mounted on /datalvm2; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/mapper/datavg-datalv2 is now 1179648 blocks long.
```

Here we use the `-r` option to also resize the file system.

17. Check the LV information:

Bash

```
sudo lvdisplay vgname/lvname
```

```
[root@vmresizelvm ~]# lvdisplay datavg/datalv2
  --- Logical volume ---
  LV Path                /dev/datavg/datalv2
  LV Name                datalv2
  VG Name                datavg
  LV UUID                iK3HYF-wYPX-XAso-0WGB-Zite-D1aT-dFae2o
  LV Write Access        read/write
  LV Creation host, time vmresizelvm, 2020-05-18 18:33:54 +0000
  LV Status              available
  # open                 1
  LV Size                4.49 GiB
  Current LE             1150
  Segments               3
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:3
```

18. Check the file system utilization:

Bash

```
df -h /mountpoint
```

```
[root@vmresizelvm ~]# df -h /datalvm2/
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/datavg-datalv2  4.4G  2.4G  1.8G  58% /datalvm2
```

Apply the same resizing procedure to any other LV that requires it.

Next steps

Troubleshoot Azure Disk Encryption

Azure Disk Encryption for Linux VMs troubleshooting guide

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This guide is for IT professionals, information security analysts, and cloud administrators whose organizations use Azure Disk Encryption. This article is to help with troubleshooting disk-encryption-related problems.

Before taking any of these steps, first ensure that the VMs you are attempting to encrypt are among the supported VM sizes and operating systems, and that you meet all the prerequisites:

- Additional requirements for VMs
- Networking requirements
- Encryption key storage requirements

Troubleshooting Linux OS disk encryption

Linux operating system (OS) disk encryption must unmount the OS drive before running it through the full disk encryption process. If it can't unmount the drive, an error message of "failed to unmount after ..." is likely to occur.

This error can occur when OS disk encryption is attempted on a VM with an environment that has been changed from the supported stock gallery image. Deviations from the supported image can interfere with the extension's ability to unmount the OS drive. Examples of deviations can include the following items:

- Customized images no longer match a supported file system or partitioning scheme.
- Large applications such as SAP, MongoDB, Apache Cassandra, and Docker aren't supported when they're installed and running in the OS before encryption. Azure Disk Encryption is unable to shut down these processes safely as required in preparation of the

OS drive for disk encryption. If there are still active processes holding open file handles to the OS drive, the OS drive can't be unmounted, resulting in a failure to encrypt the OS drive.

- Custom scripts that run in close time proximity to the encryption being enabled, or if any other changes are being made on the VM during the encryption process. This conflict can happen when an Azure Resource Manager template defines multiple extensions to execute simultaneously, or when a custom script extension or other action runs simultaneously to disk encryption. Serializing and isolating such steps might resolve the issue.
- Security Enhanced Linux (SELinux) hasn't been disabled before enabling encryption, so the unmount step fails. SELinux can be reenabled after encryption is complete.
- The OS disk uses a Logical Volume Manager (LVM) scheme. Although limited LVM data disk support is available, an LVM OS disk isn't.
- Minimum memory requirements aren't met (7 GB is suggested for OS disk encryption).
- Data drives are recursively mounted under the /mnt/ directory, or each other (for example, /mnt/data1, /mnt/data2, /data3 + /data3/data4).

Update the default kernel for Ubuntu 14.04 LTS

The Ubuntu 14.04 LTS image ships with a default kernel version of 4.4. This kernel version has a known issue in which Out of Memory Killer improperly terminates the dd command during the OS encryption process. This bug has been fixed in the most recent Azure tuned Linux kernel. To avoid this error, prior to enabling encryption on the image, update to the Azure tuned kernel 4.15 or later using the following commands:

```
sudo apt-get update
sudo apt-get install linux-azure
sudo reboot
```

After the VM has restarted into the new kernel, the new kernel version can be confirmed using:

```
uname -a
```

Update the Azure Virtual Machine Agent and extension versions

Azure Disk Encryption operations may fail on virtual machine images using unsupported versions of the Azure Virtual Machine Agent. Linux images with agent versions earlier than 2.2.38 should be updated prior to enabling encryption. For more information, see How to update the Azure Linux Agent on a VM and Minimum version support for virtual machine agents in Azure.

The correct version of the Microsoft.Azure.Security.AzureDiskEncryption or Microsoft.Azure.Security.AzureDiskEncryptionForLinux guest agent extension is also required. Extension versions are maintained and updated automatically by the platform when Azure Virtual Machine agent prerequisites are satisfied and a supported version of the virtual machine agent is used.

The Microsoft.OSTCExtensions.AzureDiskEncryptionForLinux extension has been deprecated and is no longer supported.

Unable to encrypt Linux disks

In some cases, the Linux disk encryption appears to be stuck at "OS disk encryption started" and SSH is disabled. The encryption process can take between 3-16 hours to finish on a stock gallery image. If multi-terabyte-sized data disks are added, the process might take days.

The Linux OS disk encryption sequence unmounts the OS drive temporarily. It then performs block-by-block encryption of the entire OS disk, before it remounts it in its encrypted state. Linux Disk Encryption doesn't allow for concurrent use of the VM while the encryption is in progress. The performance characteristics of the VM can make a significant difference in the time required to complete encryption. These characteristics include the size of the disk and whether the storage account is standard or premium (SSD) storage.

While the OS drive is being encrypted, the VM enters a servicing state and disables SSH to prevent any disruption to the ongoing process. To check the encryption status, use the Azure PowerShell Get-AzVmDiskEncryptionStatus command, and check the **ProgressMessage** field. **ProgressMessage** will report a series of statuses as the data and OS disks are encrypted:

Azure PowerShell

```
PS > Get-AzVMDiskEncryptionStatus -ResourceGroupName "MyResourceGroup" -VMName
"myVM"

OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : EncryptionInProgress
OsVolumeEncryptionSettings :
ProgressMessage            : Transitioning

PS > Get-AzVMDiskEncryptionStatus -ResourceGroupName "MyResourceGroup" -VMName
"myVM"
```

```
OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : EncryptionInProgress
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : Encryption succeeded for data volumes

PS > Get-AzVMDiskEncryptionStatus -ResourceGroupName "MyResourceGroup" -VMName
"myVM"

OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : EncryptionInProgress
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : Provisioning succeeded

PS > Get-AzVMDiskEncryptionStatus -ResourceGroupName "MyResourceGroup" -VMName
"myVM"

OsVolumeEncrypted          : EncryptionInProgress
DataVolumesEncrypted       : EncryptionInProgress
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : OS disk encryption started
```

The **ProgressMessage** will remain in **OS disk encryption started** for most of the encryption process. When encryption is complete and successful, **ProgressMessage** will return:

Azure PowerShell

```
PS > Get-AzVMDiskEncryptionStatus -ResourceGroupName "MyResourceGroup" -VMName
"myVM"

OsVolumeEncrypted          : Encrypted
DataVolumesEncrypted       : NotMounted
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : Encryption succeeded for all volumes
```

After this message is available, the encrypted OS drive is expected to be ready for use and the VM is ready to be used again.

If the boot information, the progress message, or an error reports that OS encryption has failed in the middle of this process, restore the VM to the snapshot or backup taken immediately before encryption. An example of a message is the "failed to unmount" error that is described in this guide.

Before reattempting encryption, reevaluate the characteristics of the VM and make sure that all of the prerequisites are satisfied.

Troubleshooting Azure Disk Encryption behind a firewall

See Disk Encryption on an isolated network

Troubleshooting encryption status

The portal may display a disk as encrypted even after it has been unencrypted within the VM. This can occur when low-level commands are used to directly unencrypt the disk from within the VM, instead of using the higher level Azure Disk Encryption management commands. The higher level commands not only unencrypt the disk from within the VM, but outside of the VM they also update important platform level encryption settings and extension settings associated with the VM. If these are not kept in alignment, the platform will not be able to report encryption status or provision the VM properly.

To disable Azure Disk Encryption with PowerShell, use Disable-AzVMDiskEncryption followed by Remove-AzVMDiskEncryptionExtension. Running Remove-AzVMDiskEncryptionExtension before the encryption is disabled will fail.

To disable Azure Disk Encryption with CLI, use az vm encryption disable.

Next steps

In this document, you learned more about some common problems in Azure Disk Encryption and how to troubleshoot those problems. For more information about this service and its capabilities, see the following articles:

- Apply disk encryption in Microsoft Defender for Cloud
- Azure data encryption at rest

Azure Disk Encryption for Linux virtual machines FAQ

This article provides answers to frequently asked questions (FAQ) about Azure Disk Encryption for Linux virtual machines (VMs). For more information about this service, see Azure Disk Encryption overview.

What is Azure Disk Encryption for Linux virtual machines?

Azure Disk Encryption for Linux virtual machines uses the dm-crypt feature of Linux to provide full disk encryption of the OS disk* and data disks. Additionally, it provides encryption of the temporary disk when using the EncryptFormatAll feature. The content flows encrypted from the VM to the Storage backend with a customer-managed key.

See Supported virtual machines and operating systems.

Where is Azure Disk Encryption in general availability (GA)?

Azure Disk Encryption for Linux virtual machines is in general availability in all Azure public regions.

What user experiences are available with Azure Disk Encryption?

Azure Disk Encryption GA supports Azure Resource Manager templates, Azure PowerShell, and Azure CLI. The different user experiences give you flexibility. You have three different options for enabling disk encryption for your virtual machines. For more information on the user experience and step-by-step guidance available in Azure Disk Encryption, see Azure Disk Encryption scenarios for Linux.

How much does Azure Disk Encryption cost?

There's no charge for encrypting VM disks with Azure Disk Encryption, but there are charges associated with the use of Azure Key Vault. For more information on Azure Key Vault costs, see the Key Vault pricing page.

How can I start using Azure Disk Encryption?

To get started, read the Azure Disk Encryption overview.

What VM sizes and operating systems support Azure Disk Encryption?

The Azure Disk Encryption overview article lists the VM sizes and VM operating systems that support Azure Disk Encryption.

Can I encrypt both boot and data volumes with Azure Disk Encryption?

Yes, you can encrypt both boot and data volumes, or you can encrypt the data volume without having to encrypt the OS volume first.

After you've encrypted the OS volume, disabling encryption on the OS volume isn't supported. For Linux virtual machines in a scale set, only the data volume can be encrypted.

Can I encrypt an unmounted volume with Azure Disk Encryption?

No, Azure Disk Encryption only encrypts mounted volumes.

What is Storage server-side encryption?

Storage server-side encryption encrypts Azure managed disks in Azure Storage. Managed disks are encrypted by default with Server-side encryption with a platform-managed key (as of June 10, 2017). You can manage encryption of managed disks with your own keys by specifying a customer-managed key. For more information see: Server-side encryption of Azure managed disks.

How is Azure Disk Encryption different from other disk encryption solutions and when should I use each solution?

See Overview of managed disk encryption options.

How do I rotate secrets or encryption keys?

To rotate secrets, just call the same command you used originally to enable disk encryption, specifying a different Key Vault. To rotate the key encryption key, call the same command you used originally to enable disk encryption, specifying the new key encryption.

> ⚠ **Warning**
>
> - If you previously used **Azure Disk Encryption with Microsoft Entra app** by specifying Microsoft Entra credentials to encrypt this VM, you must continue to use this option to encrypt your VM. You can't use Azure Disk Encryption on this encrypted VM as this isn't a supported scenario, meaning switching away from Microsoft Entra application for this encrypted VM isn't supported yet.

How do I add or remove a key encryption key if I didn't originally use one?

To add a key encryption key, call the enable command again passing the key encryption key parameter. To remove a key encryption key, call the enable command again without the key encryption key parameter.

Does Azure Disk Encryption allow you to bring your own key (BYOK)?

Yes, you can supply your own key encryption keys. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption keys support scenarios, see Creating and configuring a key vault for Azure Disk Encryption.

Can I use an Azure-created key encryption key?

Yes, you can use Azure Key Vault to generate a key encryption key for Azure disk encryption use. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption key, see Creating and configuring a key vault for Azure Disk Encryption.

Can I use an on-premises key management service or HSM to safeguard the encryption keys?

You can't use the on-premises key management service or HSM to safeguard the encryption keys with Azure Disk Encryption. You can only use the Azure Key Vault service to safeguard the encryption keys. For more information on the key encryption key support scenarios, see Creating and configuring a key vault for Azure Disk Encryption.

What are the prerequisites to configure Azure Disk Encryption?

There are prerequisites for Azure Disk Encryption. See the Creating and configuring a key vault for Azure Disk Encryption article to create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see Creating and configuring a key vault for Azure Disk Encryption.

What are the prerequisites to configure Azure Disk Encryption with a Microsoft Entra app (previous release)?

There are prerequisites for Azure Disk Encryption. See the Azure Disk Encryption with Microsoft Entra ID content to create an Microsoft Entra application, create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and

keys. For more information on the key encryption key support scenarios, see Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID.

Is Azure Disk Encryption using a Microsoft Entra app (previous release) still supported?

Yes. Disk encryption using a Microsoft Entra app is still supported. However, when encrypting new virtual machines it's recommended that you use the new method rather than encrypting with a Microsoft Entra app.

Can I migrate virtual machines that were encrypted with a Microsoft Entra app to encryption without a Microsoft Entra app?

Currently, there isn't a direct migration path for machines that were encrypted with a Microsoft Entra app to encryption without a Microsoft Entra app. Additionally, there isn't a direct path from encryption without a Microsoft Entra app to encryption with an AD app.

What version of Azure PowerShell does Azure Disk Encryption support?

Use the latest version of the Azure PowerShell SDK to configure Azure Disk Encryption. Download the latest version of Azure PowerShell. Azure Disk Encryption is *not* supported by Azure SDK version 1.1.0.

> **Note**
>
> The Linux Azure disk encryption preview extension "Microsoft.OSTCExtension.AzureDiskEncryptionForLinux" is deprecated. This extension was published for Azure disk encryption preview release. You should not use the preview version of the extension in your testing or production deployment.

> For deployment scenarios like Azure Resource Manager (ARM), where you have a need to deploy Azure disk encryption extension for Linux VM to enable encryption on your Linux

> IaaS VM, you must use the Azure disk encryption production supported extension "Microsoft.Azure.Security.AzureDiskEncryptionForLinux".

Can I apply Azure Disk Encryption on my custom Linux image?

You can't apply Azure Disk Encryption on your custom Linux image. Only the gallery Linux images for the supported distributions called out previously are supported. Custom Linux images aren't currently supported.

Can I apply updates to a Linux Red Hat VM that uses the yum update?

Yes, you can perform a yum update on a Red Hat Linux VM. For more information, see Azure Disk Encryption on an isolated network.

What is the recommended Azure disk encryption workflow for Linux?

The following workflow is recommended to have the best results on Linux:

- Start from the unmodified stock gallery image corresponding to the needed OS distro and version
- Back up any mounted drives you want encrypted. This back up allows for recovery if there's a failure, for example if the VM is rebooted before encryption has completed.
- Encrypt (can take several hours or even days depending on VM characteristics and size of any attached data disks)
- Customize, and add software to the image as needed.

If this workflow isn't possible, relying on Storage Service Encryption (SSE) at the platform storage account layer may be an alternative to full disk encryption using dm-crypt.

What is the disk "Bek Volume" or "/mnt/azure_bek_disk"?

The "Bek volume" is a local data volume that securely stores the encryption keys for Encrypted Azure virtual machines.

> **Note**
>
> Do not delete or edit any contents in this disk. Do not unmount the disk since the encryption key presence is needed for any encryption operations on the IaaS VM.

What encryption method does Azure Disk Encryption use?

Azure Disk Encryption uses the decrypt default of aes-xts-plain64 with a 256-bit volume master key.

If I use EncryptFormatAll and specify all volume types, will it erase the data on the data drives that we already encrypted?

No, data won't be erased from data drives that are already encrypted using Azure Disk Encryption. Similar to how EncryptFormatAll didn't re-encrypt the OS drive, it won't re-encrypt the already encrypted data drive. For more information, see the EncryptFormatAll criteria.

Is XFS filesystem supported?

Encryption of XFS OS disks is supported.

Encryption of XFS data disks is supported only when the EncryptFormatAll parameter is used. This option reformats the volume, erasing any data previously there. For more information, see the EncryptFormatAll criteria.

Is resizing the OS partition supported?

Resize of an Azure Disk Encryption encrypted OS disk isn't supported.

Can I backup and restore an encrypted VM?

Azure Backup provides a mechanism to backup and restore encrypted VM's within the same subscription and region. For instructions, please see Back up and restore encrypted virtual machines with Azure Backup. Restoring an encrypted VM to a different region is not currently supported.

Where can I go to ask questions or provide feedback?

You can ask questions or provide feedback on the Microsoft Q&A question page for Azure Disk Encryption.

Next steps

In this document, you learned more about the most frequent questions related to Azure Disk Encryption. For more information about this service, see the following articles:

- Azure Disk Encryption Overview
- Apply disk encryption in Azure Security Center
- Azure data encryption at rest

Upgrading the Azure Disk Encryption version

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

The first version of Azure Disk Encryption (ADE) relied on Microsoft Entra ID for authentication; the current version doesn't. We strongly encourage the use of the newest version.

Determine ADE version

The versions of ADE in scope for migration are:

- **Windows**: 1.1.* (ADE on the VM must be upgraded to 2.2)
- **Linux**: 0.1.* (ADE on the VM must be upgraded to 1.2)

You can determine the version of ADE with which a VM was encrypted via Azure CLI, Azure PowerShell, or the Azure portal.

CLI

To determine the ADE version, run the Azure CLI az vm get-instance-view command.

Azure CLI

```
az vm get-instance-view --resource-group  <ResourceGroupName> --name <VMName>
```

Locate the AzureDiskEncryption extension in the output and identify the version number from the "TypeHandlerVersion" field in the output.

How to migrate

Migration from Azure Disk Encryption (with Microsoft Entra ID) to Azure Disk Encryption (without Microsoft Entra ID) is only available through Azure PowerShell. Ensure you have the latest version of Azure PowerShell and at least the Azure PowerShell Az module version 5.9.0 installed.

To upgrade from Azure Disk Encryption (with Microsoft Entra ID) to Azure Disk Encryption (without Microsoft Entra ID), use the Set-AzVMDiskEncryptionExtension PowerShell cmdlet.

> **⚠ Warning**
>
> The Set-AzVMDiskEncryptionExtension cmdlet must only be used on VMs encrypted with Azure Disk Encryption (with Microsoft Entra ID). Attempting to migrate an unencrypted VM, or a VM encrypted with Azure Disk Encryption (without Microsoft Entra ID), will result in a terminal error.

Azure PowerShell

```
Set-AzVMDiskEncryptionExtension -ResourceGroupName <resourceGroupName> -VMName
<vmName> -Migrate
```

When the cmdlet prompts you for confirmation, enter "Y". The ADE version is updated and the VM rebooted. The output looks similar to the following:

Output

```
Update AzureDiskEncryption version?
This cmdlet updates Azure Disk Encryption version to single pass (Azure Disk
Encryption without Azure AD). This may reboot
the machine and takes 10-15 minutes to finish. Are you sure you want to continue?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
Azure Disk Encryption Extension Public Settings
"KeyVaultResourceId": /subscriptions/ea500758-3163-4849-bd2c-
3e50f06efa7a/resourceGroups/myResourceGroup/providers/Microsoft.KeyVault/vaults/my
KeyVault
"SequenceVersion":
"MigrateFlag": Migrate
"KeyVaultURL": https://myKeyVault.vault.azure.net/
"Azure ADClientID": 00001111-aaaa-2222-bbbb-3333cccc4444
"KeyEncryptionKeyURL":
"KekVaultResourceId":
"EncryptionOperation": EnableEncryption
"Azure ADClientCertThumbprint":
"VolumeType":
"KeyEncryptionAlgorithm":

Running ADE extension (with Azure AD) for -Migrate..
ADE extension (with Azure AD) is now complete. Updating VM model..
```

```
Running ADE extension (without Azure AD) for -Migrate..
ADE extension (without Azure AD) is now complete. Clearing VM model..

RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

> **Important**
>
> The upgrade takes at least 10 - 15 minutes to complete. Don't cancel the cmdlet while the upgrade is in progress. Doing so puts the health of the VM at risk.

Next steps

- Azure Disk Encryption troubleshooting

Azure Disk Encryption with Microsoft Entra ID (previous release)

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The new release of Azure Disk Encryption eliminates the requirement for providing a Microsoft Entra application parameter to enable VM disk encryption. With the new release, you're no longer required to provide Microsoft Entra credentials during the enable encryption step. All new VMs must be encrypted without the Microsoft Entra application parameters by using the new release. For instructions on how to enable VM disk encryption by using the new release, see Azure Disk Encryption for Linux VMs. VMs that were already encrypted with Microsoft Entra application parameters are still supported and should continue to be maintained with the Microsoft Entra syntax.

This article provides supplements to Azure Disk Encryption for Linux VMs with additional requirements and prerequisites for Azure Disk Encryption with Microsoft Entra ID (previous release).

The information in these sections remains the same:

- Supported VMs and operating systems
- Additional VM requirements

Networking and Group Policy

To enable the Azure Disk Encryption feature by using the older Microsoft Entra parameter syntax, the infrastructure as a service (IaaS) VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the IaaS VM must be able to connect to a Microsoft Entra endpoint, [login.microsoftonline.com].

- To write the encryption keys to your key vault, the IaaS VM must be able to connect to the key vault endpoint.
- The IaaS VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see Azure Key Vault behind a firewall.
- On Windows, if TLS 1.0 is explicitly disabled and the .NET version isn't updated to 4.6 or higher, the following registry change enables Azure Disk Encryption to select the more recent TLS version:

config-registry

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001`
```

Group Policy

- The Azure Disk Encryption solution uses the BitLocker external key protector for Windows IaaS VMs. For domain-joined VMs, don't push any Group Policies that enforce TPM protectors. For information about the Group Policy for the option **Allow BitLocker without a compatible TPM**, see BitLocker Group Policy reference.

- BitLocker policy on domain-joined virtual machines with a custom Group Policy must include the following setting: Configure user storage of BitLocker recovery information -> Allow 256-bit recovery key. Azure Disk Encryption fails when custom Group Policy settings for BitLocker are incompatible. On machines that don't have the correct policy setting, apply the new policy, force the new policy to update (gpupdate.exe /force), and then restart if it's required.

Encryption key storage requirements

Azure Disk Encryption requires Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For more information, see Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release).

Next steps

- Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release)
- Enable Azure Disk Encryption with Microsoft Entra ID on Linux VMs (previous release)
- Azure Disk Encryption prerequisites CLI script
- Azure Disk Encryption prerequisites PowerShell script

Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release) for Linux VMs

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The new release of Azure Disk Encryption eliminates the requirement for providing a Microsoft Entra application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Microsoft Entra credentials during the enable encryption step. All new VMs must be encrypted without the Microsoft Entra application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see Azure Disk Encryption. VMs that were already encrypted with Microsoft Entra application parameters are still supported and should continue to be maintained with the Microsoft Entra syntax.

Azure Disk Encryption uses Azure Key Vault to control and manage disk encryption keys and secrets. For more information about key vaults, see Get started with Azure Key Vault and Secure your key vault.

Creating and configuring a key vault for use with Azure Disk Encryption with Microsoft Entra ID (previous release) involves three steps:

1. Create a key vault.
2. Set up a Microsoft Entra application and service principal.
3. Set the key vault access policy for the Microsoft Entra app.
4. Set key vault advanced access policies.

You may also, if you wish, generate or import a key encryption key (KEK).

See the main Creating and configuring a key vault for Azure Disk Encryption article for steps on how to Install tools and connect to Azure.

> **Note**
>
> The steps in this article are automated in the Azure Disk Encryption prerequisites CLI script and Azure Disk Encryption prerequisites PowerShell script.

Create a key vault

Azure Disk Encryption is integrated with Azure Key Vault to help you control and manage the disk-encryption keys and secrets in your key vault subscription. You can create a key vault or use an existing one for Azure Disk Encryption. For more information about key vaults, see Get started with Azure Key Vault and Secure your key vault. You can use a Resource Manager template, Azure PowerShell, or the Azure CLI to create a key vault.

> **Warning**
>
> In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Create a key vault with PowerShell

You can create a key vault with Azure PowerShell using the New-AzKeyVault cmdlet. For additional cmdlets for Key Vault, see Az.KeyVault.

1. Create a new resource group, if needed, with New-AzResourceGroup. To list data center locations, use Get-AzLocation.

   Azure PowerShell

   ```
   # Get-AzLocation
   New-AzResourceGroup -Name 'MyKeyVaultResourceGroup' -Location 'East US'
   ```

2. Create a new key vault using New-AzKeyVault

   Azure PowerShell

   ```
   New-AzKeyVault -VaultName 'MySecureVault' -ResourceGroupName
   'MyKeyVaultResourceGroup' -Location 'East US'
   ```

3. Note the **Vault Name**, **Resource Group Name**, **Resource ID**, **Vault URI**, and the **Object ID** that are returned for later use when you encrypt the disks.

Create a key vault with Azure CLI

You can manage your key vault with Azure CLI using the az keyvault commands. To create a key vault, use az keyvault create.

1. Create a new resource group, if needed, with az group create. To list locations, use az account list-locations

   Azure CLI

   ```
   # To list locations: az account list-locations --output table
   az group create -n "MyKeyVaultResourceGroup" -l "East US"
   ```

2. Create a new key vault using az keyvault create.

   Azure CLI

   ```
   az keyvault create --name "MySecureVault" --resource-group
   "MyKeyVaultResourceGroup" --location "East US"
   ```

3. Note the **Vault Name** (name), **Resource Group Name**, **Resource ID** (ID), **Vault URI**, and the **Object ID** that are returned for use later.

Create a key vault with a Resource Manager template

You can create a key vault by using the Resource Manager template.

1. On the Azure Quickstart Template, select **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then select **Purchase**.

Set up a Microsoft Entra app and service principal

When you need encryption to be enabled on a running VM in Azure, Azure Disk Encryption generates and writes the encryption keys to your key vault. Managing encryption keys in your key vault requires Microsoft Entra authentication. Create a Microsoft Entra application for this purpose. For authentication purposes, you can use either client secret-based authentication or client certificate-based Microsoft Entra authentication.

Set up a Microsoft Entra app and service principal with Azure PowerShell

To execute the following commands, get and use the Azure PowerShell module.

1. Use the New-AzADApplication PowerShell cmdlet to create a Microsoft Entra application. MyApplicationHomePage and the MyApplicationUri can be any values you wish.

   Azure PowerShell

   ```
   $aadClientSecret = "My AAD client secret"
   $aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -
   AsPlainText -Force
   $azureAdApplication = New-AzADApplication -DisplayName "My Application
   Display Name" -HomePage "https://MyApplicationHomePage" -IdentifierUris
   "https://MyApplicationUri" -Password $aadClientSecretSec
   $servicePrincipal = New-AzADServicePrincipal -ApplicationId
   $azureAdApplication.ApplicationId -Role Contributor
   ```

2. The $azureAdApplication.ApplicationId is the Microsoft Entra ClientID and the $aadClientSecret is the client secret that you'll use later to enable Azure Disk Encryption. Safeguard the Microsoft Entra client secret appropriately. Running `$azureAdApplication.ApplicationId` will show you the ApplicationID.

Set up a Microsoft Entra app and service principal with Azure CLI

You can manage your service principals with Azure CLI using the az ad sp commands. For more information, see Create an Azure service principal.

1. Create a new service principal.

   Azure CLI

   ```
   az ad sp create-for-rbac --name "ServicePrincipalName" --password "My-AAD-
   client-secret" --role Contributor --scopes /subscriptions/<subscription_id>
   ```

2. The appId returned is the Microsoft Entra ClientID used in other commands. It's also the SPN you'll use for az keyvault set-policy. The password is the client secret that you should use later to enable Azure Disk Encryption. Safeguard the Microsoft Entra client secret appropriately.

Set up a Microsoft Entra app and service principal through the Azure portal

Use the steps from the Use portal to create a Microsoft Entra application and service principal that can access resources article to create a Microsoft Entra application. Each step listed below will take you directly to the article section to complete.

1. Verify required permissions
2. Create a Microsoft Entra application

    - You can use any name and sign-on URL you would like when creating the application.

3. Get the application ID and the authentication key.

    - The authentication key is the client secret and is used as the AadClientSecret for Set-AzVMDiskEncryptionExtension.
        - The authentication key is used by the application as a credential to sign in to Microsoft Entra ID. In the Azure portal, this secret is called keys, but has no relation to key vaults. Secure this secret appropriately.
    - The application ID will be used later as the AadClientId for Set-AzVMDiskEncryptionExtension and as the ServicePrincipalName for Set-AzKeyVaultAccessPolicy.

Set the key vault access policy for the Microsoft Entra app

To write encryption secrets to a specified Key Vault, Azure Disk Encryption needs the Client ID and the Client Secret of the Microsoft Entra application that has permissions to write secrets to the Key Vault.

> **Note**
>
> Azure Disk Encryption requires you to configure the following access policies to your Microsoft Entra client application: *WrapKey* and *Set* permissions.

Set the key vault access policy for the Microsoft Entra app with Azure PowerShell

Your Microsoft Entra application needs rights to access the keys or secrets in the vault. Use the Set-AzKeyVaultAccessPolicy cmdlet to grant permissions to the application, using the client ID (which was generated when the application was registered) as the *–ServicePrincipalName* parameter value. To learn more, see the blog post Azure Key Vault - Step by Step.

1. Set the key vault access policy for the AD application with PowerShell.

Azure PowerShell

```
$keyVaultName = 'MySecureVault'
$aadClientID = 'MyAadAppClientID'
$KVRGname = 'MyKeyVaultResourceGroup'
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName $KVRGname
```

Set the key vault access policy for the Microsoft Entra app with Azure CLI

Use az keyvault set-policy to set the access policy. For more information, see Manage Key Vault using CLI 2.0.

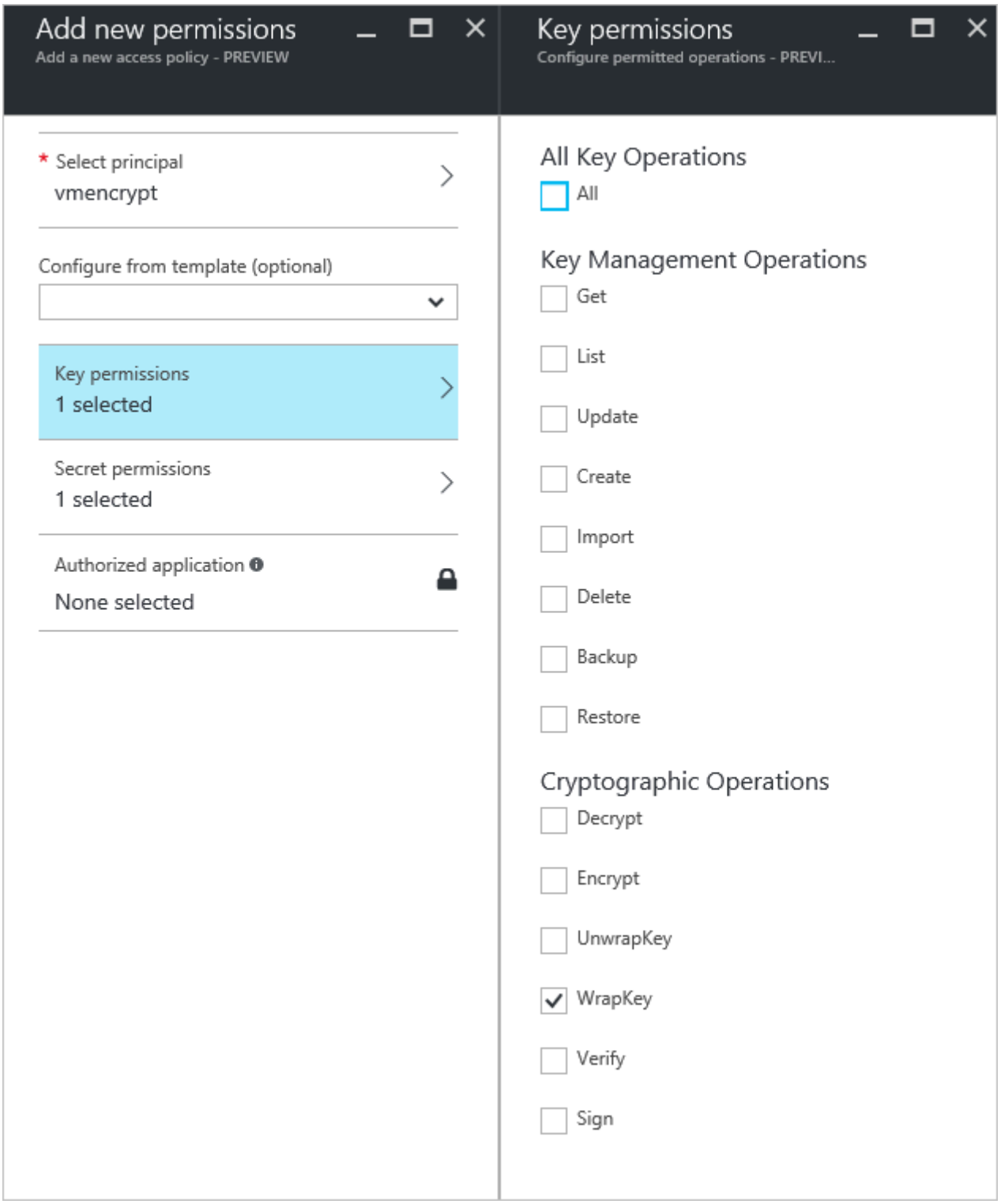
Give the service principal you created via the Azure CLI access to get secrets and wrap keys with the following command:

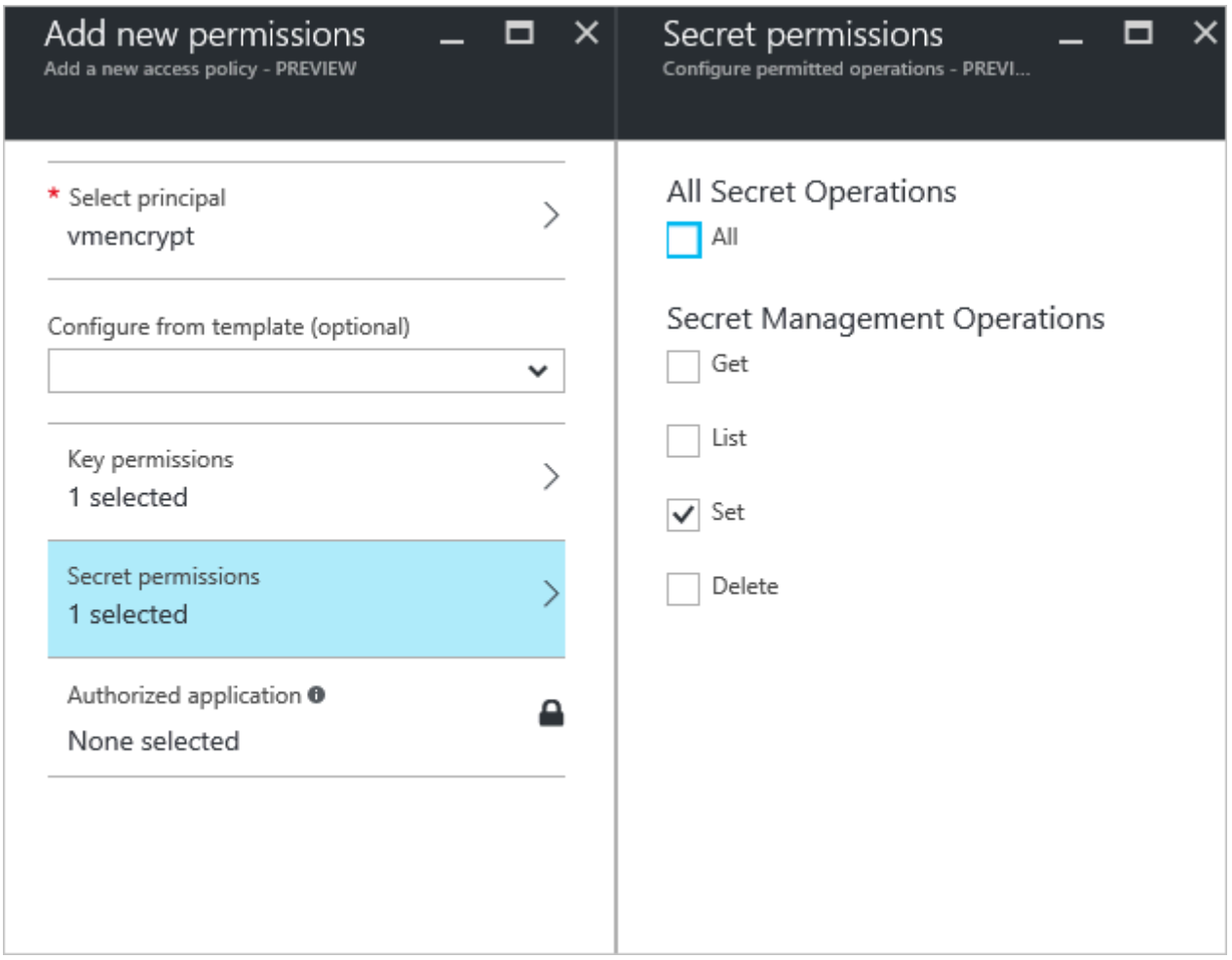
Azure CLI

```
az keyvault set-policy --name "MySecureVault" --spn "<spn created with CLI/the
Azure AD ClientID>" --key-permissions wrapKey --secret-permissions set
```

Set the key vault access policy for the Microsoft Entra app with the portal

1. Open the resource group with your key vault.
2. Select your key vault, go to **Access Policies**, then select **Add new**.
3. Under **Select principal**, search for the Microsoft Entra application you created and select it.
4. For **Key permissions**, check **Wrap Key** under **Cryptographic Operations**.
5. For **Secret permissions**, check **Set** under **Secret Management Operations**.
6. Select **OK** to save the access policy.


Add new permissions
Add a new access policy - PREVIEW
* Select principal
vmencrypt
Configure from template (optional)
Key permissions
1 selected
Secret permissions
1 selected
Authorized application
None selected
Key permissions
Configure permitted operations - PREVI...
All Key Operations
All
Key Management Operations
Get
List
Update
Create
Import
Delete
Backup
Restore
Cryptographic Operations
Decrypt
Encrypt
UnwrapKey
WrapKey
Verify
Sign



Set key vault advanced access policies

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the VM for booting and decrypting the volumes. Enable disk encryption on the key vault or deployments will fail.

Set key vault advanced access policies with Azure PowerShell

Use the key vault PowerShell cmdlet Set-AzKeyVaultAccessPolicy to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

  Azure PowerShell

  ```
  Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName
  'MyKeyVaultResourceGroup' -EnabledForDiskEncryption
  ```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft. Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in

resource creation, for example when creating a virtual machine.

Azure PowerShell

```
 Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName
'MyKeyVaultResourceGroup' -EnabledForDeployment
```

- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

Azure PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName
'MyKeyVaultResourceGroup' -EnabledForTemplateDeployment
```

Set key vault advanced access policies using the Azure CLI

Use az keyvault update to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

Azure CLI

```
az keyvault update --name "MySecureVault" --resource-group
"MyKeyVaultResourceGroup" --enabled-for-disk-encryption "true"
```

- **Enable Key Vault for deployment, if needed:** Allow Virtual Machines to retrieve certificates stored as secrets from the vault.

Azure CLI

```
az keyvault update --name "MySecureVault" --resource-group
"MyKeyVaultResourceGroup" --enabled-for-deployment "true"
```

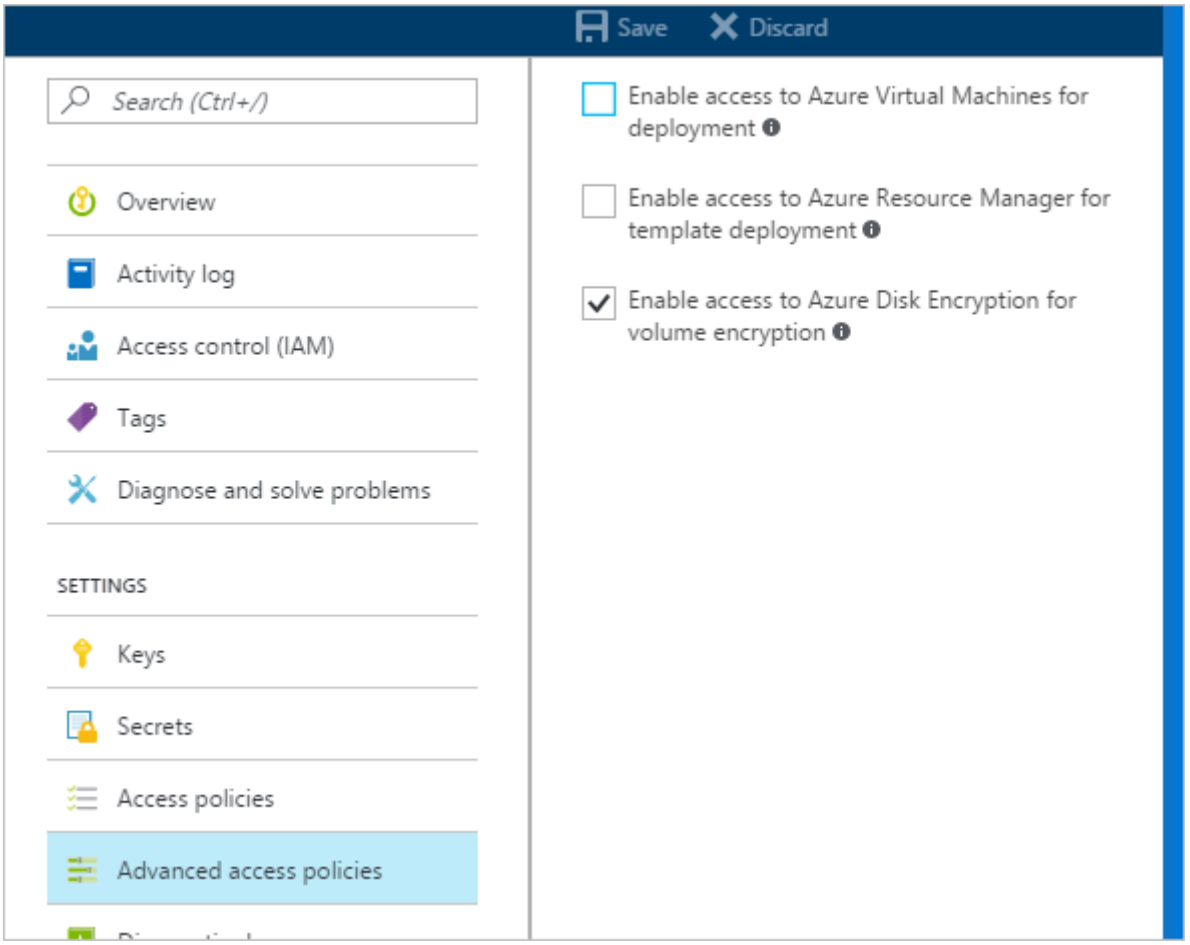
- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

Azure CLI

```
az keyvault update --name "MySecureVault" --resource-group
"MyKeyVaultResourceGroup" --enabled-for-template-deployment "true"
```

Set key vault advanced access policies through the Azure portal

1. Select your keyvault, go to **Access Policies**, and **Click to show advanced access policies**.
2. Select the box labeled **Enable access to Azure Disk Encryption for volume encryption**.
3. Select **Enable access to Azure Virtual Machines for deployment** and/or **Enable Access to Azure Resource Manager for template deployment**, if needed.
4. Select **Save**.



Set up a key encryption key (optional)

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. Use the Add-AzKeyVaultKey cmdlet to create a key encryption key in the key vault. You can also import a KEK from your on-premises key management HSM. For more information, see Key Vault Documentation. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

- When generating keys, use an RSA key type. Azure Disk Encryption doesn't yet support using Elliptic Curve keys.

- Your key vault secret and KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:
  - Example of a valid secret URL:
    `https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx`
  - Example of a valid KEK URL:
    `https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxx xxxxxxxxxx`

- Azure Disk Encryption doesn't support specifying port numbers as part of key vault secrets and KEK URLs. For examples of non-supported and supported key vault URLs, see the following examples:
  - Unacceptable key vault URL
    `https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxx xxxxxxxxxxxxx`
  - Acceptable key vault URL:
    `https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxx xxxxxxxxx`

Set up a key encryption key with Azure PowerShell

Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment. This script creates all Azure Disk Encryption prerequisites and encrypts an existing IaaS VM, wrapping the disk encryption key by using a key encryption key.

PowerShell

```
# Step 1: Create a new resource group and key vault in the same location.
     # Fill in 'MyLocation', 'MyKeyVaultResourceGroup', and 'MySecureVault' with
your values.
     # Use Get-AzLocation to get available locations and use the DisplayName.
     # To use an existing resource group, comment out the line for New-
AzResourceGroup

    $Loc = 'MyLocation';
    $KVRGname = 'MyKeyVaultResourceGroup';
    $KeyVaultName = 'MySecureVault';
    New-AzResourceGroup –Name  $KVRGname –Location $Loc;
    New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location
$Loc;
    $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
    $KeyVaultResourceId = (Get-AzKeyVault -VaultName $KeyVaultName -
ResourceGroupName  $KVRGname).ResourceId;
```

```
    $diskEncryptionKeyVaultUrl = (Get-AzKeyVault -VaultName $KeyVaultName -
ResourceGroupName  $KVRGname).VaultUri;

# Step 2: Create the AD application and service principal.
     # Fill in 'MyAADClientSecret', "<My Application Display Name>", "
<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your
values.
     # MyApplicationHomePage and the MyApplicationUri can be any values you wish.

     $aadClientSecret =  'MyAADClientSecret';
    $aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -
AsPlainText -Force;
    $azureAdApplication = New-AzADApplication -DisplayName "<My Application
Display Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "
<https://MyApplicationUri>" -Password $aadClientSecretSec
    $servicePrincipal = New-AzADServicePrincipal –ApplicationId
$azureAdApplication.ApplicationId -Role Contributor;
    $aadClientID = $azureAdApplication.ApplicationId;

#Step 3: Enable the vault for disk encryption and set the access policy for the
Microsoft Entra application.

     Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$KVRGname -EnabledForDiskEncryption;
    Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName  $KVRGname;

#Step 4: Create a new key in the key vault with the Add-AzKeyVaultKey cmdlet.
     # Fill in 'MyKeyEncryptionKey' with your value.

     $keyEncryptionKeyName = 'MyKeyEncryptionKey';
    Add-AzKeyVaultKey -VaultName $KeyVaultName -Name $keyEncryptionKeyName -
Destination 'Software';
    $keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;

#Step 5: Encrypt the disks of an existing IaaS VM
     # Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

     $VMName = 'MySecureVM';
     $VMRGName = 'MyVirtualMachineResourceGroup';
    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -
AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $KeyVaultResourceId;
```

Certificate-based authentication (optional)

If you would like to use certificate authentication, you can upload one to your key vault and deploy it to the client. Before using the PowerShell script, you should be familiar with the Azure

Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

PowerShell

```
# Fill in "MyKeyVaultResourceGroup", "MySecureVault", and 'MyLocation' ('My
location' only if needed)

  $KVRGname = 'MyKeyVaultResourceGroup'
  $KeyVaultName= 'MySecureVault'

  # Create a key vault and set enabledForDiskEncryption property on it.
  # Comment out the next three lines if you already have an existing key vault
enabled for encryption. No need to set 'My location' in this case.

  $Loc = 'MyLocation'
  New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location
$Loc
  Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $KVRGname
-EnabledForDiskEncryption

  #Setting some variables with the key vault information
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname
  $DiskEncryptionKeyVaultUrl = $KeyVault.VaultUri
  $KeyVaultResourceId = $KeyVault.ResourceId

  # Create the Microsoft Entra application and associate the certificate with it.
  # Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display
Name>", "<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with
your values.
  # MyApplicationHomePage and the MyApplicationUri can be any values you wish

  $CertPath = "C:\certificates\mycert.pfx"
  $CertPassword = "Password"
  $Cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath,
$CertPassword)
  $CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

  $AzureAdApplication = New-AzADApplication -DisplayName "<My Application Display
Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "
<https://MyApplicationUri>" -CertValue $CertValue
  $ServicePrincipal = New-AzADServicePrincipal -ApplicationId
$AzureAdApplication.ApplicationId -Role Contributor

  $AADClientID = $AzureAdApplication.ApplicationId
  $aadClientCertThumbprint= $cert.Thumbprint

  Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName $KVRGname

  # Upload the pfx file to the key vault.
```

```
  # Fill in "MyAADCert".

  $KeyVaultSecretName = "MyAADCert"
  $FileContentBytes = get-content $CertPath -Encoding Byte
  $FileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
          $JSONObject = @"
          {
              "data" : "$filecontentencoded",
              "dataType" : "pfx",
              "password" : "$CertPassword"
          }
"@

  $JSONObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
  $JSONEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

  #Set the secret and set the key vault policy for -EnabledForDeployment

  $Secret = ConvertTo-SecureString -String $JSONEncoded -AsPlainText -Force
  Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $KeyVaultSecretName -
SecretValue $Secret
  Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $KVRGname
-EnabledForDeployment

  # Deploy the certificate to the VM
  # Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

  $VMName = 'MySecureVM'
  $VMRGName = 'MyVirtualMachineResourceGroup'
  $CertUrl = (Get-AzKeyVaultSecret -VaultName $KeyVaultName -Name
$KeyVaultSecretName).Id
  $SourceVaultId = (Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGName).ResourceId
  $VM = Get-AzVM -ResourceGroupName $VMRGName -Name $VMName
  $VM = Add-AzVMSecret -VM $VM -SourceVaultId $SourceVaultId -CertificateStore
"My" -CertificateUrl $CertUrl
  Update-AzVM -VM $VM -ResourceGroupName $VMRGName

  #Enable encryption on the VM using Microsoft Entra client ID and the client
certificate thumbprint

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -
AadClientID $AADClientID -AadClientCertThumbprint $AADClientCertThumbprint -
DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId
```

Certificate-based authentication and a KEK (optional)

If you would like to use certificate authentication and wrap the encryption key with a KEK, you can use the below script as an example. Before using the PowerShell script, you should be

familiar with all of the previous Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

> **Important**
>
> Microsoft Entra certificate-based authentication is currently not supported on Linux VMs.

PowerShell

```
# Fill in 'MyKeyVaultResourceGroup', 'MySecureVault', and 'MyLocation' (if needed)

   $KVRGname = 'MyKeyVaultResourceGroup'
   $KeyVaultName= 'MySecureVault'

   # Create a key vault and set enabledForDiskEncryption property on it.
   # Comment out the next three lines if you already have an existing key vault
enabled for encryption.

   $Loc = 'MyLocation'
   New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location
$Loc
   Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$KVRGname -EnabledForDiskEncryption

   # Create the Azure AD application and associate the certificate with it.
   # Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display
Name>", "<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with
your values.
   # MyApplicationHomePage and the MyApplicationUri can be any values you wish

   $CertPath = "C:\certificates\mycert.pfx"
   $CertPassword = "Password"
   $Cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath,
$CertPassword)
   $CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

   $AzureAdApplication = New-AzADApplication -DisplayName "<My Application Display
Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "
<https://MyApplicationUri>" -CertValue $CertValue
   $ServicePrincipal = New-AzADServicePrincipal -ApplicationId
$AzureAdApplication.ApplicationId -Role Contributor

   $AADClientID = $AzureAdApplication.ApplicationId
   $aadClientCertThumbprint= $cert.Thumbprint

   ## Give access for setting secrets and wraping keys
   Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName $KVRGname

   # Upload the pfx file to the key vault.
```

```
   # Fill in "MyAADCert".

   $KeyVaultSecretName = "MyAADCert"
   $FileContentBytes = get-content $CertPath -Encoding Byte
   $FileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
           $JSONObject = @"
           {
               "data" : "$filecontentencoded",
               "dataType" : "pfx",
               "password" : "$CertPassword"
           }
"@

   $JSONObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
   $JSONEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

   #Set the secret and set the key vault policy for deployment

   $Secret = ConvertTo-SecureString -String $JSONEncoded -AsPlainText -Force
   Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $KeyVaultSecretName -
SecretValue $Secret
   Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$KVRGname -EnabledForDeployment

   #Setting some variables with the key vault information and generating a KEK
   # FIll in 'KEKName'

   $KEKName ='KEKName'
   $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname
   $DiskEncryptionKeyVaultUrl = $KeyVault.VaultUri
   $KeyVaultResourceId = $KeyVault.ResourceId
   $KEK = Add-AzKeyVaultKey -VaultName $KeyVaultName -Name $KEKName -Destination
"Software"
   $KeyEncryptionKeyUrl = $KEK.Key.kid



   # Deploy the certificate to the VM
   # Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

   $VMName = 'MySecureVM';
   $VMRGName = 'MyVirtualMachineResourceGroup';
   $CertUrl = (Get-AzKeyVaultSecret -VaultName $KeyVaultName -Name
$KeyVaultSecretName).Id
   $SourceVaultId = (Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGName).ResourceId
   $VM = Get-AzVM -ResourceGroupName $VMRGName -Name $VMName
   $VM = Add-AzVMSecret -VM $VM -SourceVaultId $SourceVaultId -CertificateStore
"My" -CertificateUrl $CertUrl
   Update-AzVM -VM $VM -ResourceGroupName $VMRGName

   #Enable encryption on the VM using Azure AD client ID and the client
certificate thumbprint
```

```
    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -
AadClientID $AADClientID -AadClientCertThumbprint $AADClientCertThumbprint -
DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $KeyVaultResourceId
```

Next steps

Enable Azure Disk Encryption with Microsoft Entra ID on Linux VMs (previous release)

Enable Azure Disk Encryption with Microsoft Entra ID on Linux VMs (previous release)

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The new release of Azure Disk Encryption eliminates the requirement for providing a Microsoft Entra application parameter to enable VM disk encryption. With the new release, you're no longer required to provide Microsoft Entra credentials during the enable encryption step. All new VMs must be encrypted without the Microsoft Entra application parameters by using the new release. For instructions on how to enable VM disk encryption by using the new release, see Azure Disk Encryption for Linux VMS. VMs that were already encrypted with Microsoft Entra application parameters are still supported and should continue to be maintained with the Microsoft Entra syntax.

You can enable many disk-encryption scenarios, and the steps might vary according to the scenario. The following sections cover the scenarios in greater detail for Linux infrastructure as a service (IaaS) VMs. You can only apply disk encryption to virtual machines of supported VM sizes and operating systems. You must also meet the following prerequisites:

- Additional requirements for VMs
- Networking and Group Policy
- Encryption key storage requirements

Take a snapshot, make a backup, or both before you encrypt the disks. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. After a backup is made, you can use the Set-AzVMDiskEncryptionExtension cmdlet to encrypt managed disks by specifying the -skipVmBackup parameter. For more information about how to back up and restore encrypted VMs, see Azure Backup.

> ⚠ **Warning**
>
> - If you previously used Azure Disk Encryption with the Microsoft Entra app to encrypt this VM, you must continue to use this option to encrypt your VM. You can't use Azure Disk Encryption on this encrypted VM because this isn't a supported scenario, which means switching away from the Microsoft Entra application for this encrypted VM isn't supported yet.
> - To make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the key vault and the VMs to be colocated in the same region. Create and use a key vault that's in the same region as the VM to be encrypted.
> - When you encrypt Linux OS volumes, the process can take a few hours. It's normal for Linux OS volumes to take longer than data volumes to encrypt.
> - When you encrypt Linux OS volumes, the VM should be considered unavailable. We strongly recommend that you avoid SSH logins while the encryption is in progress to avoid blocking any open files that need to be accessed during the encryption process. To check progress, use the Get-AzVMDiskEncryptionStatus or vm encryption show commands. You can expect this process to take a few hours for a 30-GB OS volume, plus more time for encrypting data volumes. Data volume encryption time is proportional to the size and quantity of the data volumes unless the **encrypt format all** option is used.
> - Disabling encryption on Linux VMs is only supported for data volumes. It's not supported on data or OS volumes if the OS volume has been encrypted.

Enable encryption on an existing or running IaaS Linux VM

In this scenario, you can enable encryption by using the Azure Resource Manager template, PowerShell cmdlets, or Azure CLI commands.

> ⓘ **Important**
>
> It's mandatory to take a snapshot or back up a managed disk-based VM instance outside of and prior to enabling Azure Disk Encryption. You can take a snapshot of the managed disk from the Azure portal, or you can use Azure Backup. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. After a backup is made, use the Set-AzVMDiskEncryptionExtension cmdlet to encrypt managed disks by

specifying the -skipVmBackup parameter. The Set-AzVMDiskEncryptionExtension command fails against managed disk-based VMs until a backup is made and this parameter is specified.

Encrypting or disabling encryption might cause the VM to reboot.

Enable encryption on an existing or running Linux VM by using the Azure CLI

You can enable disk encryption on your encrypted VHD by installing and using the Azure CLI 2.0 command-line tool. You can use it in your browser with Azure Cloud Shell, or you can install it on your local machine and use it in any PowerShell session. To enable encryption on existing or running IaaS Linux VMs in Azure, use the following CLI commands:

Use the az vm encryption enable command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM by using a client secret:**

  Azure CLI

  ```
  az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type [All|OS|Data]
  ```

- **Encrypt a running VM by using KEK to wrap the client secret:**

  Azure CLI

  ```
  az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault  "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
  ```

  > **Note**
  >
  > The syntax for the value of the disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name].

The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id].

- **Verify that the disks are encrypted:** To check on the encryption status of an IaaS VM, use the az vm encryption show command.

Azure CLI

```
    az vm encryption show --name "MySecureVM" --resource-group
"MyVirtualMachineResourceGroup"
```

- **Disable encryption:** To disable encryption, use the az vm encryption disable command. Disabling encryption is only allowed on data volumes for Linux VMs.

Azure CLI

```
    az vm encryption disable --name "MySecureVM" --resource-group
"MyVirtualMachineResourceGroup" --volume-type DATA
```

Enable encryption on an existing or running Linux VM by using PowerShell

Use the Set-AzVMDiskEncryptionExtension cmdlet to enable encryption on a running IaaS virtual machine in Azure. Take a snapshot or make a backup of the VM with Azure Backup before the disks are encrypted. The -skipVmBackup parameter is already specified in the PowerShell scripts to encrypt a running Linux VM.

- **Encrypt a running VM by using a client secret:** The following script initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, Microsoft Entra app, and client secret should have already been created as prerequisites. Replace MyVirtualMachineResourceGroup, MyKeyVaultResourceGroup, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values. Modify the -VolumeType parameter to specify which disks you're encrypting.

Azure PowerShell

```
    $VMRGName = 'MyVirtualMachineResourceGroup';
    $KVRGname = 'MyKeyVaultResourceGroup';
    $vmName = 'MySecureVM';
    $aadClientID = 'My-AAD-client-ID';
    $aadClientSecret = 'My-AAD-client-secret';
    $KeyVaultName = 'MySecureVault';
    $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
```

```
    $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
    $KeyVaultResourceId = $KeyVault.ResourceId;
    $sequenceVersion = [Guid]::NewGuid();

    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName
$vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -VolumeType '[All|OS|Data]' -
SequenceVersion $sequenceVersion -skipVmBackup;
```

- **Encrypt a running VM by using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to the key vault. Modify the -VolumeType parameter to specify which disks you're encrypting.

Azure PowerShell

```
    $KVRGname = 'MyKeyVaultResourceGroup';
    $VMRGName = 'MyVirtualMachineResourceGroup';
    $aadClientID = 'My-AAD-client-ID';
    $aadClientSecret = 'My-AAD-client-secret';
    $KeyVaultName = 'MySecureVault';
    $keyEncryptionKeyName = 'MyKeyEncryptionKey';
    $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
    $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
    $KeyVaultResourceId = $KeyVault.ResourceId;
    $keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;
    $sequenceVersion = [Guid]::NewGuid();

    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName
$vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -KeyEncryptionKeyUrl
$keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $KeyVaultResourceId -VolumeType
'[All|OS|Data]' -SequenceVersion $sequenceVersion -skipVmBackup;
```

> **Note**
>
> The syntax for the value of the disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[KVresource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name].
>
> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id].

- **Verify that the disks are encrypted:** To check on the encryption status of an IaaS VM, use the Get-AzVmDiskEncryptionStatus cmdlet.

  Azure PowerShell

  ```
  Get-AzVmDiskEncryptionStatus -ResourceGroupName
  MyVirtualMachineResourceGroup -VMName MySecureVM
  ```

- **Disable disk encryption:** To disable the encryption, use the Disable-AzureRmVMDiskEncryption cmdlet. Disabling encryption is only allowed on data volumes for Linux VMs.

  Azure PowerShell

  ```
  Disable-AzVMDiskEncryption -ResourceGroupName
  'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
  ```

Enable encryption on an existing or running IaaS Linux VM with a template

You can enable disk encryption on an existing or running IaaS Linux VM in Azure by using the Resource Manager template.

1. Select **Deploy to Azure** on the Azure quickstart template.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Select **Create** to enable encryption on the existing or running IaaS VM.

The following table lists Resource Manager template parameters for existing or running VMs that use a Microsoft Entra client ID:

Expand table

Parameter	Description
AADClientID	Client ID of the Microsoft Entra application that has permissions to write secrets to the key vault.
AADClientSecret	Client secret of the Microsoft Entra application that has permissions to write secrets to your key vault.
keyVaultName	Name of the key vault that the key should be uploaded to. You can get it by using the Azure CLI command `az keyvault show --name "MySecureVault" --query KVresourceGroup`.
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated key. This parameter is optional if you select **nokek** in the **UseExistingKek** drop-down list. If

Parameter	Description
	you select **kek** in the **UseExistingKek** drop-down list, you must enter the *keyEncryptionKeyURL* value.
volumeType	Type of volume that the encryption operation is performed on. Valid supported values are *OS* or *All*. (See supported Linux distributions and their versions for OS and data disks in the prerequisites section earlier.)
sequenceVersion	Sequence version of the BitLocker operation. Increment this version number every time a disk-encryption operation is performed on the same VM.
vmName	Name of the VM that the encryption operation is to be performed on.
passphrase	Type a strong passphrase as the data encryption key.

Use the EncryptFormatAll feature for data disks on Linux IaaS VMs

The EncryptFormatAll parameter reduces the time for Linux data disks to be encrypted. Partitions that meet certain criteria are formatted (with their current file system). Then they're remounted back to where they were before command execution. If you want to exclude a data disk that meets the criteria, you can unmount it before you run the command.

After you run this command, any drives that were mounted previously are formatted. Then the encryption layer starts on top of the now empty drive. When this option is selected, the temporary disk attached to the VM is also encrypted. If the ephemeral drive is reset, it's reformatted and re-encrypted for the VM by the Azure Disk Encryption solution at the next opportunity.

> **⚠ Warning**
>
> EncryptFormatAll shouldn't be used when there's needed data on a VM's data volumes. You can exclude disks from encryption by unmounting them. Try out the EncryptFormatAll parameter on a test VM first to understand the feature parameter and its implication before you try it on the production VM. The EncryptFormatAll option formats the data disk, so all the data on it will be lost. Before you proceed, verify that any disks you want to exclude are properly unmounted.
>
> If you set this parameter while you update encryption settings, it might lead to a reboot before the actual encryption. In this case, you also want to remove the disk you don't want formatted from the fstab file. Similarly, you should add the partition you want encrypt-formatted to the fstab file before you initiate the encryption operation.

EncryptFormatAll criteria

The parameter goes through all partitions and encrypts them as long as they meet *all* of the following criteria:

- Is not a root/OS/boot partition
- Is not already encrypted
- Is not a BEK volume
- Is not a RAID volume
- Is not an LVM volume
- Is mounted

Encrypt the disks that compose the RAID or LVM volume rather than the RAID or LVM volume.

Use the EncryptFormatAll parameter with a template

To use the EncryptFormatAll option, use any preexisting Azure Resource Manager template that encrypts a Linux VM and change the **EncryptionOperation** field for the AzureDiskEncryption resource.

1. As an example, use the Resource Manager template to encrypt a running Linux IaaS VM.
2. Select **Deploy to Azure** on the Azure quickstart template.
3. Change the **EncryptionOperation** field from **EnableEncryption** to **EnableEncryptionFormatAl**.
4. Select the subscription, resource group, resource group location, other parameters, legal terms, and agreement. Select **Create** to enable encryption on the existing or running IaaS VM.

Use the EncryptFormatAll parameter with a PowerShell cmdlet

Use the Set-AzVMDiskEncryptionExtension cmdlet with the EncryptFormatAll parameter.

Encrypt a running VM by using a client secret and EncryptFormatAll: As an example, the following script initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet with the EncryptFormatAll parameter. The resource group, VM, key vault, Microsoft Entra app, and client secret should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values.

Azure PowerShell

```
  $KVRGname = 'MyKeyVaultResourceGroup';
  $VMRGName = 'MyVirtualMachineResourceGroup';
  $aadClientID = 'My-AAD-client-ID';
  $aadClientSecret = 'My-AAD-client-secret';
  $KeyVaultName = 'MySecureVault';
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
  $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
  $KeyVaultResourceId = $KeyVault.ResourceId;

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -
AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -EncryptFormatAll
```

Use the EncryptFormatAll parameter with Logical Volume Manager (LVM)

We recommend an LVM-on-crypt setup. For all the following examples, replace the device-path and mountpoints with whatever suits your use case. This setup can be done as follows:

- Add the data disks that will compose the VM.
- Format, mount, and add these disks to the fstab file.

    1. Format the newly added disk. We use symlinks generated by Azure here. Using symlinks avoids problems related to device names changing. For more information, see Troubleshoot device names problems.

        Console

        ```
        mkfs -t ext4 /dev/disk/azure/scsi1/lun0
        ```

    2. Mount the disks.

        Console

        ```
        mount /dev/disk/azure/scsi1/lun0 /mnt/mountpoint
        ```

    3. Add to fstab.

        Console

        ```
        echo "/dev/disk/azure/scsi1/lun0 /mnt/mountpoint ext4 defaults,nofail 1
        2" >> /etc/fstab
        ```

4. Run the Set-AzVMDiskEncryptionExtension PowerShell cmdlet with -EncryptFormatAll to encrypt these disks.

   Azure PowerShell

   ```
   Set-AzVMDiskEncryptionExtension -ResourceGroupName "MySecureGroup" -VMName "MySecureVM" -DiskEncryptionKeyVaultUrl "https://mykeyvault.vault.azure.net/" -EncryptFormatAll
   ```

5. Set up LVM on top of these new disks. Note the encrypted drives are unlocked after the VM has finished booting. So, the LVM mounting will also have to be subsequently delayed.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using the Resource Manager template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail the Resource Manager template and CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- Prepare a pre-encrypted Linux VHD

> **ⓘ Important**
>
> It's mandatory to take a snapshot or back up a managed disk-based VM instance outside of and prior to enabling Azure Disk Encryption. You can take a snapshot of the managed disk from the portal, or you can use Azure Backup. Backups ensure that a recovery option is possible in the case of any unexpected failure during encryption. After a backup is made, use the Set-AzVMDiskEncryptionExtension cmdlet to encrypt managed disks by specifying the -skipVmBackup parameter. The Set-AzVMDiskEncryptionExtension command fails against managed disk-based VMs until a backup is made and this parameter is specified.
>
> Encrypting or disabling encryption might cause the VM to reboot.

Use Azure PowerShell to encrypt IaaS VMs with pre-encrypted VHDs

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet Set-AzVMOSDisk. The following example gives you some common parameters.

PowerShell

```
$VirtualMachine = New-AzVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri
"os.vhd" Caching ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaa
a" -DiskEncryptionKeyVaultId "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myresourcegroup/providers/Microsoft.KeyVault/vaults/my
testvault"
New-AzVM -VM $VirtualMachine -ResourceGroupName "MyVirtualMachineResourceGroup"
```

Enable encryption on a newly added data disk

You can add a new data disk by using az vm disk attach or through the Azure portal. Before you can encrypt, you need to mount the newly attached data disk first. You must request encryption of the data drive because the drive will be unusable while encryption is in progress.

Enable encryption on a newly added disk with the Azure CLI

If the VM was previously encrypted with "All," then the --volume-type parameter should remain All. All includes both OS and data disks. If the VM was previously encrypted with a volume type of "OS," then the --volume-type parameter should be changed to All so that both the OS and the new data disk will be included. If the VM was encrypted with only the volume type of "Data," then it can remain Data as demonstrated here. Adding and attaching a new data disk to a VM isn't sufficient preparation for encryption. The newly attached disk must also be formatted and properly mounted within the VM before you enable encryption. On Linux, the disk must be mounted in /etc/fstab with a persistent block device name.

In contrast to PowerShell syntax, the CLI doesn't require you to provide a unique sequence version when you enable encryption. The CLI automatically generates and uses its own unique sequence version value.

- **Encrypt a running VM by using a client secret:**

  Azure CLI

  ```
      az vm encryption enable --resource-group "MyVirtualMachineResourceGroup"
  --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD
  ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-encryption-
  keyvault "MySecureVault" --volume-type "Data"
  ```

- **Encrypt a running VM by using KEK to wrap the client secret:**

Azure CLI

```
    az vm encryption enable --resource-group "MyVirtualMachineResourceGroup"
--name "MySecureVM" --aad-client-id "<my spn created with CLI which is the
Azure AD ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-
encryption-keyvault  "MySecureVault" --key-encryption-key "MyKEK_URI" --key-
encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "Data"
```

Enable encryption on a newly added disk with Azure PowerShell

When you use PowerShell to encrypt a new disk for Linux, a new sequence version needs to be specified. The sequence version has to be unique. The following script generates a GUID for the sequence version.

- **Encrypt a running VM by using a client secret:** The following script initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, Microsoft Entra app, and client secret should have already been created as prerequisites. Replace MyVirtualMachineResourceGroup, MyKeyVaultResourceGroup, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values. The -VolumeType parameter is set to data disks and not the OS disk. If the VM was previously encrypted with a volume type of "OS" or "All," then the -VolumeType parameter should be changed to All so that both the OS and the new data disk will be included.

Azure PowerShell

```
    $KVRGname = 'MyKeyVaultResourceGroup';
    $VMRGName = 'MyVirtualMachineResourceGroup';
    $vmName = 'MySecureVM';
    $aadClientID = 'My-AAD-client-ID';
    $aadClientSecret = 'My-AAD-client-secret';
    $KeyVaultName = 'MySecureVault';
    $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
    $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
    $KeyVaultResourceId = $KeyVault.ResourceId;
    $sequenceVersion = [Guid]::NewGuid();

    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName
$vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -VolumeType 'data' –
SequenceVersion $sequenceVersion;
```

- **Encrypt a running VM by using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to the key vault. The -VolumeType parameter is set to data disks and not the OS disk. If the VM was previously encrypted with a volume type of "OS" or "All," then the -VolumeType parameter should be changed to All so that both the OS and the new data disk will be included.

Azure PowerShell

```
    $KVRGname = 'MyKeyVaultResourceGroup';
    $VMRGName = 'MyVirtualMachineResourceGroup';
    $vmName = 'MyExtraSecureVM';
    $aadClientID = 'My-AAD-client-ID';
    $aadClientSecret = 'My-AAD-client-secret';
    $KeyVaultName = 'MySecureVault';
    $keyEncryptionKeyName = 'MyKeyEncryptionKey';
    $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
    $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
    $KeyVaultResourceId = $KeyVault.ResourceId;
    $keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;
    $sequenceVersion = [Guid]::NewGuid();

    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName
$vmName -AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -KeyEncryptionKeyUrl
$keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $KeyVaultResourceId -VolumeType
'data' –SequenceVersion $sequenceVersion;
```

> **Note**
>
> The syntax for the value of the disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name].
>
> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id].

Disable encryption for Linux VMs

You can disable encryption by using Azure PowerShell, the Azure CLI, or a Resource Manager template.

> **Important**
>
> Disabling encryption with Azure Disk Encryption on Linux VMs is only supported for data volumes. It's not supported on data or OS volumes if the OS volume has been encrypted.

- **Disable disk encryption with Azure PowerShell:** To disable encryption, use the Disable-AzureRmVMDiskEncryption cmdlet.

  Azure PowerShell

  ```
  Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup' -VMName 'MySecureVM' [--volume-type {ALL, DATA, OS}]
  ```

- **Disable encryption with the Azure CLI:** To disable encryption, use the az vm encryption disable command.

  Azure CLI

  ```
  az vm encryption disable --name "MySecureVM" --resource-group "MyVirtualMachineResourceGroup" --volume-type [ALL, DATA, OS]
  ```

- **Disable encryption with a Resource Manager template:** To disable encryption, use the Disable encryption on a running Linux VM template.

  1. Select **Deploy to Azure**.
  2. Select the subscription, resource group, location, VM, legal terms, and agreement.
  3. Select **Purchase** to disable disk encryption on a running Windows VM.

Next steps

- Azure Disk Encryption for Linux overview
- Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release)

Azure Disk Encryption for Windows VMs

Summarize this article for me

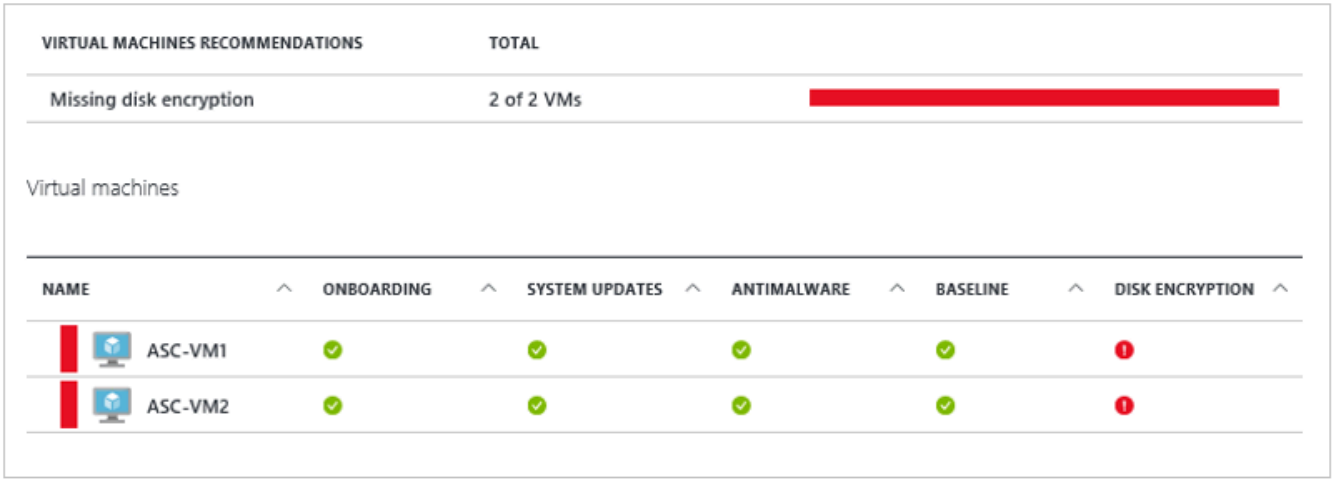
> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Windows VMs ✔ Flexible scale sets.

Azure Disk Encryption helps protect and safeguard your data to meet your organizational security and compliance commitments. It uses the BitLocker feature of Windows to provide volume encryption for the OS and data disks of Azure virtual machines (VMs), and is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets.

Azure Disk Encryption is zone resilient, the same way as Virtual Machines. For details, see Azure Services that support Availability Zones.

If you use Microsoft Defender for Cloud, you're alerted if you have VMs that aren't encrypted. The alerts show as High Severity and the recommendation is to encrypt these VMs.



> **Warning**

- For VMs previously encrypted using Azure Disk Encryption with Microsoft Entra ID, you must continue using that same method. See **Azure Disk Encryption with Microsoft Entra ID (previous release)** for details.
- Implementation of these recommendations may increase your data, network, or compute resource usage, potentially resulting in more license or subscription costs. A valid active Azure subscription is required to create resources in supported regions.
- Never use BitLocker directly to decrypt VMs or disks that were encrypted through Azure Disk Encryption, as this can lead to data loss.

You can learn the fundamentals of Azure Disk Encryption for Windows in just a few minutes with the Create and encrypt a Windows VM with Azure CLI quickstart or the Create and encrypt a Windows VM with Azure PowerShell quickstart.

Supported VMs and operating systems

Supported VMs

Windows VMs are available in a range of sizes. Azure Disk Encryption is supported on Generation 1 and Generation 2 VMs. Azure Disk Encryption is also available for VMs with premium storage.

Azure Disk Encryption is not available on Basic, A-series VMs, v6 series VMs, v7 series VMs, or newer, or on virtual machines with a less than 2 GB of memory. For more exceptions, see Azure Disk Encryption: Restrictions.

Supported operating systems

All versions of Windows that support BitLocker and are configured to meet the requirements of BitLocker. For more information, see BitLocker overview.

> **Note**
>
> Windows Server 2022 and Windows 11 do not support an RSA 2048 bit key. For more information, see **FAQ: What size should I use for my key encryption key?**
>
> Windows Server 2012 R2 Core and Windows Server 2016 Core requires the bdehdcfg component to be installed on the VM for encryption.

Windows Server 2008 R2 requires the .NET Framework 4.5 to be installed for encryption; install it from Windows Update with the optional update Microsoft .NET Framework 4.5.2 for Windows Server 2008 R2 x64-based systems (KB2901983).

Networking requirements

To enable Azure Disk Encryption, the VMs must meet the following network endpoint configuration requirements:

- The Windows VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.
- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see Azure Key Vault behind a firewall.

Group Policy requirements

Azure Disk Encryption uses the BitLocker external key protector for Windows VMs. For domain joined VMs, don't push any group policies that enforce TPM protectors. For information about the group policy for "Allow BitLocker without a compatible TPM," see BitLocker Group Policy Reference.

BitLocker policy on domain joined virtual machines with custom group policy must include the following setting: Configure user storage of BitLocker recovery information -> Allow 256-bit recovery key. Azure Disk Encryption will fail when custom group policy settings for BitLocker are incompatible. On machines that didn't have the correct policy setting, apply the new policy, and force the new policy to update (gpupdate.exe /force). Restarting may be required.

Microsoft BitLocker Administration and Monitoring (MBAM) group policy features aren't compatible with Azure Disk Encryption.

> ⚠ **Warning**
>
> Azure Disk Encryption **does not store recovery keys**. If the **Interactive logon: Machine account lockout threshold** security setting is enabled, machines can only be recovered by providing a recovery key via the serial console. Instructions for ensuring the appropriate recovery policies are enabled can be found in the **Bitlocker recovery guide plan**.

Azure Disk Encryption will fail if domain level group policy blocks the AES-CBC algorithm, which is used by BitLocker.

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see Creating and configuring a key vault for Azure Disk Encryption.

Terminology

The following table defines some of the common terms used in Azure disk encryption documentation:

Expand table

Terminology	Definition
Azure Key Vault	Key Vault is a cryptographic, key management service that's based on Federal Information Processing Standards (FIPS) validated hardware security modules. These standards help to safeguard your cryptographic keys and sensitive secrets. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption.
Azure CLI	The Azure CLI is optimized for managing and administering Azure resources from the command line.
BitLocker	BitLocker is an industry-recognized Windows volume encryption technology that's used to enable disk encryption on Windows VMs.
Key encryption key (KEK)	The asymmetric key (RSA 2048) that you can use to protect or wrap the secret. You can provide a hardware security module (HSM)-protected key or software-protected key. For more information, see the Azure Key Vault documentation and Creating and configuring a key vault for Azure Disk Encryption.
PowerShell cmdlets	For more information, see Azure PowerShell cmdlets.

Next steps

- Quickstart - Create and encrypt a Windows VM with Azure CLI
- Quickstart - Create and encrypt a Windows VM with Azure PowerShell
- Azure Disk Encryption scenarios on Windows VMs
- Migrate from Azure Disk Encryption to server-side encryption
- Azure Disk Encryption prerequisites CLI script
- Azure Disk Encryption prerequisites PowerShell script

- Creating and configuring a key vault for Azure Disk Encryption

---

Last updated on 02/20/2026

Quickstart: Create and encrypt a Windows VM with the Azure CLI

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Windows VMs ✔ Flexible scale sets

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This quickstart shows you how to use the Azure CLI to create and encrypt a Windows Server 2016 virtual machine (VM).

If you don't have an Azure account, create a free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Get started with Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Authenticate to Azure using Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use and manage extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

- This article requires version 2.0.30 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a resource group

Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Create a virtual machine

Create a VM with az vm create. The following example creates a VM named *myVM*. This example uses *azureuser* for an administrative user name and *myPassword12* as the password.

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image win2016datacenter \
    --admin-username azureuser \
    --admin-password myPassword12
```

It takes a few minutes to create the VM and supporting resources. The following example output shows the VM create operation was successful.

Console

```
{
  "fqdns": "",
  "id":
"/subscriptions/<guid>/resourceGroups/myResourceGroup/providers/Microsoft.Compute/
virtualMachines/myVM",
  "location": "eastus",
  "macAddress": "00-0D-3A-23-9A-49",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.174.34.95",
  "resourceGroup": "myResourceGroup"
}
```

Create a Key Vault configured for encryption keys

Azure disk encryption stores its encryption key in an Azure Key Vault. Create a Key Vault with az keyvault create. To enable the Key Vault to store encryption keys, use the --enabled-for-disk-encryption parameter.

> **ⓘ Important**
>
> Each Key Vault must have a unique name. This example creates a Key Vault named *myKV*, but you must name yours something different.

Azure CLI

```
az keyvault create --name "myKV" --resource-group "myResourceGroup" --location
eastus --enabled-for-disk-encryption
```

Encrypt the virtual machine

Encrypt your VM with az vm encryption, providing your unique Key Vault name to the --disk-encryption-keyvault parameter.

Azure CLI

```
az vm encryption enable -g MyResourceGroup --name MyVM --disk-encryption-keyvault
myKV
```

You can verify that encryption is enabled on your VM with az vm show

Azure CLI

```
az vm encryption show --name MyVM -g MyResourceGroup
```

You will see the following in the returned output:

Console

```
"EncryptionOperation": "EnableEncryption"
```

Clean up resources

When no longer needed, you can use the az group delete command to remove the resource group, VM, and Key Vault.

Azure CLI

```
az group delete --name myResourceGroup
```

Next steps

In this quickstart, you created a virtual machine, created a Key Vault that was enabled for encryption keys, and encrypted the VM. Advance to the next article to learn more about Azure Disk Encryption prerequisites for IaaS VMs.

Azure Disk Encryption overview

Quickstart: Create and encrypt a Windows virtual machine in Azure with PowerShell

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Windows VMs

The Azure PowerShell module is used to create and manage Azure resources from the PowerShell command line or in scripts. This quickstart shows you how to use the Azure PowerShell module to create a Windows virtual machine (VM), create a Key Vault for the storage of encryption keys, and encrypt the VM.

If you don't have an Azure subscription, create a free account before you begin.

Create a resource group

Create an Azure resource group with New-AzResourceGroup. A resource group is a logical container into which Azure resources are deployed and managed:

PowerShell

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create a virtual machine

Create an Azure virtual machine with New-AzVM. You must supply credentials to the cmdlet.

PowerShell

```
$cred = Get-Credential
```

```
New-AzVM -Name MyVm -Credential $cred -ResourceGroupName MyResourceGroup -Image
win2016datacenter -Size Standard_D2S_V3
```

It will take a few minutes for your VM to be deployed.

Create a Key Vault configured for encryption keys

Azure disk encryption stores its encryption key in an Azure Key Vault. Create a Key Vault with New-AzKeyvault. To enable the Key Vault to store encryption keys, use the -EnabledForDiskEncryption parameter.

> **Important**
>
> Each Key Vault must have a unique name. The following example creates a Key Vault named *myKV*, but you must name yours something different.

PowerShell

```
New-AzKeyvault -name MyKV -ResourceGroupName myResourceGroup -Location EastUS -
EnabledForDiskEncryption
```

Encrypt the virtual machine

Encrypt your VM with Set-AzVmDiskEncryptionExtension.

Set-AzVmDiskEncryptionExtension requires some values from your Key Vault object. You can obtain these values by passing the unique name of your key vault to Get-AzKeyvault.

PowerShell

```
$KeyVault = Get-AzKeyVault -VaultName MyKV -ResourceGroupName MyResourceGroup

Set-AzVMDiskEncryptionExtension -ResourceGroupName MyResourceGroup -VMName MyVM -
DiskEncryptionKeyVaultUrl $KeyVault.VaultUri -DiskEncryptionKeyVaultId
$KeyVault.ResourceId
```

After a few minutes the process will return the following output:

```
RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
```

You can verify the encryption process by running Get-AzVmDiskEncryptionStatus.

PowerShell

```powershell
Get-AzVmDiskEncryptionStatus -VMName MyVM -ResourceGroupName MyResourceGroup
```

When encryption is enabled, you will see the following fields in the returned output:

```
OsVolumeEncrypted          : Encrypted
DataVolumesEncrypted       : NoDiskFound
OsVolumeEncryptionSettings :
Microsoft.Azure.Management.Compute.Models.DiskEncryptionSettings
ProgressMessage            : Provisioning succeeded
```

Clean up resources

When no longer needed, you can use the Remove-AzResourceGroup cmdlet to remove the resource group, VM, and all related resources:

PowerShell

```powershell
Remove-AzResourceGroup -Name "myResourceGroup"
```

Next steps

In this quickstart, you created a virtual machine, created a Key Vault that was enabled for encryption keys, and encrypted the VM. Advance to the next article to learn more about Azure Disk Encryption prerequisites for IaaS VMs.

Azure Disk Encryption overview

Last updated on 10/10/2025

Quickstart: Create and encrypt a Windows virtual machine with the Azure portal

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Windows VMs

Azure virtual machines (VMs) can be created through the Azure portal. The Azure portal is a browser-based user interface to create VMs and their associated resources. In this quickstart you use the Azure portal to deploy a Windows virtual machine, create a key vault for the storage of encryption keys, and encrypt the VM.

If you don't have an Azure subscription, create a free account before you begin.

Sign in to Azure

Sign in to the Azure portal.

Create a virtual machine

1. Choose **Create a resource** in the upper left corner of the Azure portal.
2. In the New page, under Popular, select **Windows Server 2016 Datacenter**.
3. In the Basics tab, under Project details, make sure the correct subscription is selected.
4. For "Resource Group", select **Create new**. Enter *myResourceGroup* as the name and select **Ok**.
5. For **Virtual machine name**, enter *MyVM*.
6. For **Region**, select *(US) East US*.

7. Verify that the **Size** is *Standard D2s v3*.

8. Under **Administrator account**, select **Password**. Enter a user name and a password.

Home > New > Create a virtual machine

Create a virtual machine

Complete the Basics tab then Review + create to provision a virtual machine with defau
customization.
Looking for classic VMs? Create VM from Azure Marketplace

Project details

Select the subscription to manage deployed resources and costs. Use resource groups l
your resources.

Subscription * ⓘ — Free Trial

Resource group * ⓘ — myResourceGroup
Create new

Instance details

Virtual machine name * ⓘ — myVM

Region * ⓘ — (US) East US

Availability options ⓘ — No infrastructure redundancy required

Image * ⓘ — Windows Server 2016 Datacenter
Browse all public and private images

Size * ⓘ — **Standard DS1 v2**
1 vcpu, 3.5 GiB memory
Change size

Administrator account

Username * ⓘ — azureUser

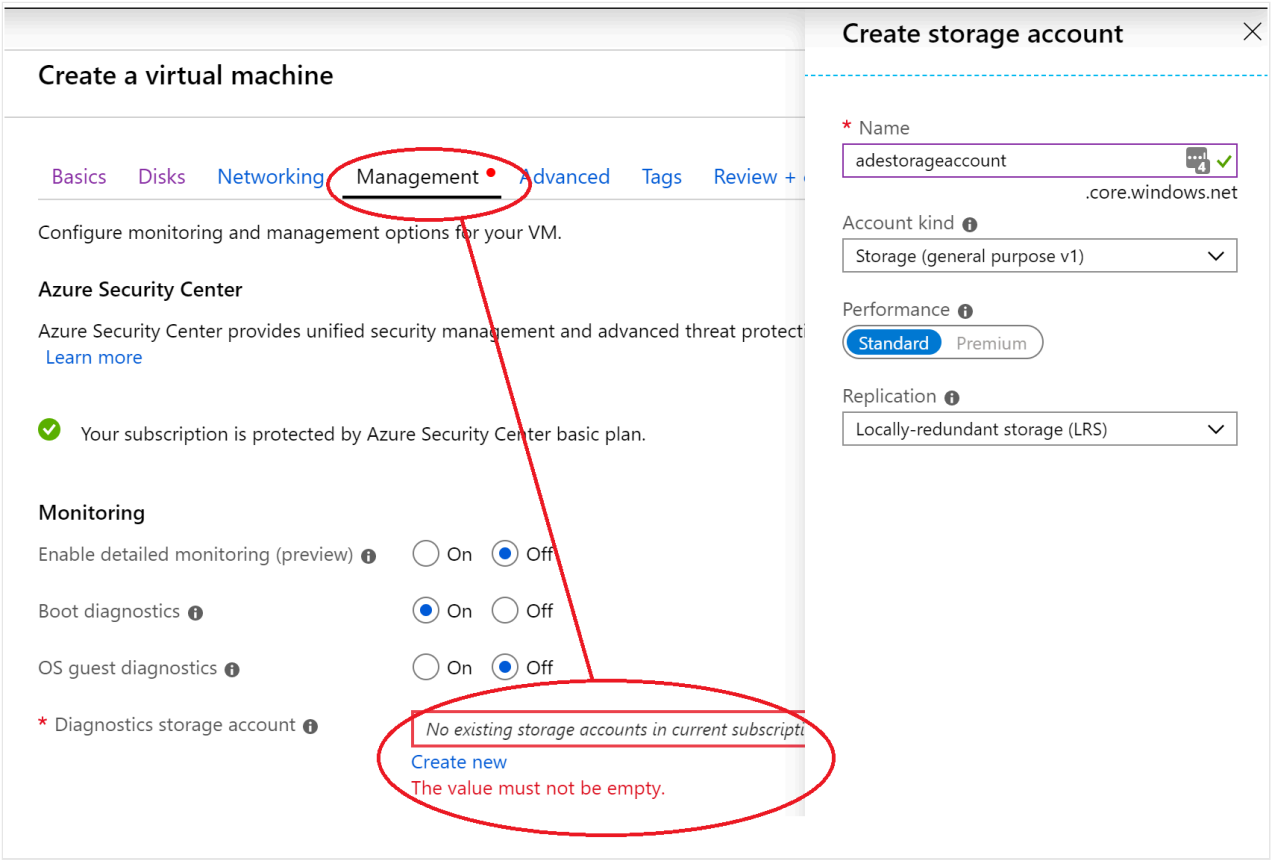
Password * ⓘ — ••••••••••••

Confirm password * ⓘ — ••••••••••••

Review + create | < Previous | Next : Disks >

> **⚠ Warning**
>
> The "Disks" tab features an "Encryption Type" field under **Disk options**. This field is used to specify encryption options for Managed Disks + CMK, not for Azure Disk Encryption.
>
> To avoid confusion, we suggest you skip the *Disks* tab entirely while completing this tutorial.

9. Select the "Management" tab and verify that you have a Diagnostics Storage Account. If you have no storage accounts, select "Create New", give your new account a name, and select "Ok"



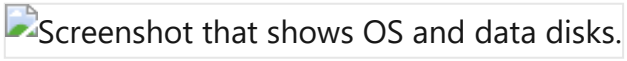
10. Select "Review + Create".

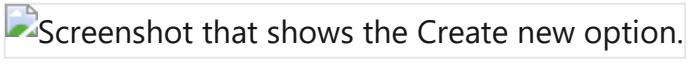
11. On the **Create a virtual machine** page, you can see the details about the VM you're about to create. When you're ready, select **Create**.

It takes a few minutes for your VM to be deployed. When the deployment is finished, move on to the next section.

Encrypt the virtual machine

1. When the VM deployment is complete, select **Go to resource**.
2. On the left-hand sidebar, select **Disks**.
3. On the top bar, select **Additional Settings** .
4. Under **Encryption settings** > **Disks to encrypt**, select **OS and data disks**.

   
   Screenshot that shows OS and data disks.
5. Under **Encryption settings**, choose **Select a key vault and key for encryption**.
6. On the **Select key from Azure Key Vault** screen, select **Create New**.

   
   Screenshot that shows the Create new option.
7. To the left of **Key vault and key**, select **Click to select a key**.
8. On the **Select key from Azure Key Vault**, under the **Key Vault** field, select **Create new**.
9. On the **Create key vault** screen, ensure that the Resource Group is *myResourceGroup*, and give your key vault a name. Every key vault across Azure must have a unique name.
10. On the **Access Policies** tab, check the **Azure Disk Encryption for volume encryption** box.

    
    disks and encryption selection
11. Select **Review + create**.
12. After the key vault has passed validation, select **Create**. You return to the **Select key from Azure Key Vault** screen.
13. Leave the **Key** field blank and choose **Select**.
14. At the top of the encryption screen, select **Save**. A popup warns you that the VM will reboot. Select **Yes**.

Clean up resources

When no longer needed, you can delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the virtual machine, select Delete, then confirm the name of the resource group to delete.

Next steps

In this quickstart, you created a Key Vault that was enabled for encryption keys, created a virtual machine, and enabled the virtual machine for encryption.

Azure Disk Encryption overview

Last updated on 11/20/2025

Azure Disk Encryption scenarios on Windows VMs

Summarize this article for me

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Applies to: ✔ Windows VMs ✔ Flexible scale sets

Azure Disk Encryption for Windows virtual machines (VMs) uses the BitLocker feature of Windows to provide full disk encryption of the OS disk and data disk. Additionally, it provides encryption of the temporary disk when the VolumeType parameter is All.

Azure Disk Encryption is integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets. For an overview of the service, see Azure Disk Encryption for Windows VMs.

Prerequisites

You can only apply disk encryption to virtual machines of supported VM sizes and operating systems. You must also meet the following prerequisites:

- Networking requirements
- Group Policy requirements
- Encryption key storage requirements

Restrictions

If you have previously used Azure Disk Encryption with Microsoft Entra ID to encrypt a VM, you must continue use this option to encrypt your VM. See Azure Disk Encryption with Microsoft Entra ID (previous release) for details.

You should take a snapshot and/or create a backup before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the Set-AzVMDiskEncryptionExtension cmdlet to encrypt managed disks by specifying the -skipVmBackup parameter. For more information about how to back up and restore encrypted VMs, see Back up and restore encrypted Azure VM.

Encrypting or disabling encryption may cause a VM to reboot.

Azure Disk Encryption does not work for the following scenarios, features, and technology:

- Encrypting basic tier VM or VMs created through the classic VM creation method.
- Encrypting v6 series VMs with temporary disks (Ddsv6, Dldsv6, Edsv6, Dadsv6, Daldsv6, Eadsv6, Dpdsv6, Dpldsv6, Epdsv6, or Endsv6). For more information, see the individual pages for each of these VM sizes listed on Sizes for virtual machines in Azure
- Encrypting V7 series and newer VM sizes.
- All requirements and restrictions of BitLocker, such as requiring NTFS. For more information, see BitLocker overview.
- Encrypting VMs configured with software-based RAID systems.
- Encrypting VMs configured with Storage Spaces Direct (S2D), or Windows Server versions before 2016 configured with Windows Storage Spaces.
- Integration with an on-premises key management system.
- Azure Files (shared file system).
- Network File System (NFS).
- Dynamic volumes.
- Windows Server containers, which create dynamic volumes for each container.
- Ephemeral OS disks.
- iSCSI disks.
- Encryption of shared/distributed file systems like (but not limited to) DFS, GFS, DRDB, and CephFS.
- Moving an encrypted VM to another subscription or region.
- Creating an image or snapshot of an encrypted VM and using it to deploy additional VMs.
- 'L' family storage optimized VM size series.
- M-series VMs with Write Accelerator disks.
- Applying ADE to a VM that has disks encrypted with Encryption at Host or server-side encryption with customer-managed keys (SSE + CMK). Applying SSE + CMK to a data disk or adding a data disk with SSE + CMK configured to a VM encrypted with ADE is an unsupported scenario as well.
- Migrating a VM that is encrypted with ADE, or has **ever** been encrypted with ADE, to Encryption at Host or server-side encryption with customer-managed keys.
- Encrypting VMs in failover clusters.

- Encryption of Azure Ultra Disks.
- Encryption of Premium SSD v2 disks.
- Encryption of VMs in subscriptions that have the Secrets should have the specified maximum validity period policy enabled with the DENY effect.
- Encryption of VMs in subscriptions that have the Key Vault secrets should have an expiration date policy enabled with the DENY effect

Install tools and connect to Azure

Azure Disk Encryption can be enabled and managed through the Azure CLI and Azure PowerShell. To do so you must install the tools locally and connect to your Azure subscription.

Azure CLI

The Azure CLI 2.0 is a command-line tool for managing Azure resources. The CLI is designed to flexibly query data, support long-running operations as non-blocking processes, and make scripting easy. You can install it locally by following the steps in Install the Azure CLI.

To Sign in to your Azure account with the Azure CLI, use the az login command.

Azure CLI

```
az login
```

If you would like to select a tenant to sign in under, use:

Azure CLI

```
az login --tenant <tenant>
```

If you have multiple subscriptions and want to specify a specific one, get your subscription list with az account list and specify with az account set.

Azure CLI

```
az account list
az account set --subscription "<subscription name or ID>"
```

For more information, see Get started with Azure CLI 2.0.

Azure PowerShell

The Azure PowerShell az module provides a set of cmdlets that uses the Azure Resource Manager model for managing your Azure resources. You can use it in your browser with Azure Cloud Shell, or you can install it on your local machine using the instructions in Install the Azure PowerShell module.

If you already have it installed locally, make sure you use the latest version of Azure PowerShell SDK version to configure Azure Disk Encryption. Download the latest version of Azure PowerShell release.

To Sign in to your Azure account with Azure PowerShell, use the Connect-AzAccount cmdlet.

PowerShell

```
Connect-AzAccount
```

If you have multiple subscriptions and want to specify one, use the Get-AzSubscription cmdlet to list them, followed by the Set-AzContext cmdlet:

PowerShell

```
Set-AzContext -Subscription <SubscriptionId>
```

Running the Get-AzContext cmdlet will verify that the correct subscription has been selected.

To confirm the Azure Disk Encryption cmdlets are installed, use the Get-command cmdlet:

PowerShell

```
Get-command *diskencryption*
```

For more information, see Getting started with Azure PowerShell.

Enable encryption on an existing or running Windows VM

In this scenario, you can enable encryption by using the Resource Manager template, PowerShell cmdlets, or CLI commands. If you need schema information for the virtual machine extension, see the Azure Disk Encryption for Windows extension article.

Enable encryption on existing or running VMs with Azure PowerShell

Use the Set-AzVMDiskEncryptionExtension cmdlet to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:** The script below initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, and MySecureVault with your values.

  Azure PowerShell

  ```
  $KVRGname = 'MyKeyVaultResourceGroup';
  $VMRGName = 'MyVirtualMachineResourceGroup';
  $vmName = 'MySecureVM';
  $KeyVaultName = 'MySecureVault';
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
  $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
  $KeyVaultResourceId = $KeyVault.ResourceId;

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName
-DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId;
  ```

- **Encrypt a running VM using KEK:**

  Azure PowerShell

  ```
  $KVRGname = 'MyKeyVaultResourceGroup';
  $VMRGName = 'MyVirtualMachineResourceGroup';
  $vmName = 'MyExtraSecureVM';
  $KeyVaultName = 'MySecureVault';
  $keyEncryptionKeyName = 'MyKeyEncryptionKey';
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
  $KVRGname;
  $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
  $KeyVaultResourceId = $KeyVault.ResourceId;
  $keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
  $keyEncryptionKeyName).Key.kid;

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName -
  DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
  $KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
  KeyEncryptionKeyVaultId $KeyVaultResourceId;
  ```

  > **Note**
  >
  > The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-

name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the Get-AzVmDiskEncryptionStatus cmdlet.

Azure PowerShell

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName
'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

To disable the encryption, see Disable encryption and remove the encryption extension.

Enable encryption on existing or running VMs with the Azure CLI

Use the az vm encryption enable command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --
name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type
[All|OS|Data]
```

- **Encrypt a running VM using KEK:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --
name "MySecureVM" --disk-encryption-keyvault  "MySecureVault" --key-
encryption-key "MyKEK_URI" --key-encryption-keyvault
"MySecureVaultContainingTheKEK" --volume-type [All|OS|Data]
```

> ⓘ Note
>
> The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]

> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the az vm encryption show command.

Azure CLI

```
az vm encryption show --name "MySecureVM" --resource-group
"MyVirtualMachineResourceGroup"
```

To disable the encryption, see Disable encryption and remove the encryption extension.

Using the Resource Manager template

You can enable disk encryption on existing or running IaaS Windows VMs in Azure by using the Resource Manager template to encrypt a running Windows VM.

1. On the Azure quickstart template, click **Deploy to Azure**.
2. Select the subscription, resource group, location, settings, legal terms, and agreement. Click **Purchase** to enable encryption on the existing or running IaaS VM.

The following table lists the Resource Manager template parameters for existing or running VMs:

Expand table

Parameter	Description
vmName	Name of the VM to run the encryption operation.
keyVaultName	Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet `(Get-AzKeyVault -ResourceGroupName <MyKeyVaultResourceGroupName>). Vaultname` or the Azure CLI command `az keyvault list --resource-group "MyKeyVaultResourceGroup"`
keyVaultResourceGroup	Name of the resource group that contains the key vault
keyEncryptionKeyURL	The URL of the key encryption key, in the format https://<keyvault-name>.vault.azure.net/key/<key-name>. If you do not wish to use a KEK, leave this field blank.
volumeType	Type of volume that the encryption operation is performed on. Valid values are *OS*, *Data*, and *All*.

Parameter	Description
forceUpdateTag	Pass in a unique value like a GUID every time the operation needs to be force run.
resizeOSDisk	Should the OS partition be resized to occupy full OS VHD before splitting system volume.
location	Location for all resources.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can create a new VM from a pre-encrypted VHD and the associated encryption keys using PowerShell cmdlets or CLI commands.

Use the instructions in Prepare a pre-encrypted Windows VHD. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

Encrypt VMs with pre-encrypted VHDs with Azure PowerShell

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet Set-AzVMOSDisk. The example below gives you some common parameters.

Azure PowerShell

```
$VirtualMachine = New-AzVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri
"os.vhd" Caching ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaaa
" -DiskEncryptionKeyVaultId "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myKVresourcegroup/providers/Microsoft.KeyVault/vaults/m
ytestvault"
New-AzVM -VM $VirtualMachine -ResourceGroupName "MyVirtualMachineResourceGroup"
```

Enable encryption on a newly added data disk

You can add a new disk to a Windows VM using PowerShell, or through the Azure portal.

> **Note**

Newly added data disk encryption must be enabled via Powershell, or CLI only. Currently, the Azure portal does not support enabling encryption on new disks.

Enable encryption on a newly added disk with Azure PowerShell

When using PowerShell to encrypt a new disk for Windows VMs, a new sequence version should be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version. In some cases, a newly added data disk might be encrypted automatically by the Azure Disk Encryption extension. Auto encryption usually occurs when the VM reboots after the new disk comes online. This is typically caused because "All" was specified for the volume type when disk encryption previously ran on the VM. If auto encryption occurs on a newly added data disk, we recommend running the Set-AzVmDiskEncryptionExtension cmdlet again with new sequence version. If your new data disk is auto encrypted and you do not wish to be encrypted, decrypt all drives first then re-encrypt with a new sequence version specifying OS for the volume type.

- **Encrypt a running VM:** The script below initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, and key vault should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, and MySecureVault with your values. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

  Azure PowerShell

  ```
  $KVRGname = 'MyKeyVaultResourceGroup';
  $VMRGName = 'MyVirtualMachineResourceGroup';
  $vmName = 'MySecureVM';
  $KeyVaultName = 'MySecureVault';
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
  $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
  $KeyVaultResourceId = $KeyVault.ResourceId;
  $sequenceVersion = [Guid]::NewGuid();

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName
-DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -VolumeType "All" -
SequenceVersion $sequenceVersion;
  ```

- **Encrypt a running VM using KEK:** This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS

volume, use "OS" for the -VolumeType parameter.

Azure PowerShell

```
$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MyExtraSecureVM';
$KeyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
$KeyVaultResourceId = $KeyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;
$sequenceVersion = [Guid]::NewGuid();

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $KeyVaultResourceId -VolumeType "All" -SequenceVersion
$sequenceVersion;
```

> **Note**
>
> The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

Enable encryption on a newly added disk with Azure CLI

The Azure CLI command will automatically provide a new sequence version for you when you run the command to enable encryption. The example uses "All" for the volume-type parameter. You may need to change the volume-type parameter to OS if you're only encrypting the OS disk. In contrast to PowerShell syntax, the CLI does not require the user to provide a unique sequence version when enabling encryption. The CLI automatically generates and uses its own unique sequence version value.

- **Encrypt a running VM:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --
name "MySecureVM" --disk-encryption-keyvault "MySecureVault" --volume-type
"All"
```

- **Encrypt a running VM using KEK:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --
name "MySecureVM" --disk-encryption-keyvault  "MySecureVault" --key-
encryption-key "MyKEK_URI" --key-encryption-keyvault
"MySecureVaultContainingTheKEK" --volume-type "All"
```

Disable encryption and remove the encryption extension

You can disable the Azure disk encryption extension, and you can remove the Azure disk encryption extension. These are two distinct operations.

To remove ADE, it is recommended that you first disable encryption and then remove the extension. If you remove the encryption extension without disabling it, the disks will still be encrypted. If you disable encryption **after** removing the extension, the extension will be reinstalled (to perform the decrypt operation) and will need to be removed a second time.

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template. Disabling encryption does **not** remove the extension (see Remove the encryption extension).

> ⚠ **Warning**
>
> Disabling data disk encryption when both the OS and data disks have been encrypted can have unexpected results. Disable encryption on all disks instead.
>
> Disabling encryption will start a background process of BitLocker to decrypt the disks. This process should be given sufficient time to complete before attempting to any re-enable encryption.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the Disable-AzVMDiskEncryption cmdlet.

Azure PowerShell

```
Disable-AzVMDiskEncryption -ResourceGroupName "MyVirtualMachineResourceGroup"
-VMName "MySecureVM" -VolumeType "all"
```

- **Disable encryption with the Azure CLI:** To disable encryption, use the az vm encryption disable command.

Azure CLI

```
az vm encryption disable --name "MySecureVM" --resource-group
"MyVirtualMachineResourceGroup" --volume-type "all"
```

- **Disable encryption with a Resource Manager template:**

  1. Click **Deploy to Azure** from the Disable disk encryption on running Windows VM template.
  2. Select the subscription, resource group, location, VM, volume type, legal terms, and agreement.
  3. Click **Purchase** to disable disk encryption on a running Windows VM.

Remove the encryption extension

If you want to decrypt your disks and remove the encryption extension, you must disable encryption **before** removing the extension; see disable encryption.

You can remove the encryption extension using Azure PowerShell or the Azure CLI.

- **Disable disk encryption with Azure PowerShell:** To remove the encryption, use the Remove-AzVMDiskEncryptionExtension cmdlet.

Azure PowerShell

```
Remove-AzVMDiskEncryptionExtension -ResourceGroupName
"MyVirtualMachineResourceGroup" -VMName "MySecureVM"
```

- **Disable encryption with the Azure CLI:** To remove encryption, use the az vm extension delete command.

Azure CLI

```
az vm extension delete -g "MyVirtualMachineResourceGroup" --vm-name
"MySecureVM" -n "AzureDiskEncryption"
```

Next steps

- Azure Disk Encryption overview
- Azure Disk Encryption sample scripts
- Azure Disk Encryption troubleshooting

Last updated on 02/20/2026

Azure Disk Encryption sample scripts

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Windows VMs

This article provides sample scripts for preparing pre-encrypted VHDs and other tasks.

> **Note**
>
> All scripts refer to the latest, non-AAD version of ADE, except where noted.

Sample PowerShell scripts for Azure Disk Encryption

- **List all encrypted VMs in your subscription**

  You can find all ADE-encrypted VMs and the extension version, in all resource groups present in a subscription, using this PowerShell script.

  Alternatively, these cmdlets will show all ADE-encrypted VMs (but not the extension version):

  Azure PowerShell

```
$osVolEncrypted = {(Get-AzVMDiskEncryptionStatus -ResourceGroupName
$_.ResourceGroupName -VMName $_.Name).OsVolumeEncrypted}
$dataVolEncrypted= {(Get-AzVMDiskEncryptionStatus -ResourceGroupName
$_.ResourceGroupName -VMName $_.Name).DataVolumesEncrypted}
Get-AzVm | Format-Table @{Label="MachineName"; Expression={$_.Name}},
@{Label="OsVolumeEncrypted"; Expression=$osVolEncrypted},
@{Label="DataVolumesEncrypted"; Expression=$dataVolEncrypted}
```

- **List all encrypted VMSS instances in your subscription**

You can find all ADE-encrypted Virtual Machine Scale Sets instances and the extension version, in all resource groups present in a subscription, using this PowerShell script.

- **List all disk encryption secrets used for encrypting VMs in a key vault**

Azure PowerShell

```
Get-AzKeyVaultSecret -VaultName $KeyVaultName | where
{$_.Tags.ContainsKey('DiskEncryptionKeyFileName')} | format-table
@{Label="MachineName"; Expression={$_.Tags['MachineName']}},
@{Label="VolumeLetter"; Expression={$_.Tags['VolumeLetter']}},
@{Label="EncryptionKeyURL"; Expression={$_.Id}}
```

Using the Azure Disk Encryption prerequisites PowerShell script

If you're already familiar with the prerequisites for Azure Disk Encryption, you can use the Azure Disk Encryption prerequisites PowerShell script. For an example of using this PowerShell script, see the Encrypt a VM Quickstart. You can remove the comments from a section of the script, starting at line 211, to encrypt all disks for existing VMs in an existing resource group.

The following table shows which parameters can be used in the PowerShell script:

Expand table

Parameter	Description	Mandatory?
$resourceGroupName	Name of the resource group to which the KeyVault belongs. A new resource group with this name will be created if one doesn't exist.	True
$keyVaultName	Name of the KeyVault in which encryption keys are to be placed. A new vault with this name will be created if one doesn't exist.	True
$location	Location of the KeyVault. Make sure the KeyVault and VMs to be encrypted are in the same location. Get a location list with `Get-AzLocation`.	True
$subscriptionId	Identifier of the Azure subscription to be used. You can get your Subscription ID with `Get-AzSubscription`.	True
$aadAppName	Name of the Microsoft Entra application that will be used to write secrets to KeyVault. A new application with this name will	False

Parameter	Description	Mandatory?
	be created if one doesn't exist. If this app already exists, pass aadClientSecret parameter to the script.	
$aadClientSecret	Client secret of the Microsoft Entra application that was created earlier.	False
$keyEncryptionKeyName	Name of optional key encryption key in KeyVault. A new key with this name will be created if one doesn't exist.	False

Resource Manager templates

Encrypt or decrypt VMs without a Microsoft Entra app

- Enable disk encryption on an existing or running Windows VM
- Disable encryption on a running Windows VM

Encrypt or decrypt VMs with a Microsoft Entra app (previous release)

- Enable disk encryption on an existing or running Windows VM
- Disable encryption on a running Windows VM
- Create a new encrypted managed disk from a pre-encrypted VHD/storage blob
  - Creates a new encrypted managed disk provided a pre-encrypted VHD and its corresponding encryption settings

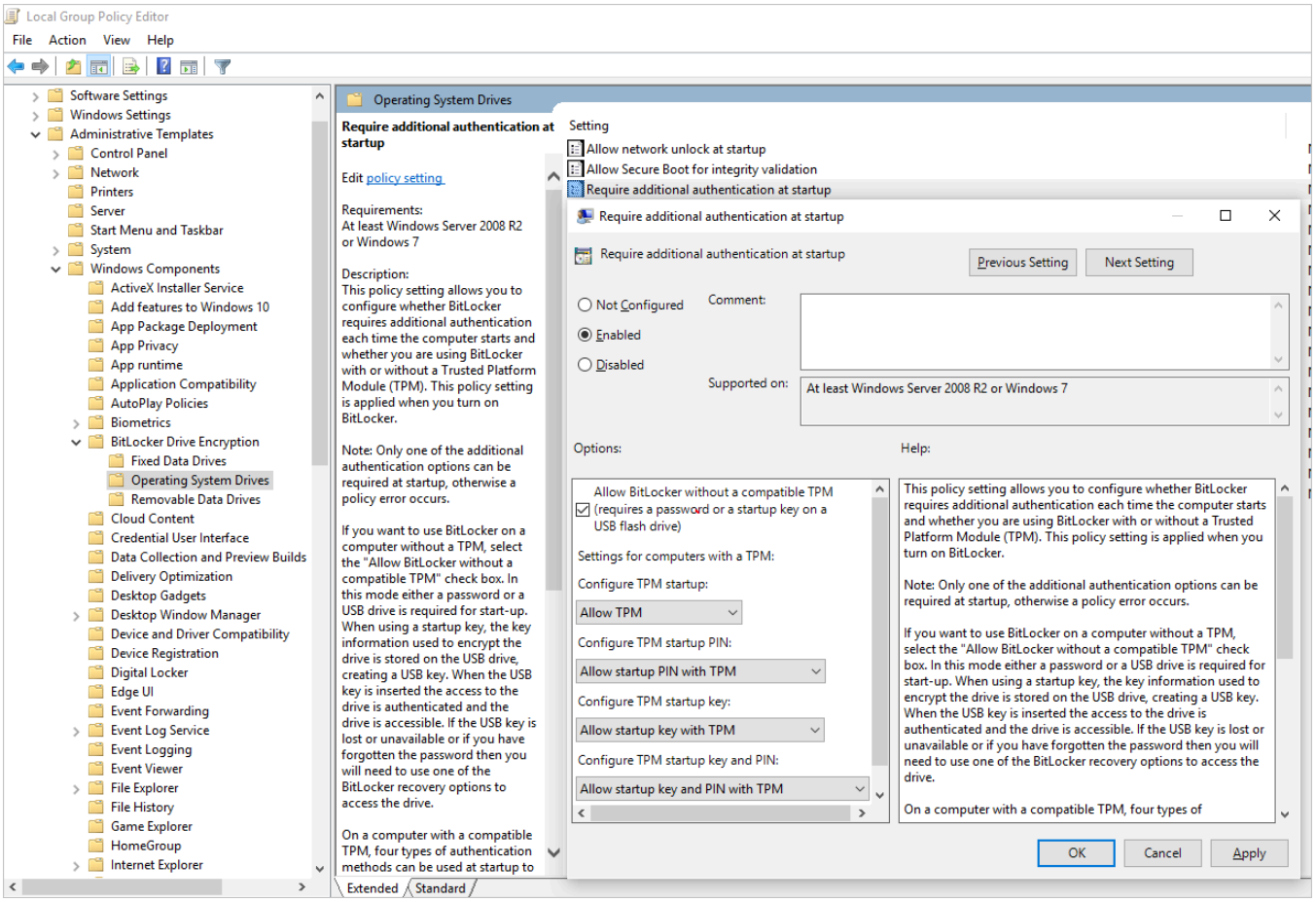
Prepare a pre-encrypted Windows VHD

The sections that follow are necessary to prepare a pre-encrypted Windows VHD for deployment as an encrypted VHD in Azure IaaS. Use the information to prepare and boot a fresh Windows VM (VHD) on Azure Site Recovery or Azure. For more information on how to prepare and upload a VHD, see Upload a generalized VHD and use it to create new VMs in Azure.

Update group policy to allow non-TPM for OS protection

Configure the BitLocker Group Policy setting **BitLocker Drive Encryption**, which you'll find under **Local Computer Policy** > **Computer Configuration** > **Administrative Templates** > **Windows Components**. Change this setting to **Operating System Drives** > **Require additional**

authentication at startup > **Allow BitLocker without a compatible TPM**, as shown in the following figure:



Install BitLocker feature components

For Windows Server 2012 and later, use the following command:

```
Console

dism /online /Enable-Feature /all /FeatureName:BitLocker /quiet /norestart
```

For Windows Server 2008 R2, use the following command:

```
Console

ServerManagerCmd -install BitLockers
```

Prepare the OS volume for BitLocker by using `bdehdcfg`

To compress the OS partition and prepare the machine for BitLocker, execute the bdehdcfg if needed:

Console

```
bdehdcfg -target c: shrink -quiet
```

Protect the OS volume by using BitLocker

Use the manage-bde command to enable encryption on the boot volume using an external key protector. Also place the external key (.bek file) on the external drive or volume. Encryption is enabled on the system/boot volume after the next reboot.

Console

```
manage-bde -on %systemdrive% -sk [ExternalDriveOrVolume]
reboot
```

> **Note**
>
> Prepare the VM with a separate data/resource VHD for getting the external key by using BitLocker.

Upload encrypted VHD to an Azure storage account

After BitLocker encryption is enabled, the local encrypted VHD needs to be uploaded to your storage account.

PowerShell

```
    Add-AzVhd [-Destination] <Uri> [-LocalFilePath] <FileInfo> [[-
NumberOfUploaderThreads] <Int32> ] [[-BaseImageUriToPatch] <Uri> ] [[-OverWrite]]
[ <CommonParameters>]
```

Upload the secret for the pre-encrypted VM to your key vault

The disk encryption secret that you obtained previously must be uploaded as a secret in your key vault. To do so, you must grant the set secret permission and the wrapkey permission to the account that will upload the secrets.

PowerShell

```
# Typically, account Id is the user principal name (in user@domain.com format)
$upn = (Get-AzureRmContext).Account.Id
Set-AzKeyVaultAccessPolicy -VaultName $kvname -UserPrincipalName $acctid -
PermissionsToKeys wrapKey -PermissionsToSecrets set

# In cloud shell, the account ID is a managed service identity, so specify the
username directly
# $upn = "user@domain.com"
# Set-AzKeyVaultAccessPolicy -VaultName $kvname -UserPrincipalName $acctid -
PermissionsToKeys wrapKey -PermissionsToSecrets set

# When running as a service principal, retrieve the service principal ID from the
account ID, and set access policy to that
# $acctid = (Get-AzureRmContext).Account.Id
# $spoid = (Get-AzureRmADServicePrincipal -ServicePrincipalName $acctid).Id
# Set-AzKeyVaultAccessPolicy -VaultName $kvname -ObjectId $spoid -
BypassObjectIdValidation -PermissionsToKeys wrapKey -PermissionsToSecrets set
```

Disk encryption secret not encrypted with a KEK

To set up the secret in your key vault, use Set-AzKeyVaultSecret. The passphrase is encoded as a base64 string and then uploaded to the key vault. In addition, make sure that the following tags are set when you create the secret in the key vault.

PowerShell

```
 # This is the passphrase that was provided for encryption during the distribution
installation
 $passphrase = "contoso-password"

 $tags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP";
"DiskEncryptionKeyFileName" = "LinuxPassPhraseFileName"}
 $secretName = [guid]::NewGuid().ToString()
 $secretValue =
[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($passphrase))
 $secureSecretValue = ConvertTo-SecureString $secretValue -AsPlainText -Force

 $secret = Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $secretName -
SecretValue $secureSecretValue -tags $tags
 $secretUrl = $secret.Id
```

Use the `$secretUrl` in the next step for attaching the OS disk without using KEK.

Disk encryption secret encrypted with a KEK

Before you upload the secret to the key vault, you can optionally encrypt it by using a key encryption key. Use the wrap API to first encrypt the secret using the key encryption key. The output of this wrap operation is a base64 URL encoded string, which you can then upload as a secret by using the Set-AzKeyVaultSecret cmdlet.

PowerShell

```
    # This is the passphrase that was provided for encryption during the
distribution installation
    $passphrase = "contoso-password"

    Add-AzKeyVaultKey -VaultName $KeyVaultName -Name "keyencryptionkey" -
Destination Software
    $KeyEncryptionKey = Get-AzKeyVaultKey -VaultName $KeyVault.OriginalVault.Name
-Name "keyencryptionkey"

    $apiversion = "2015-06-01"

    ###############################
    # Get Auth URI
    ###############################

    $uri = $KeyVault.VaultUri + "/keys"
    $headers = @{}

    $response = try { Invoke-RestMethod -Method GET -Uri $uri -Headers $headers }
catch { $_.Exception.Response }

    $authHeader = $response.Headers["www-authenticate"]
    $authUri = [regex]::match($authHeader, 'authorization="
(.*?)"').Groups[1].Value

    Write-Host "Got Auth URI successfully"

    ###############################
    # Get Auth Token
    ###############################

    $uri = $authUri + "/oauth2/token"
    $body = "grant_type=client_credentials"
    $body += "&client_id=" + $AadClientId
    $body += "&client_secret=" + [Uri]::EscapeDataString($AadClientSecret)
    $body += "&resource=" + [Uri]::EscapeDataString("https://vault.azure.net")
    $headers = @{}

    $response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body
$body

    $access_token = $response.access_token

    Write-Host "Got Auth Token successfully"

    ###############################
```

```
    # Get KEK info
    ##############################

    $uri = $KeyEncryptionKey.Id + "?api-version=" + $apiversion
    $headers = @{"Authorization" = "Bearer " + $access_token}

    $response = Invoke-RestMethod -Method GET -Uri $uri -Headers $headers

    $keyid = $response.key.kid

    Write-Host "Got KEK info successfully"

    ##############################
    # Encrypt passphrase using KEK
    ##############################

    $passphraseB64 =
[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($Passphrase))
    $uri = $keyid + "/encrypt?api-version=" + $apiversion
    $headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" =
"application/json"}
    $bodyObj = @{"alg" = "RSA-OAEP"; "value" = $passphraseB64}
    $body = $bodyObj | ConvertTo-Json

    $response = Invoke-RestMethod -Method POST -Uri $uri -Headers $headers -Body
$body

    $wrappedSecret = $response.value

    Write-Host "Encrypted passphrase successfully"

    ##############################
    # Store secret
    ##############################

    $secretName = [guid]::NewGuid().ToString()
    $uri = $KeyVault.VaultUri + "/secrets/" + $secretName + "?api-version=" +
$apiversion
    $secretAttributes = @{"enabled" = $true}
    $secretTags = @{"DiskEncryptionKeyEncryptionAlgorithm" = "RSA-OAEP";
"DiskEncryptionKeyFileName" = "LinuxPassPhraseFileName"}
    $headers = @{"Authorization" = "Bearer " + $access_token; "Content-Type" =
"application/json"}
    $bodyObj = @{"value" = $wrappedSecret; "attributes" = $secretAttributes;
"tags" = $secretTags}
    $body = $bodyObj | ConvertTo-Json

    $response = Invoke-RestMethod -Method PUT -Uri $uri -Headers $headers -Body
$body

    Write-Host "Stored secret successfully"

    $secretUrl = $response.id
```

Use `$KeyEncryptionKey` and `$secretUrl` in the next step for attaching the OS disk using KEK.

Specify a secret URL when you attach an OS disk

Without using a KEK

While you're attaching the OS disk, you need to pass `$secretUrl`. The URL was generated in the "Disk-encryption secret not encrypted with a KEK" section.

PowerShell

```
Set-AzVMOSDisk `
        -VM $VirtualMachine `
        -Name $OSDiskName `
        -SourceImageUri $VhdUri `
        -VhdUri $OSDiskUri `
        -Windows `
        -CreateOption FromImage `
        -DiskEncryptionKeyVaultId $KeyVault.ResourceId `
        -DiskEncryptionKeyUrl $SecretUrl
```

Using a KEK

When you attach the OS disk, pass `$KeyEncryptionKey` and `$secretUrl`. The URL was generated in the "Disk encryption secret encrypted with a KEK" section.

PowerShell

```
Set-AzVMOSDisk `
        -VM $VirtualMachine `
        -Name $OSDiskName `
        -SourceImageUri $CopiedTemplateBlobUri `
        -VhdUri $OSDiskUri `
        -Windows `
        -CreateOption FromImage `
        -DiskEncryptionKeyVaultId $KeyVault.ResourceId `
        -DiskEncryptionKeyUrl $SecretUrl `
        -KeyEncryptionKeyVaultId $KeyVault.ResourceId `
        -KeyEncryptionKeyURL $KeyEncryptionKey.Id
```

Azure Disk Encryption troubleshooting guide

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Windows VMs ✔ Flexible scale sets

This guide is for IT professionals, information security analysts, and cloud administrators who use Azure Disk Encryption. This article helps troubleshoot disk encryption problems.

Before taking these steps, ensure the VMs you want to encrypt are among the supported VM sizes and operating systems and that you meet all the prerequisites:

- Networking requirements
- Group policy requirements
- Encryption key storage requirements

Troubleshooting 'failed to send DiskEncryptionData'

When encrypting a VM fails with the error message "Failed to send DiskEncryptionData...", it's usually caused by one of the following situations:

- The Key Vault exists in a different region or subscription than the virtual machine
- Advanced access policies in the Key Vault aren't set to allow Azure Disk Encryption
- The Key Encryption Key is disabled or deleted in the Key Vault
- A typo exists in the Resource ID or URL for the Key Vault or Key Encryption Key (KEK)
- Special characters are used in the names of the VM, data disks, or keys. For example, "_VMName" or "élite".
- The encryption scenario isn't supported
- Network issues prevent the VM or host from accessing the required resources

Suggestions for resolving the issue

- Make sure the Key Vault exists in the same region and subscription as the Virtual Machine
- Ensure you set key vault advanced access policies correctly
- If you are using KEK, ensure the key exists and is enabled in Key Vault
- Check that the VM name, data disks, and keys follow key vault resource naming restrictions.
- Check for typos in the Key Vault name or KEK name in PowerShell or CLI commands

> **Note**
>
> The syntax for the value of the `disk-encryption-keyvault` parameter is the full identifier string: `/subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]`
> The syntax for the value of the `key-encryption-key` parameter is the full URI to the KEK, such as: `https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]`

- Ensure you are not violating any restrictions
- Ensure you are meeting network requirements and try again

Troubleshooting Azure Disk Encryption behind a firewall

When connectivity is restricted by a firewall, proxy requirement, or network security group (NSG) settings, the extension may be unable to perform the needed tasks. This disruption can result in status messages such as "Extension status not available on the VM." In typical scenarios, the encryption doesn't finish. The sections that follow have some common firewall problems that you might investigate.

Network security groups

Any network security group settings that are applied must still allow the endpoint to meet the documented network configuration prerequisites for disk encryption.

Azure Key Vault behind a firewall

When encryption is being enabled with Microsoft Entra credentials, the target VM must allow connectivity to both Microsoft Entra endpoints and Key Vault endpoints. Current Microsoft Entra authentication endpoints are maintained in sections 56 and 59 of the Microsoft 365 URLs

and IP address ranges documentation. Key Vault instructions are provided in the documentation on how to Access Azure Key Vault behind a firewall.

Azure Instance Metadata Service

The VM must be able to access the Azure Instance Metadata service endpoint (`169.254.169.254`) and the virtual public IP address (`168.63.129.16`) used for communication with Azure platform resources. Proxy configurations that alter local HTTP traffic to these addresses, like adding an X-Forwarded-For header, aren't supported.

Troubleshooting Windows Server 2016 Server Core

On Windows Server 2016 Server Core, the `bdehdcfg` component isn't available by default. Azure Disk Encryption requires this component. It's used to split the system volume from the OS volume, which is done only once for the lifetime of the VM. These binaries aren't required during later encryption operations.

To work around this issue, copy the following four files from a Windows Server 2016 Data Center VM to the same location on Server Core:

```
\windows\system32\bdehdcfg.exe
\windows\system32\bdehdcfglib.dll
\windows\system32\en-US\bdehdcfglib.dll.mui
\windows\system32\en-US\bdehdcfg.exe.mui
```

1. Run the following command:

   ```
   bdehdcfg.exe -target default
   ```

2. This command creates a 550 MB system partition. Restart the system.

3. Use DiskPart to check the volumes. Then proceed.

For example:

```
DISKPART> list vol

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
```

```
----------  ---  -----------  -----  ----------  -------  ---------  --------
Volume 0     C                NTFS   Partition    126 GB  Healthy    Boot
Volume 1                      NTFS   Partition    550 MB  Healthy    System
Volume 2     D   Temporary S  NTFS   Partition     13 GB  Healthy    Pagefile
```

Troubleshooting encryption status

The portal might display a disk as encrypted even after it is unencrypted within the VM. This situation can occur when low-level commands are used to directly unencrypt the disk from within the VM instead of using the higher-level Azure Disk Encryption management commands. The higher-level commands not only unencrypt the disk from within the VM but also update important platform-level encryption settings and extension settings associated with the VM. If these are not kept in alignment, the platform is unable to report encryption status or provision the VM properly.

To disable Azure Disk Encryption with PowerShell, use Disable-AzVMDiskEncryption followed by Remove-AzVMDiskEncryptionExtension. Running Remove-AzVMDiskEncryptionExtension before the encryption is disabled fails.

To disable Azure Disk Encryption with the CLI, use az vm encryption disable.

Next steps

In this document, you learned more about some common problems in Azure Disk Encryption and how to troubleshoot those problems. For more information about this service and its capabilities, see the following articles:

- Apply disk encryption in Microsoft Defender for Cloud
- Azure data encryption at rest

Azure Disk Encryption for Windows virtual machines FAQ

This article provides answers to frequently asked questions (FAQ) about Azure Disk Encryption for Windows VMs. For more information about this service, see Azure Disk Encryption overview.

What is Azure Disk Encryption for Windows VMs?

Azure Disk Encryption for Windows VMs uses the BitLocker feature of Windows to provide full disk encryption of the OS disk and data disks. Additionally, it provides encryption of the temporary disk when the VolumeType parameter is All. The content flows encrypted from the VM to the Storage backend. Thereby, providing end-to-end encryption with a customer-managed key.

See Supported VMs and operating systems.

Where is Azure Disk Encryption in general availability (GA)?

Azure Disk Encryption is in general availability in all Azure public regions.

What user experiences are available with Azure Disk Encryption?

Azure Disk Encryption GA supports Azure Resource Manager templates, Azure PowerShell, and Azure CLI. The different user experiences give you flexibility. You have three different options for enabling disk encryption for your VMs. For more information on the user experience and step-by-step guidance available in Azure Disk Encryption, see Azure Disk Encryption scenarios for Windows.

How much does Azure Disk Encryption cost?

There's no charge for encrypting VM disks with Azure Disk Encryption, but there are charges associated with the use of Azure Key Vault. For more information on Azure Key Vault costs, see the Key Vault pricing page.

How can I start using Azure Disk Encryption?

To get started, read the Azure Disk Encryption overview.

What VM sizes and operating systems support Azure Disk Encryption?

The Azure Disk Encryption overview article lists the VM sizes and VM operating systems that support Azure Disk Encryption.

Can I encrypt both boot and data volumes with Azure Disk Encryption?

You can encrypt both boot and data volumes, but you can't encrypt the data without first encrypting the OS volume.

Can I encrypt an unmounted volume with Azure Disk Encryption?

No, Azure Disk Encryption only encrypts mounted volumes.

What is Storage server-side encryption?

Storage server-side encryption encrypts Azure managed disks in Azure Storage. Managed disks are encrypted by default with Server-side encryption with a platform-managed key (as of June 10, 2017). You can manage encryption of managed disks with your own keys by specifying a customer-managed key. For more information, see Server-side encryption of Azure managed disks.

How is Azure Disk Encryption different from Storage server-side encryption with customer-

managed key and when should I use each solution?

Azure Disk Encryption provides end-to-end encryption for the OS disk, data disks, and the temporary disk with a customer-managed key.

- If your requirements include encrypting all of the above and end-to-end encryption, use Azure Disk Encryption.
- If your requirements include encrypting only data at rest with customer-managed key, then use Server-side encryption with customer-managed keys. You can't encrypt a disk with both Azure Disk Encryption and Storage server-side encryption with customer managed keys.
- If you're using a scenario called out in Restrictions, consider Server-side encryption with customer-managed keys.
- If your organization's policy allows you to encrypt content at rest with an Azure-managed key, then no action is needed - the content is encrypted by default. For managed disks, the content inside storage is encrypted by default with Server-side encryption with platform-managed key. The key is managed by the Azure Storage service.

How do I rotate secrets or encryption keys?

To rotate secrets, just call the same command you used originally to enable disk encryption, specifying a different Key Vault. To rotate the key encryption key, call the same command you used originally to enable disk encryption, specifying the new key encryption.

> ⚠ **Warning**
>
> - If you've previously used **Azure Disk Encryption with Microsoft Entra app** by specifying Microsoft Entra credentials to encrypt this VM, you must continue to use this option. Using Azure Disk Encryption without Microsoft Entra ID on a VM that's been encrypted using Azure Disk Encryption with Microsoft Entra ID isn't yet a supported scenario.

How do I add or remove a key encryption key (KEK) if I didn't originally use one?

To add a key encryption key, call the enable command again passing the key encryption key parameter. To remove a key encryption key, call the enable command again without the key encryption key parameter.

What size should I use for my key encryption key (KEK)?

Windows Server 2022 and Windows 11 include a newer version of BitLocker and currently doesn't work with RSA 2048 bit Key Encryption Keys. Until resolved, use an RSA 3072 or RSA 4096-bit keys, as described in Supported operating systems.

For earlier version of Windows, you may instead use RSA 2048 Key Encryption Keys.

Does Azure Disk Encryption allow you to bring your own key (BYOK)?

Yes, you can supply your own key encryption keys. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption keys support scenarios, see Creating and configuring a key vault for Azure Disk Encryption.

Can I use an Azure-created key encryption key?

Yes, you can use Azure Key Vault to generate a key encryption key for Azure disk encryption use. These keys are safeguarded in Azure Key Vault, which is the key store for Azure Disk Encryption. For more information on the key encryption key, see Creating and configuring a key vault for Azure Disk Encryption.

Can I use an on-premises key management service or HSM to safeguard the encryption keys?

You can't use the on-premises key management service or HSM to safeguard the encryption keys with Azure Disk Encryption. You can only use the Azure Key Vault service to safeguard the

encryption keys. For more information on the key encryption key support scenarios, see Creating and configuring a key vault for Azure Disk Encryption.

What are the prerequisites to configure Azure Disk Encryption?

There are prerequisites for Azure Disk Encryption. See the Creating and configuring a key vault for Azure Disk Encryption article to create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see Creating and configuring a key vault for Azure Disk Encryption.

What are the prerequisites to configure Azure Disk Encryption with a Microsoft Entra app (previous release)?

There are prerequisites for Azure Disk Encryption. See the Azure Disk Encryption with Microsoft Entra ID content to create an Microsoft Entra application, create a new key vault, or set up an existing key vault for disk encryption access to enable encryption, and safeguard secrets and keys. For more information on the key encryption key support scenarios, see Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID.

Is Azure Disk Encryption using a Microsoft Entra app (previous release) still supported?

Yes. Disk encryption using a Microsoft Entra app is still supported. However, when encrypting new VMs it's recommended that you use the new method rather than encrypting with a Microsoft Entra app.

Can I migrate VMs that were encrypted with a Microsoft Entra app to encryption without a Microsoft Entra app?

Currently, there isn't a direct migration path for machines that were encrypted with a Microsoft Entra app to encryption without a Microsoft Entra app. Additionally, there isn't a direct path from encryption without a Microsoft Entra app to encryption with an AD app.

What version of Azure PowerShell does Azure Disk Encryption support?

Use the latest version of the Azure PowerShell SDK to configure Azure Disk Encryption. Download the latest version of Azure PowerShell. Azure Disk Encryption is *not* supported by Azure SDK version 1.1.0.

What is the disk "Bek Volume" or "/mnt/azure_bek_disk"?

The "Bek volume" is a local data volume that securely stores the encryption keys for Encrypted Azure VMs.

> **Note**
>
> Don't delete or edit any contents in this disk. Don't unmount the disk since the encryption key presence is needed for any encryption operations on the IaaS VM.

What encryption method does Azure Disk Encryption use?

Azure Disk Encryption selects the encryption method in BitLocker based on the version of Windows as follows:

Expand table

Windows Versions	Version	Encryption Method
Windows Server 2012, Windows 10, or greater	>=1511	XTS-AES 256 bit
Windows Server 2012, Windows 8, 8.1, 10	< 1511	AES 256 bit *

Windows Versions	Version	Encryption Method
Windows Server 2008R2		AES 256 bit with Diffuser

* AES 256 bit with Diffuser isn't supported in Windows 2012 and later.

To determine Windows OS version, run the 'winver' tool in your virtual machine.

Can I back up and restore an encrypted VM?

Azure Backup provides a mechanism to back up and restore encrypted VMs within the same subscription and region. For instructions, see Back up and restore encrypted virtual machines with Azure Backup. Restoring an encrypted VM to a different region isn't currently supported.

Where can I go to ask questions or provide feedback?

You can ask questions or provide feedback on the Microsoft Q&A question page for Azure Disk Encryption.

Next steps

In this document, you learned more about the most frequent questions related to Azure Disk Encryption. For more information about this service, see the following articles:

- Azure Disk Encryption Overview
- Apply disk encryption in Azure Security Center
- Azure data encryption at rest

Azure Disk Encryption with Azure AD (previous release)

09/24/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Windows VMs

The new release of Azure Disk Encryption eliminates the requirement for providing a Microsoft Entra application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Microsoft Entra credentials during the enable encryption step. All new VMs must be encrypted without the Microsoft Entra application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see Azure Disk Encryption for Windows VMs. VMs that were already encrypted with Microsoft Entra application parameters are still supported and should continue to be maintained with the Microsoft Entra syntax.

This article supplements Azure Disk Encryption for Windows VMs with additional requirements and prerequisites for Azure Disk Encryption with Microsoft Entra ID (previous release). The Supported VMs and operating systems section remains the same.

Networking and Group Policy

To enable the Azure Disk Encryption feature using the older Microsoft Entra parameter syntax, the IaaS VMs must meet the following network endpoint configuration requirements:

- To get a token to connect to your key vault, the IaaS VM must be able to connect to a Microsoft Entra endpoint, [login.microsoftonline.com].
- To write the encryption keys to your key vault, the IaaS VM must be able to connect to the key vault endpoint.
- The IaaS VM must be able to connect to an Azure storage endpoint that hosts the Azure extension repository and an Azure storage account that hosts the VHD files.

- If your security policy limits access from Azure VMs to the Internet, you can resolve the preceding URI and configure a specific rule to allow outbound connectivity to the IPs. For more information, see Azure Key Vault behind a firewall.
- The VM to be encrypted must be configured to use TLS 1.2 as the default protocol. If TLS 1.0 has been explicitly disabled and the .NET version hasn't been updated to 4.6 or higher, the following registry change will enable ADE to select the more recent TLS version:

```
Console

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions"=dword:00000001
"SchUseStrongCrypto"=dword:00000001`
```

Group Policy:

- The Azure Disk Encryption solution uses the BitLocker external key protector for Windows IaaS VMs. For domain joined VMs, don't push any group policies that enforce TPM protectors. For information about the group policy for "Allow BitLocker without a compatible TPM," see BitLocker Group Policy Reference.

- BitLocker policy on domain joined virtual machines with custom group policy must include the following setting: Configure user storage of BitLocker recovery information -> Allow 256-bit recovery key. Azure Disk Encryption will fail when custom group policy settings for BitLocker are incompatible. On machines that didn't have the correct policy setting, apply the new policy, force the new policy to update (gpupdate.exe /force), and then restarting may be required.

Encryption key storage requirements

Azure Disk Encryption requires an Azure Key Vault to control and manage disk encryption keys and secrets. Your key vault and VMs must reside in the same Azure region and subscription.

For details, see Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release).

Next steps

- Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release)

- Enable Azure Disk Encryption with Microsoft Entra ID on Windows VMs (previous release)
- Azure Disk Encryption prerequisites CLI script
- Azure Disk Encryption prerequisites PowerShell script

Creating and configuring a key vault for Azure Disk Encryption with Microsoft Entra ID (previous release)

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Windows VMs

The new release of Azure Disk Encryption eliminates the requirement for providing a Microsoft Entra application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Microsoft Entra credentials during the enable encryption step. All new VMs must be encrypted without the Microsoft Entra application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see Azure Disk Encryption. VMs that were already encrypted with Microsoft Entra application parameters are still supported and should continue to be maintained with the Microsoft Entra syntax.

Azure Disk Encryption uses Azure Key Vault to control and manage disk encryption keys and secrets. For more information about key vaults, see Get started with Azure Key Vault and Secure your key vault.

Creating and configuring a key vault for use with Azure Disk Encryption with Microsoft Entra ID (previous release) involves three steps:

1. Create a key vault.
2. Set up a Microsoft Entra application and service principal.
3. Set the key vault access policy for the Microsoft Entra app.
4. Set key vault advanced access policies.

You may also, if you wish, generate or import a key encryption key (KEK).

See the main Creating and configuring a key vault for Azure Disk Encryption article for steps on how to Install tools and connect to Azure.

> **Note**
>
> The steps in this article are automated in the Azure Disk Encryption prerequisites CLI script and Azure Disk Encryption prerequisites PowerShell script.

Create a key vault

Azure Disk Encryption is integrated with Azure Key Vault to help you control and manage the disk-encryption keys and secrets in your key vault subscription. You can create a key vault or use an existing one for Azure Disk Encryption. For more information about key vaults, see Get started with Azure Key Vault and Secure your key vault. You can use a Resource Manager template, Azure PowerShell, or the Azure CLI to create a key vault.

> **Warning**
>
> In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

Create a key vault with PowerShell

You can create a key vault with Azure PowerShell using the New-AzKeyVault cmdlet. For another cmdlets for Key Vault, see Az.KeyVault.

1. Create a new resource group, if needed, with New-AzResourceGroup. To list data center locations, use Get-AzLocation.

   Azure PowerShell

   ```
   # Get-AzLocation
   New-AzResourceGroup –Name 'MyKeyVaultResourceGroup' –Location 'East US'
   ```

2. Create a new key vault using New-AzKeyVault

   Azure PowerShell

   ```
   New-AzKeyVault -VaultName 'MySecureVault' -ResourceGroupName
   'MyKeyVaultResourceGroup' -Location 'East US'
   ```

3. Note the **Vault Name**, **Resource Group Name**, **Resource ID**, **Vault URI**, and the **Object ID** that are returned for later use when you encrypt the disks.

Create a key vault with Azure CLI

You can manage your key vault with Azure CLI using the az keyvault commands. To create a key vault, use az keyvault create.

1. Create a new resource group, if needed, with az group create. To list locations, use az account list-locations

   Azure CLI

   ```
   # To list locations: az account list-locations --output table
   az group create -n "MyKeyVaultResourceGroup" -l "East US"
   ```

2. Create a new key vault using az keyvault create.

   Azure CLI

   ```
   az keyvault create --name "MySecureVault" --resource-group
   "MyKeyVaultResourceGroup" --location "East US"
   ```

3. Note the **Vault Name** (name), **Resource Group Name**, **Resource ID** (ID), **Vault URI**, and the **Object ID** that are returned for use later.

Create a key vault with a Resource Manager template

You can create a key vault by using the Resource Manager template.

1. On the Azure Quickstart Template, select **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, Key Vault name, Object ID, legal terms, and agreement, and then select **Purchase**.

Set up a Microsoft Entra app and service principal

When you need encryption to be enabled on a running VM in Azure, Azure Disk Encryption generates and writes the encryption keys to your key vault. Managing encryption keys in your key vault requires Microsoft Entra authentication. Create a Microsoft Entra application for this purpose. For authentication purposes, you can use either client secret-based authentication or client certificate-based Microsoft Entra authentication.

Set up a Microsoft Entra app and service principal with Azure PowerShell

To execute the following commands, get and use the Azure PowerShell module.

1. Use the New-AzADApplication PowerShell cmdlet to create a Microsoft Entra application. MyApplicationHomePage and the MyApplicationUri can be any values you wish.

   Azure PowerShell

   ```
   $aadClientSecret = "My AAD client secret"
   $aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -
   AsPlainText -Force
   $azureAdApplication = New-AzADApplication -DisplayName "My Application
   Display Name" -HomePage "https://MyApplicationHomePage" -IdentifierUris
   "https://MyApplicationUri" -Password $aadClientSecretSec
   $servicePrincipal = New-AzADServicePrincipal -ApplicationId
   $azureAdApplication.ApplicationId -Role Contributor
   ```

2. The $azureAdApplication.ApplicationId is the Microsoft Entra ClientID and the $aadClientSecret is the client secret that you'll use later to enable Azure Disk Encryption. Safeguard the Microsoft Entra client secret appropriately. Running `$azureAdApplication.ApplicationId` will show you the ApplicationID.

Set up a Microsoft Entra app and service principal with Azure CLI

You can manage your service principals with Azure CLI using the az ad sp commands. For more information, see Create an Azure service principal.

1. Create a new service principal.

   Azure CLI

   ```
   az ad sp create-for-rbac --name "ServicePrincipalName" --password "My-AAD-
   client-secret" --role Contributor --scopes /subscriptions/<subscription_id>
   ```

2. The appId returned is the Microsoft Entra ClientID used in other commands. It's also the SPN you'll use for az keyvault set-policy. The password is the client secret that you should use later to enable Azure Disk Encryption. Safeguard the Microsoft Entra client secret appropriately.

Set up a Microsoft Entra app and service principal through the Azure portal

Use the steps from the Use portal to create a Microsoft Entra application and service principal that can access resources article to create a Microsoft Entra application. Each of these steps will take you directly to the article section to complete.

1. Verify required permissions
2. Create a Microsoft Entra application

    - You can use any name and sign-on URL you would like when creating the application.

3. Get the application ID and the authentication key.

    - The authentication key is the client secret and is used as the AadClientSecret for Set-AzVMDiskEncryptionExtension.
        - The authentication key is used by the application as a credential to sign in to Microsoft Entra ID. In the Azure portal, this secret is called keys, but has no relation to key vaults. Secure this secret appropriately.
    - The application ID will be used later as the AadClientId for Set-AzVMDiskEncryptionExtension and as the ServicePrincipalName for Set-AzKeyVaultAccessPolicy.

Set the key vault access policy for the Microsoft Entra app

To write encryption secrets to a specified Key Vault, Azure Disk Encryption needs the Client ID and the Client Secret of the Microsoft Entra application that has permissions to write secrets to the Key Vault.

> **Note**
>
> Azure Disk Encryption requires you to configure the following access policies to your Microsoft Entra client application: *WrapKey* and *Set* permissions.

Set the key vault access policy for the Microsoft Entra app with Azure PowerShell

Your Microsoft Entra application needs rights to access the keys or secrets in the vault. Use the Set-AzKeyVaultAccessPolicy cmdlet to grant permissions to the application, using the client ID (which was generated when the application was registered) as the *–ServicePrincipalName* parameter value. To learn more, see the blog post Azure Key Vault - Step by Step.

1. Set the key vault access policy for the AD application with PowerShell.

Azure PowerShell

```
$keyVaultName = 'MySecureVault'
$aadClientID = 'MyAadAppClientID'
$KVRGname = 'MyKeyVaultResourceGroup'
Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName $KVRGname
```

Set the key vault access policy for the Microsoft Entra app with Azure CLI

Use az keyvault set-policy to set the access policy. For more information, see Manage Key Vault using CLI 2.0.

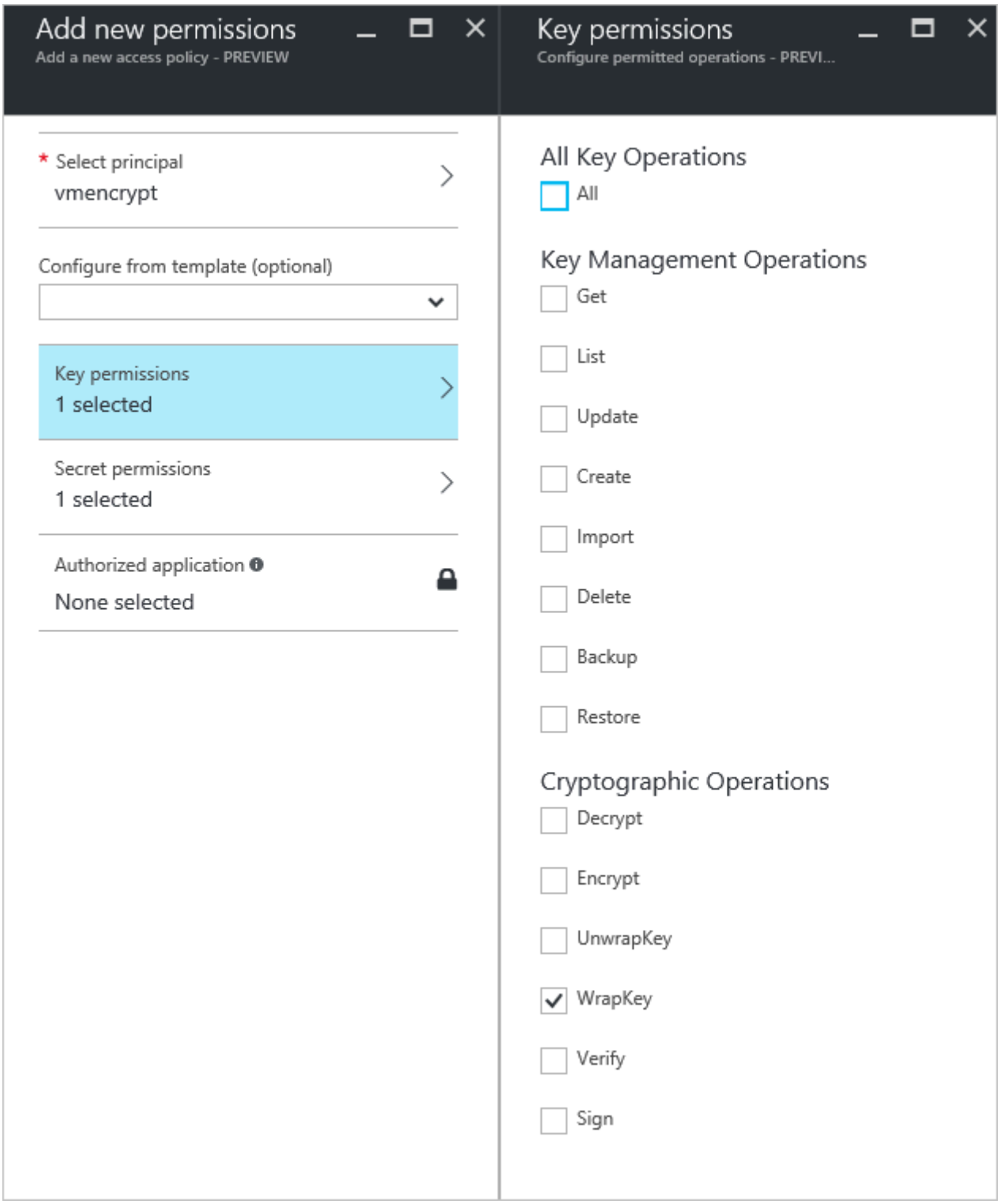
Give the service principal you created via the Azure CLI access to get secrets and wrap keys with the following command:

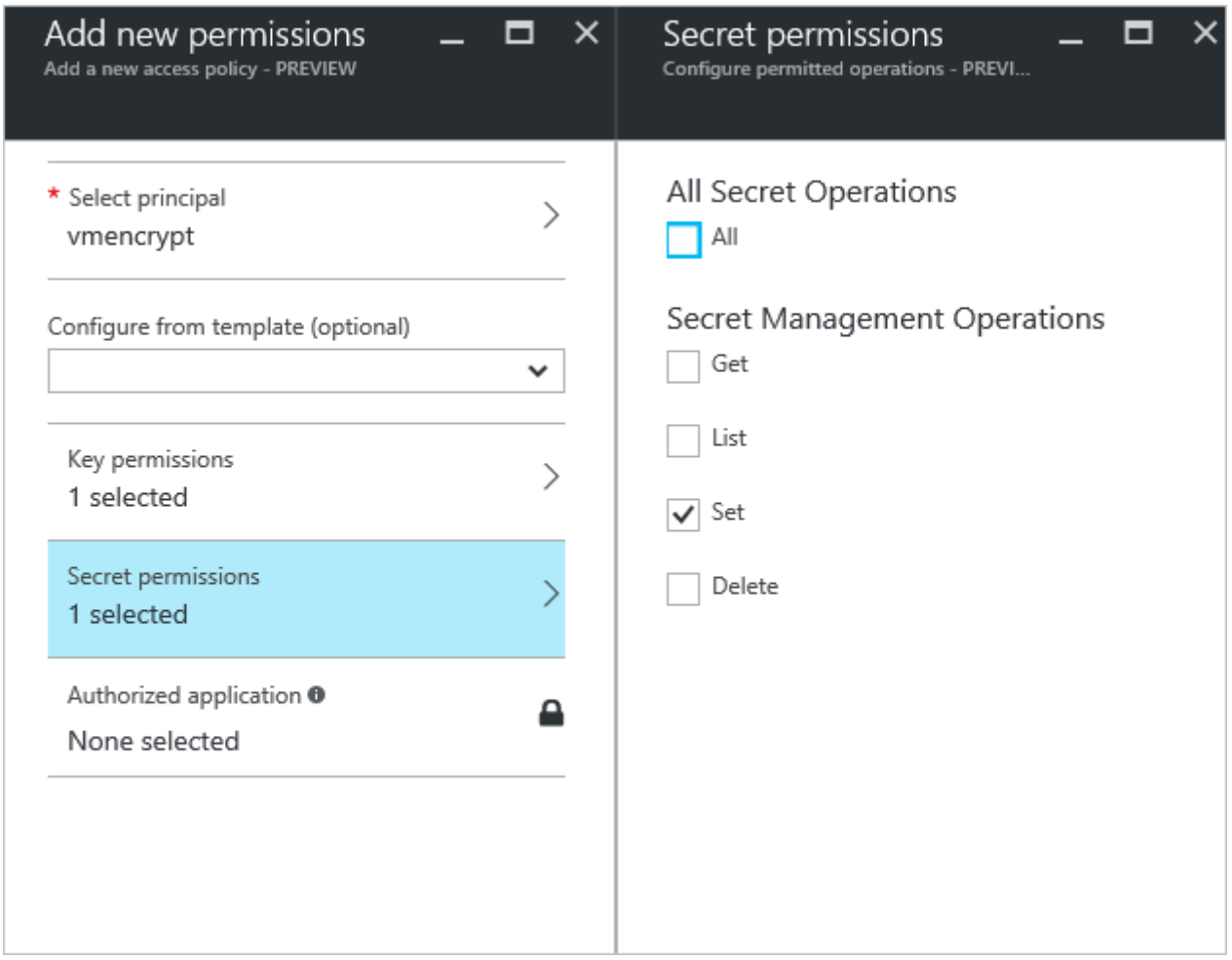
Azure CLI

```
az keyvault set-policy --name "MySecureVault" --spn "<spn created with CLI/the
Azure AD ClientID>" --key-permissions wrapKey --secret-permissions set
```

Set the key vault access policy for the Microsoft Entra app with the portal

1. Open the resource group with your key vault.
2. Select your key vault, go to **Access Policies**, then select **Add new**.
3. Under **Select principal**, search for the Microsoft Entra application you created and select it.
4. For **Key permissions**, check **Wrap Key** under **Cryptographic Operations**.
5. For **Secret permissions**, check **Set** under **Secret Management Operations**.
6. Select **OK** to save the access policy.


Add new permissions
Add a new access policy - PREVIEW
* Select principal
vmencrypt
Configure from template (optional)
Key permissions
1 selected
Secret permissions
1 selected
Authorized application
None selected
Key permissions
Configure permitted operations - PREVI...
All Key Operations
All
Key Management Operations
Get
List
Update
Create
Import
Delete
Backup
Restore
Cryptographic Operations
Decrypt
Encrypt
UnwrapKey
WrapKey
Verify
Sign



Set key vault advanced access policies

The Azure platform needs access to the encryption keys or secrets in your key vault to make them available to the VM for booting and decrypting the volumes. Enable disk encryption on the key vault or deployments will fail.

Set key vault advanced access policies with Azure PowerShell

Use the key vault PowerShell cmdlet Set-AzKeyVaultAccessPolicy to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** EnabledForDiskEncryption is required for Azure Disk encryption.

  Azure PowerShell

  ```
  Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName
  'MyKeyVaultResourceGroup' -EnabledForDiskEncryption
  ```

- **Enable Key Vault for deployment, if needed:** Enables the Microsoft. Compute resource provider to retrieve secrets from this key vault when this key vault is referenced in

resource creation, for example when creating a virtual machine.

Azure PowerShell

```
 Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName
'MyKeyVaultResourceGroup' -EnabledForDeployment
```

- **Enable Key Vault for template deployment, if needed:** Enables Azure Resource Manager to get secrets from this key vault when this key vault is referenced in a template deployment.

Azure PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName 'MySecureVault' -ResourceGroupName
'MyKeyVaultResourceGroup' -EnabledForTemplateDeployment
```

Set key vault advanced access policies using the Azure CLI

Use az keyvault update to enable disk encryption for the key vault.

- **Enable Key Vault for disk encryption:** Enabled-for-disk-encryption is required.

Azure CLI

```
az keyvault update --name "MySecureVault" --resource-group
"MyKeyVaultResourceGroup" --enabled-for-disk-encryption "true"
```

- **Enable Key Vault for deployment, if needed:** Allow Virtual Machines to retrieve certificates stored as secrets from the vault.

Azure CLI

```
az keyvault update --name "MySecureVault" --resource-group
"MyKeyVaultResourceGroup" --enabled-for-deployment "true"
```

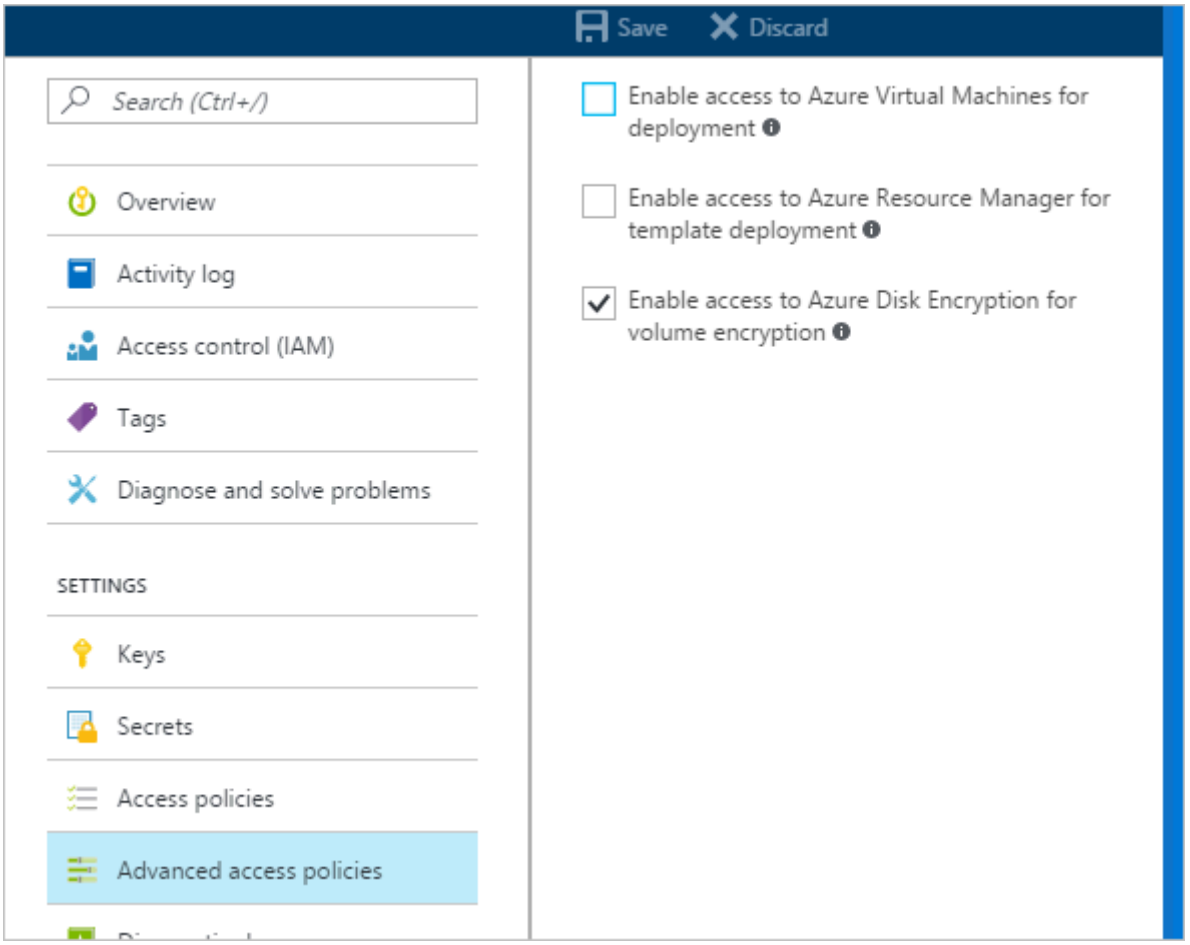
- **Enable Key Vault for template deployment, if needed:** Allow Resource Manager to retrieve secrets from the vault.

Azure CLI

```
az keyvault update --name "MySecureVault" --resource-group
"MyKeyVaultResourceGroup" --enabled-for-template-deployment "true"
```

Set key vault advanced access policies through the Azure portal

1. Select your keyvault, go to **Access Policies**, and **Click to show advanced access policies**.
2. Select the box labeled **Enable access to Azure Disk Encryption for volume encryption**.
3. Select **Enable access to Azure Virtual Machines for deployment** and/or **Enable Access to Azure Resource Manager for template deployment**, if needed.
4. Select **Save**.



Set up a key encryption key (optional)

If you want to use a key encryption key (KEK) for an additional layer of security for encryption keys, add a KEK to your key vault. Use the Add-AzKeyVaultKey cmdlet to create a key encryption key in the key vault. You can also import a KEK from your on-premises key management HSM. For more information, see Key Vault Documentation. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

- When generating keys, use an RSA key type. Azure Disk Encryption doesn't yet support using Elliptic Curve keys.

- Your key vault secret and KEK URLs must be versioned. Azure enforces this restriction of versioning. For valid secret and KEK URLs, see the following examples:
  - Example of a valid secret URL:
    `https://contosovault.vault.azure.net/secrets/EncryptionSecretWithKek/xxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxx`
  - Example of a valid KEK URL:
    `https://contosovault.vault.azure.net/keys/diskencryptionkek/xxxxxxxxxxxxxxxxxxxx xxxxxxxxxx`

- Azure Disk Encryption doesn't support specifying port numbers as part of key vault secrets and KEK URLs. For examples of non-supported and supported key vault URLs, see the following examples:
  - Unacceptable key vault URL
    `https://contosovault.vault.azure.net:443/secrets/contososecret/xxxxxxxxxxxxxxxxx xxxxxxxxxxxxx`
  - Acceptable key vault URL:
    `https://contosovault.vault.azure.net/secrets/contososecret/xxxxxxxxxxxxxxxxxxxx xxxxxxxxx`

Set up a key encryption key with Azure PowerShell

Before using the PowerShell script, you should be familiar with the Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment. This script creates all Azure Disk Encryption prerequisites and encrypts an existing IaaS VM, wrapping the disk encryption key by using a key encryption key.

PowerShell

```
# Step 1: Create a new resource group and key vault in the same location.
     # Fill in 'MyLocation', 'MyKeyVaultResourceGroup', and 'MySecureVault' with
your values.
     # Use Get-AzLocation to get available locations and use the DisplayName.
     # To use an existing resource group, comment out the line for New-
AzResourceGroup

    $Loc = 'MyLocation';
    $KVRGname = 'MyKeyVaultResourceGroup';
    $KeyVaultName = 'MySecureVault';
    New-AzResourceGroup –Name  $KVRGname –Location $Loc;
    New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location
$Loc;
    $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
    $KeyVaultResourceId = (Get-AzKeyVault -VaultName $KeyVaultName -
ResourceGroupName  $KVRGname).ResourceId;
```

```
    $diskEncryptionKeyVaultUrl = (Get-AzKeyVault -VaultName $KeyVaultName -
ResourceGroupName  $KVRGname).VaultUri;

# Step 2: Create the AD application and service principal.
     # Fill in 'MyAADClientSecret', "<My Application Display Name>", "
<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with your
values.
     # MyApplicationHomePage and the MyApplicationUri can be any values you wish.

     $aadClientSecret =  'MyAADClientSecret';
    $aadClientSecretSec = ConvertTo-SecureString -String $aadClientSecret -
AsPlainText -Force;
    $azureAdApplication = New-AzADApplication -DisplayName "<My Application
Display Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "
<https://MyApplicationUri>" -Password $aadClientSecretSec
    $servicePrincipal = New-AzADServicePrincipal –ApplicationId
$azureAdApplication.ApplicationId -Role Contributor;
    $aadClientID = $azureAdApplication.ApplicationId;

#Step 3: Enable the vault for disk encryption and set the access policy for the
Microsoft Entra application.

     Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$KVRGname -EnabledForDiskEncryption;
    Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName  $KVRGname;

#Step 4: Create a new key in the key vault with the Add-AzKeyVaultKey cmdlet.
     # Fill in 'MyKeyEncryptionKey' with your value.

     $keyEncryptionKeyName = 'MyKeyEncryptionKey';
    Add-AzKeyVaultKey -VaultName $KeyVaultName -Name $keyEncryptionKeyName -
Destination 'Software';
    $keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;

#Step 5: Encrypt the disks of an existing IaaS VM
     # Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

     $VMName = 'MySecureVM';
     $VMRGName = 'MyVirtualMachineResourceGroup';
    Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName -
AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $KeyVaultResourceId;
```

Certificate-based authentication (optional)

If you would like to use certificate authentication, you can upload one to your key vault and deploy it to the client. Before using the PowerShell script, you should be familiar with the Azure

Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

PowerShell

```
# Fill in "MyKeyVaultResourceGroup", "MySecureVault", and 'MyLocation' ('My
location' only if needed)

  $KVRGname = 'MyKeyVaultResourceGroup'
  $KeyVaultName= 'MySecureVault'

  # Create a key vault and set enabledForDiskEncryption property on it.
  # Comment out the next three lines if you already have an existing key vault
enabled for encryption. No need to set 'My location' in this case.

  $Loc = 'MyLocation'
  New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location
$Loc
  Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $KVRGname
-EnabledForDiskEncryption

  #Setting some variables with the key vault information
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname
  $DiskEncryptionKeyVaultUrl = $KeyVault.VaultUri
  $KeyVaultResourceId = $KeyVault.ResourceId

  # Create the Microsoft Entra application and associate the certificate with it.
  # Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display
Name>", "<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with
your values.
  # MyApplicationHomePage and the MyApplicationUri can be any values you wish

  $CertPath = "C:\certificates\mycert.pfx"
  $CertPassword = "Password"
  $Cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath,
$CertPassword)
  $CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

  $AzureAdApplication = New-AzADApplication -DisplayName "<My Application Display
Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "
<https://MyApplicationUri>" -CertValue $CertValue
  $ServicePrincipal = New-AzADServicePrincipal -ApplicationId
$AzureAdApplication.ApplicationId -Role Contributor

  $AADClientID = $AzureAdApplication.ApplicationId
  $aadClientCertThumbprint= $cert.Thumbprint

  Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName $KVRGname

  # Upload the pfx file to the key vault.
```

```
  # Fill in "MyAADCert".

  $KeyVaultSecretName = "MyAADCert"
  $FileContentBytes = get-content $CertPath -Encoding Byte
  $FileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
          $JSONObject = @"
          {
              "data" : "$filecontentencoded",
              "dataType" : "pfx",
              "password" : "$CertPassword"
          }
"@

  $JSONObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
  $JSONEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

  #Set the secret and set the key vault policy for -EnabledForDeployment

  $Secret = ConvertTo-SecureString -String $JSONEncoded -AsPlainText -Force
  Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $KeyVaultSecretName -
SecretValue $Secret
  Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName $KVRGname
-EnabledForDeployment

  # Deploy the certificate to the VM
  # Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

  $VMName = 'MySecureVM'
  $VMRGName = 'MyVirtualMachineResourceGroup'
  $CertUrl = (Get-AzKeyVaultSecret -VaultName $KeyVaultName -Name
$KeyVaultSecretName).Id
  $SourceVaultId = (Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGName).ResourceId
  $VM = Get-AzVM -ResourceGroupName $VMRGName -Name $VMName
  $VM = Add-AzVMSecret -VM $VM -SourceVaultId $SourceVaultId -CertificateStore
"My" -CertificateUrl $CertUrl
  Update-AzVM -VM $VM -ResourceGroupName $VMRGName

  #Enable encryption on the VM using Microsoft Entra client ID and the client
certificate thumbprint

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -
AadClientID $AADClientID -AadClientCertThumbprint $AADClientCertThumbprint -
DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId
```

Certificate-based authentication and a KEK (optional)

If you would like to use certificate authentication and wrap the encryption key with a KEK, you can use the below script as an example. Before using the PowerShell script, you should be

familiar with all of the previous Azure Disk Encryption prerequisites to understand the steps in the script. The sample script might need changes for your environment.

PowerShell

```
# Fill in 'MyKeyVaultResourceGroup', 'MySecureVault', and 'MyLocation' (if needed)

   $KVRGname = 'MyKeyVaultResourceGroup'
   $KeyVaultName= 'MySecureVault'

   # Create a key vault and set enabledForDiskEncryption property on it.
   # Comment out the next three lines if you already have an existing key vault
enabled for encryption.

   $Loc = 'MyLocation'
   New-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname -Location
$Loc
   Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$KVRGname -EnabledForDiskEncryption

   # Create the Azure AD application and associate the certificate with it.
   # Fill in "C:\certificates\mycert.pfx", "Password", "<My Application Display
Name>", "<https://MyApplicationHomePage>", and "<https://MyApplicationUri>" with
your values.
   # MyApplicationHomePage and the MyApplicationUri can be any values you wish

   $CertPath = "C:\certificates\mycert.pfx"
   $CertPassword = "Password"
   $Cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath,
$CertPassword)
   $CertValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

   $AzureAdApplication = New-AzADApplication -DisplayName "<My Application Display
Name>" -HomePage "<https://MyApplicationHomePage>" -IdentifierUris "
<https://MyApplicationUri>" -CertValue $CertValue
   $ServicePrincipal = New-AzADServicePrincipal -ApplicationId
$AzureAdApplication.ApplicationId -Role Contributor

   $AADClientID = $AzureAdApplication.ApplicationId
   $aadClientCertThumbprint= $cert.Thumbprint

   ## Give access for setting secrets and wraping keys
   Set-AzKeyVaultAccessPolicy -VaultName $keyVaultName -ServicePrincipalName
$aadClientID -PermissionsToKeys 'WrapKey' -PermissionsToSecrets 'Set' -
ResourceGroupName $KVRGname

   # Upload the pfx file to the key vault.
   # Fill in "MyAADCert".

   $KeyVaultSecretName = "MyAADCert"
   $FileContentBytes = get-content $CertPath -Encoding Byte
   $FileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
           $JSONObject = @"
```

```
            {
                "data" : "$filecontentencoded",
                "dataType" : "pfx",
                "password" : "$CertPassword"
            }
"@

   $JSONObjectBytes = [System.Text.Encoding]::UTF8.GetBytes($jsonObject)
   $JSONEncoded = [System.Convert]::ToBase64String($jsonObjectBytes)

   #Set the secret and set the key vault policy for deployment

   $Secret = ConvertTo-SecureString -String $JSONEncoded -AsPlainText -Force
   Set-AzKeyVaultSecret -VaultName $KeyVaultName -Name $KeyVaultSecretName -
SecretValue $Secret
   Set-AzKeyVaultAccessPolicy -VaultName $KeyVaultName -ResourceGroupName
$KVRGname -EnabledForDeployment

   #Setting some variables with the key vault information and generating a KEK
   # FIll in 'KEKName'

   $KEKName ='KEKName'
   $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname
   $DiskEncryptionKeyVaultUrl = $KeyVault.VaultUri
   $KeyVaultResourceId = $KeyVault.ResourceId
   $KEK = Add-AzKeyVaultKey -VaultName $KeyVaultName -Name $KEKName -Destination
"Software"
   $KeyEncryptionKeyUrl = $KEK.Key.kid



   # Deploy the certificate to the VM
   # Fill in 'MySecureVM' and 'MyVirtualMachineResourceGroup' with your values.

   $VMName = 'MySecureVM';
   $VMRGName = 'MyVirtualMachineResourceGroup';
   $CertUrl = (Get-AzKeyVaultSecret -VaultName $KeyVaultName -Name
$KeyVaultSecretName).Id
   $SourceVaultId = (Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGName).ResourceId
   $VM = Get-AzVM -ResourceGroupName $VMRGName -Name $VMName
   $VM = Add-AzVMSecret -VM $VM -SourceVaultId $SourceVaultId -CertificateStore
"My" -CertificateUrl $CertUrl
   Update-AzVM -VM $VM -ResourceGroupName $VMRGName

   #Enable encryption on the VM using Azure AD client ID and the client
certificate thumbprint

   Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -
AadClientID $AADClientID -AadClientCertThumbprint $AADClientCertThumbprint -
DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $KeyVaultResourceId
```

Next steps

Enable Azure Disk Encryption with Microsoft Entra ID on Windows VMs (previous release)

Azure Disk Encryption with Microsoft Entra ID for Windows VMs (previous release)

09/23/2025

> **Important**
>
> Azure Disk Encryption for Virtual Machines and Virtual Machine Scale Sets will be retired on **September 15, 2028**. New customers should use encryption at host for all new VMs. Existing customers should plan to migrate current ADE-enabled VMs to encryption at host before the retirement date to avoid service disruption -- see Migrate from Azure Disk Encryption to encryption at host.

Applies to: ✔ Windows VMs

The new release of Azure Disk Encryption eliminates the requirement for providing a Microsoft Entra application parameter to enable VM disk encryption. With the new release, you are no longer required to provide Microsoft Entra credentials during the enable encryption step. All new VMs must be encrypted without the Microsoft Entra application parameters using the new release. To view instructions to enable VM disk encryption using the new release, see Azure Disk Encryption for Windows VMS. VMs that were already encrypted with Microsoft Entra application parameters are still supported and should continue to be maintained with the Microsoft Entra syntax. You can enable many disk-encryption scenarios, and the steps may vary according to the scenario. The following sections cover the scenarios in greater detail for Windows IaaS VMs. Before you can use disk encryption, the Azure Disk Encryption prerequisites need to be completed.

> **Important**
>
> - You should take a snapshot and/or create a backup before disks are encrypted. Backups ensure that a recovery option is possible if an unexpected failure occurs during encryption. VMs with managed disks require a backup before encryption occurs. Once a backup is made, you can use the Set-AzVMDiskEncryptionExtension cmdlet to encrypt managed disks by specifying the -skipVmBackup parameter. For more information about how to back up and restore encrypted VMs, see Back up and restore encrypted Azure VM.
>
> - Encrypting or disabling encryption may cause a VM to reboot.

Enable encryption on new IaaS VMs created from the Marketplace

You can enable disk encryption on new IaaS Windows VM from the Marketplace in Azure using a Resource Manager template. The template creates a new encrypted Windows VM using the Windows Server 2012 gallery image.

1. On the Resource Manager template, click **Deploy to Azure**.
2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Purchase** to deploy a new IaaS VM where encryption is enabled.
3. After you deploy the template, verify the VM encryption status using your preferred method:

   - Verify with the Azure CLI by using the az vm encryption show command.

     Azure CLI

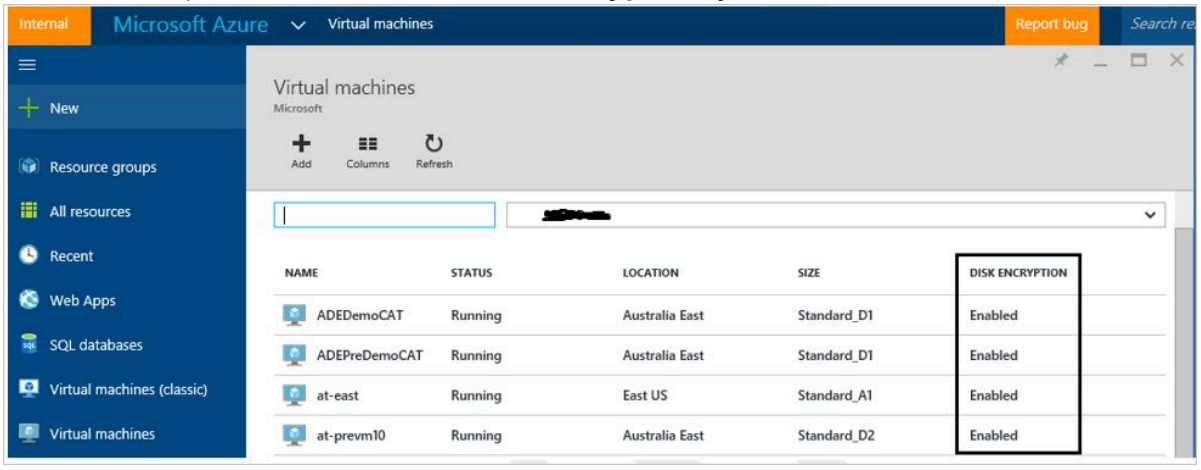
     ```
     az vm encryption show --name "MySecureVM" --resource-group
     "MyVirtualMachineResourceGroup"
     ```

   - Verify with Azure PowerShell by using the Get-AzVmDiskEncryptionStatus cmdlet.

     Azure PowerShell

     ```
     Get-AzVmDiskEncryptionStatus -ResourceGroupName
     'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
     ```

   - Select the VM, then click on **Disks** under the **Settings** heading to verify encryption status in the portal. In the chart under **Encryption**, you'll see if it's enabled.



The following table lists the Resource Manager template parameters for new VMs from the Marketplace scenario using Microsoft Entra client ID:

Expand table

Parameter	Description
adminUserName	Admin user name for the virtual machine.
adminPassword	Admin user password for the virtual machine.
newStorageAccountName	Name of the storage account to store OS and data VHDs.
vmSize	Size of the VM. Currently, only Standard A, D, and G series are supported.
virtualNetworkName	Name of the VNet that the VM NIC should belong to.
subnetName	Name of the subnet in the VNet that the VM NIC should belong to.
AADClientID	Client ID of the Microsoft Entra application that has permissions to write secrets to your key vault.
AADClientSecret	Client secret of the Microsoft Entra application that has permissions to write secrets to your key vault.
keyVaultURL	URL of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet `(Get-AzKeyVault -VaultName "MyKeyVault" -ResourceGroupName "MyKeyVaultResourceGroupName").VaultURI` or the Azure CLI `az keyvault show --name "MySecureVault" --query properties.vaultUri`
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated BitLocker key (optional). KeyEncryptionKeyURL is an optional parameter. You can bring your own KEK to further safeguard the data encryption key (Passphrase secret) in your key vault.
keyVaultResourceGroup	Resource group of the key vault.
vmName	Name of the VM that the encryption operation is to be performed on.

Enable encryption on existing or running IaaS Windows VMs

In this scenario, you can enable encryption by using a template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail how to enable Azure Disk Encryption.

Enable encryption on existing or running VMs with Azure PowerShell

Use the Set-AzVMDiskEncryptionExtension cmdlet to enable encryption on a running IaaS virtual machine in Azure. For information about enabling encryption with Azure Disk Encryption by using PowerShell cmdlets, see the blog posts Explore Azure Disk Encryption with Azure PowerShell - Part 1 and Explore Azure Disk Encryption with Azure PowerShell - Part 2.

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, Microsoft Entra app, and client secret should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values.

  Azure PowerShell

  ```
  $KVRGname = 'MyKeyVaultResourceGroup';
  $VMRGName = 'MyVirtualMachineResourceGroup';
  $vmName = 'MySecureVM';
  $aadClientID = 'My-AAD-client-ID';
  $aadClientSecret = 'My-AAD-client-secret';
  $KeyVaultName = 'MySecureVault';
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
  $KVRGname;
  $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
  $KeyVaultResourceId = $KeyVault.ResourceId;

  Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $vmName
  -AadClientID $aadClientID -AadClientSecret $aadClientSecret -
  DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
  DiskEncryptionKeyVaultId $KeyVaultResourceId;
  ```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault.

  Azure PowerShell

  ```
  $KVRGname = 'MyKeyVaultResourceGroup';
  $VMRGName = 'MyVirtualMachineResourceGroup';
  $vmName = 'MyExtraSecureVM';
  $aadClientID = 'My-AAD-client-ID';
  $aadClientSecret = 'My-AAD-client-secret';
  $KeyVaultName = 'MySecureVault';
  $keyEncryptionKeyName = 'MyKeyEncryptionKey';
  $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
  $KVRGname;
  $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
  $KeyVaultResourceId = $KeyVault.ResourceId;
  $keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
  $keyEncryptionKeyName).Key.kid;
  ```

```
Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName
-AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -KeyEncryptionKeyUrl
$keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $KeyVaultResourceId;
```

> **Note**
>
> The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the Get-AzVmDiskEncryptionStatus cmdlet.

Azure PowerShell

```
Get-AzVmDiskEncryptionStatus -ResourceGroupName
'MyVirtualMachineResourceGroup' -VMName 'MySecureVM'
```

- **Disable disk encryption:** To disable the encryption, use the Disable-AzureRmVMDisk Encryption cmdlet.

Azure PowerShell

```
Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup'
-VMName 'MySecureVM'
```

Enable encryption on existing or running VMs with Azure CLI

Use the az vm encryption enable command to enable encryption on a running IaaS virtual machine in Azure.

- **Encrypt a running VM using a client secret:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --
name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD
```

```
ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-encryption-
keyvault "MySecureVault" --volume-type [All|OS|Data]
```

- **Encrypt a running VM using KEK to wrap the client secret:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --
name "MySecureVM" --aad-client-id "<my spn created with CLI which is the
Azure AD ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-
encryption-keyvault  "MySecureVault" --key-encryption-key "MyKEK_URI" --key-
encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type
[All|OS|Data]
```

> **Note**
>
> The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

- **Verify the disks are encrypted:** To check on the encryption status of an IaaS VM, use the az vm encryption show command.

Azure CLI

```
az vm encryption show --name "MySecureVM" --resource-group
"MyVirtualMachineResourceGroup"
```

- **Disable encryption:** To disable encryption, use the az vm encryption disable command.

Azure CLI

```
az vm encryption disable --name "MySecureVM" --resource-group
"MyVirtualMachineResourceGroup" --volume-type [ALL, DATA, OS]
```

Using the Resource Manager template

You can enable disk encryption on existing or running IaaS Windows VMs in Azure by using the Resource Manager template to encrypt a running Windows VM.

1. On the Azure quickstart template, click **Deploy to Azure**.

2. Select the subscription, resource group, resource group location, parameters, legal terms, and agreement. Click **Purchase** to enable encryption on the existing or running IaaS VM.

The following table lists the Resource Manager template parameters for existing or running VMs that use a Microsoft Entra client ID:

Expand table

Parameter	Description
AADClientID	Client ID of the Microsoft Entra application that has permissions to write secrets to the key vault.
AADClientSecret	Client secret of the Microsoft Entra application that has permissions to write secrets to the key vault.
keyVaultName	Name of the key vault that the BitLocker key should be uploaded to. You can get it by using the cmdlet `(Get-AzKeyVault -ResourceGroupName <MyKeyVaultResourceGroupName>). Vaultname` or the Azure CLI command `az keyvault list --resource-group "MySecureGroup"`
keyEncryptionKeyURL	URL of the key encryption key that's used to encrypt the generated BitLocker key. This parameter is optional if you select **nokek** in the UseExistingKek drop-down list. If you select **kek** in the UseExistingKek drop-down list, you must enter the *keyEncryptionKeyURL* value.
volumeType	Type of volume that the encryption operation is performed on. Valid values are *OS*, *Data*, and *All*.
sequenceVersion	Sequence version of the BitLocker operation. Increment this version number every time a disk-encryption operation is performed on the same VM.
vmName	Name of the VM that the encryption operation is to be performed on.

New IaaS VMs created from customer-encrypted VHD and encryption keys

In this scenario, you can enable encrypting by using the Resource Manager template, PowerShell cmdlets, or CLI commands. The following sections explain in greater detail the Resource Manager template and CLI commands.

Use the instructions in the appendix for preparing pre-encrypted images that can be used in Azure. After the image is created, you can use the steps in the next section to create an encrypted Azure VM.

- Prepare a pre-encrypted Windows VHD

Encrypt VMs with pre-encrypted VHDs with Azure PowerShell

You can enable disk encryption on your encrypted VHD by using the PowerShell cmdlet Set-AzVMOSDisk. The example below gives you some common parameters.

PowerShell

```
$VirtualMachine = New-AzVMConfig -VMName "MySecureVM" -VMSize "Standard_A1"
$VirtualMachine = Set-AzVMOSDisk -VM $VirtualMachine -Name "SecureOSDisk" -VhdUri
"os.vhd" Caching ReadWrite -Windows -CreateOption "Attach" -DiskEncryptionKeyUrl
"https://mytestvault.vault.azure.net/secrets/Test1/514ceb769c984379a7e0230bddaaaaa
a" -DiskEncryptionKeyVaultId "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myKVresourcegroup/providers/Microsoft.KeyVault/vaults/
mytestvault"
New-AzVM -VM $VirtualMachine -ResourceGroupName "MyVirtualMachineResourceGroup"
```

Enable encryption on a newly added data disk

You can add a new disk to a Windows VM using PowerShell, or through the Azure portal.

Enable encryption on a newly added disk with Azure PowerShell

When using PowerShell to encrypt a new disk for Windows VMs, a new sequence version should be specified. The sequence version has to be unique. The script below generates a GUID for the sequence version. In some cases, a newly added data disk might be encrypted automatically by the Azure Disk Encryption extension. Auto encryption usually occurs when the VM reboots after the new disk comes online. This is typically caused because "All" was specified for the volume type when disk encryption previously ran on the VM. If auto encryption occurs on a newly added data disk, we recommend running the Set-AzVmDiskEncryptionExtension cmdlet again with new sequence version. If your new data disk is auto encrypted and you do not wish to be encrypted, decrypt all drives first then re-encrypt with a new sequence version specifying OS for the volume type.

- **Encrypt a running VM using a client secret:** The script below initializes your variables and runs the Set-AzVMDiskEncryptionExtension cmdlet. The resource group, VM, key vault, Microsoft Entra app, and client secret should have already been created as prerequisites. Replace MyKeyVaultResourceGroup, MyVirtualMachineResourceGroup, MySecureVM, MySecureVault, My-AAD-client-ID, and My-AAD-client-secret with your values. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

Azure PowerShell

```
 $sequenceVersion = [Guid]::NewGuid();
 $KVRGname = 'MyKeyVaultResourceGroup';
 $VMRGName = 'MyVirtualMachineResourceGroup';
 $vmName = 'MySecureVM';
 $aadClientID = 'My-AAD-client-ID';
 $aadClientSecret = 'My-AAD-client-secret';
 $KeyVaultName = 'MySecureVault';
 $KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
 $diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
 $KeyVaultResourceId = $KeyVault.ResourceId;

 Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName
-AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -VolumeType 'all' –
SequenceVersion $sequenceVersion;
```

- **Encrypt a running VM using KEK to wrap the client secret:** Azure Disk Encryption lets you specify an existing key in your key vault to wrap disk encryption secrets that were generated while enabling encryption. When a key encryption key is specified, Azure Disk Encryption uses that key to wrap the encryption secrets before writing to Key Vault. This example uses "All" for the -VolumeType parameter, which includes both OS and Data volumes. If you only want to encrypt the OS volume, use "OS" for the -VolumeType parameter.

Azure PowerShell

```
$sequenceVersion = [Guid]::NewGuid();
$KVRGname = 'MyKeyVaultResourceGroup';
$VMRGName = 'MyVirtualMachineResourceGroup';
$vmName = 'MyExtraSecureVM';
$aadClientID = 'My-AAD-client-ID';
$aadClientSecret = 'My-AAD-client-secret';
$KeyVaultName = 'MySecureVault';
$keyEncryptionKeyName = 'MyKeyEncryptionKey';
$KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName
$KVRGname;
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
$KeyVaultResourceId = $KeyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGname -VMName $vmName
-AadClientID $aadClientID -AadClientSecret $aadClientSecret -
DiskEncryptionKeyVaultUrl $diskEncryptionKeyVaultUrl -
DiskEncryptionKeyVaultId $KeyVaultResourceId -KeyEncryptionKeyUrl
$keyEncryptionKeyUrl -KeyEncryptionKeyVaultId $KeyVaultResourceId -VolumeType
```

```
'all' -SequenceVersion $sequenceVersion;
```

> **Note**
>
> The syntax for the value of disk-encryption-keyvault parameter is the full identifier string: /subscriptions/[subscription-id-guid]/resourceGroups/[resource-group-name]/providers/Microsoft.KeyVault/vaults/[keyvault-name]
> The syntax for the value of the key-encryption-key parameter is the full URI to the KEK as in: https://[keyvault-name].vault.azure.net/keys/[kekname]/[kek-unique-id]

Enable encryption on a newly added disk with Azure CLI

The Azure CLI command will automatically provide a new sequence version for you when you run the command to enable encryption. Acceptable values for the volume-yype parameter are All, OS, and Data. You may need to change the volume-type parameter to OS or Data if you're only encrypting one type of disk for the VM. The examples use "All" for the volume-type parameter.

- **Encrypt a running VM using a client secret:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-client-id "<my spn created with CLI/my Azure AD ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault "MySecureVault" --volume-type "All"
```

- **Encrypt a running VM using KEK to wrap the client secret:**

Azure CLI

```
az vm encryption enable --resource-group "MyVirtualMachineResourceGroup" --name "MySecureVM" --aad-client-id "<my spn created with CLI which is the Azure AD ClientID>"  --aad-client-secret "My-AAD-client-secret" --disk-encryption-keyvault  "MySecureVault" --key-encryption-key "MyKEK_URI" --key-encryption-keyvault "MySecureVaultContainingTheKEK" --volume-type "all"
```

Enable encryption using Microsoft Entra client certificate-based authentication.

You can use client certificate authentication with or without KEK. Before using the PowerShell scripts, you should already have the certificate uploaded to the key vault and deployed to the VM. If you're using KEK too, the KEK should already exist. For more information, see the Certificate-based authentication for Microsoft Entra ID section of the prerequisites article.

Enable encryption using certificate-based authentication with Azure PowerShell

PowerShell

```
## Fill in 'MyVirtualMachineResourceGroup', 'MyKeyVaultResourceGroup', 'My-AAD-client-ID', 'MySecureVault, and 'MySecureVM'.

$VMRGName = 'MyVirtualMachineResourceGroup'
$KVRGname = 'MyKeyVaultResourceGroup';
$AADClientID ='My-AAD-client-ID';
$KeyVaultName = 'MySecureVault';
$VMName = 'MySecureVM';
$KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
$KeyVaultResourceId = $KeyVault.ResourceId;

# Fill in the certificate path and the password so the thumbprint can be set as a variable.

$certPath = '$CertPath = "C:\certificates\mycert.pfx';
$CertPassword ='Password'
$Cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath,
$CertPassword)
$aadClientCertThumbprint = $cert.Thumbprint;

# Enable disk encryption using the client certificate thumbprint

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -
AadClientID $AADClientID -AadClientCertThumbprint $AADClientCertThumbprint -
DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId
```

Enable encryption using certificate-based authentication and a KEK with Azure PowerShell

PowerShell

```
# Fill in 'MyVirtualMachineResourceGroup', 'MyKeyVaultResourceGroup', 'My-AAD-client-ID', 'MySecureVault,, 'MySecureVM', and "KEKName.

$VMRGName = 'MyVirtualMachineResourceGroup';
```

```
$KVRGname = 'MyKeyVaultResourceGroup';
$AADClientID ='My-AAD-client-ID';
$KeyVaultName = 'MySecureVault';
$VMName = 'MySecureVM';
$keyEncryptionKeyName ='KEKName';
$KeyVault = Get-AzKeyVault -VaultName $KeyVaultName -ResourceGroupName $KVRGname;
$diskEncryptionKeyVaultUrl = $KeyVault.VaultUri;
$KeyVaultResourceId = $KeyVault.ResourceId;
$keyEncryptionKeyUrl = (Get-AzKeyVaultKey -VaultName $KeyVaultName -Name
$keyEncryptionKeyName).Key.kid;

## Fill in the certificate path and the password so the thumbprint can be read and
set as a variable.

$certPath = '$CertPath = "C:\certificates\mycert.pfx';
$CertPassword ='Password'
$Cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2($CertPath,
$CertPassword)
$aadClientCertThumbprint = $cert.Thumbprint;

# Enable disk encryption using the client certificate thumbprint and a KEK

Set-AzVMDiskEncryptionExtension -ResourceGroupName $VMRGName -VMName $VMName -
AadClientID $AADClientID -AadClientCertThumbprint $AADClientCertThumbprint -
DiskEncryptionKeyVaultUrl $DiskEncryptionKeyVaultUrl -DiskEncryptionKeyVaultId
$KeyVaultResourceId -KeyEncryptionKeyUrl $keyEncryptionKeyUrl -
KeyEncryptionKeyVaultId $KeyVaultResourceId
```

Disable encryption

You can disable encryption using Azure PowerShell, the Azure CLI, or with a Resource Manager template.

- **Disable disk encryption with Azure PowerShell:** To disable the encryption, use the Disable-AzureRmVMDiskEncryption cmdlet.

  Azure PowerShell

  ```
  Disable-AzVMDiskEncryption -ResourceGroupName 'MyVirtualMachineResourceGroup'
  -VMName 'MySecureVM'
  ```

- **Disable encryption with the Azure CLI:** To disable encryption, use the az vm encryption disable command.

  Azure CLI

  ```
  az vm encryption disable --name "MySecureVM" --resource-group
  ```

```
"MyVirtualMachineResourceGroup" --volume-type [ALL, DATA, OS]
```

- **Disable encryption with a Resource Manager Template:**

    1. Click **Deploy to Azure** from the Disable disk encryption on running Windows VM template.
    2. Select the subscription, resource group, location, VM, legal terms, and agreement.
    3. Click **Purchase** to disable disk encryption on a running Windows VM.

Next steps

Azure Disk Encryption overview

Overview of options to improve Azure managed disk performance

Article • 04/14/2025

This article outlines the capabilities each disk type offers to adjust its performance. Use the headings to navigate to the disk types you're interested in to get a quick overview of the capabilities available to it.

The demands and needs of your workload can shift over time, either due to high demand during a holiday, sudden bursts of traffic, or scaling up to meet client needs. Azure managed disks have several capabilities you can take advantage of to improve their performance and match the shifting needs of your workloads. Different disk types offer different capabilities, some disk types have capabilities you can use to ensure their performance automatically shifts to meet the changing demands of your workload, others require manual adjustment, and other disk types can't do either.

Ultra Disks and Premium SSD v2

Ultra Disks and Premium solid-state drives (SSD) v2 are designed to be highly performant and easily adjustable. They offer the most flexibility and ease among all the disk types when fine tuning your disk's performance, letting you programmatically (or directly) set the performance of these disk types. Within every 24 hours, you can adjust the performance of these disks up to four times. If you just created one of these disks, for the first 24 hours you can only adjust its performance up to three times. To learn how to adjust the performance of Ultra Disks and Premium SSD v2, see Adjust disk performance for Premium SSD v2 or Adjust the performance of an Ultra Disk.

Premium SSD

Premium SSD supports several performance options, each geared towards different use cases. The following table outlines the main differences and ideal uses.

Expand table

	Credit-based bursting	On-demand bursting	Changing performance tier	Caching	Write accelerator	Performance plus
Scenarios	Ideal for short-term	Ideal for short-term uses like	Ideal if your workload would	Ideal for workloads that can be	Ideal for scenarios where log	Ideal for workloads that require

	Credit-based bursting	On-demand bursting	Changing performance tier	Caching	Write accelerator	Performance plus
	scaling (30 minutes or less).	determining your workloads performance needs.	otherwise continually be running in burst.	configured to rely on data from the cache.	files are required to persist to a disk in a performant manner for modern databases.	higher sustained performance.
Cost	Free	Cost is variable, see the Billing section of the bursting article for details.	The cost of each performance tier is fixed, see Managed Disks pricing for details.	Free.	Free.	Free.
Availability	Only available for Premium SSD managed disks 512 GiB and smaller, and Standard SSDs 1,024 GiB and smaller.	Only available for Premium SSD managed disks larger than 512 GiB.	Available to all Premium SSD sizes.	Available to all Premium SSD managed disk, Standard SSD, and Standard HDD sizes.	Only available to M-series Azure Virtual Machines, but can be configured with all Premium SSD sizes attached to M-series virtual machines.	Available for Standard HDD, Standard SSD, and Premium SSD managed disks 513 GiB and larger.
Enablement	Enabled by default on eligible disks.	User must enable manually.	User must manually change their tier.	User must manually configure both their cache settings and their workloads and applications.	User must manually configure and enable.	Must enable performance plus when you create the disk, can't be enabled on existing disks.

Credit-based disk bursting

With credit-based bursting, a disk bursts only if it has burst credits accumulated in its credit bucket. This model doesn't incur extra charges when the disk bursts. For Premium SSD managed disks, credit-based bursting is available for disk sizes P20 and smaller. By default, disk bursting is enabled on all new and existing deployments of supported disk sizes. For more information, see the credit-based bursting section of the disk bursting models article.

On-demand disk bursting

With on-demand disk bursting enabled, the disk bursts whenever its needs exceed its current capacity. This model incurs extra charges anytime the disk bursts. On-demand bursting is only available for Premium SSDs larger than 512 GiB. To learn more, see the on-demand bursting section of the disk bursting models article.

Change performance tiers

The performance of a Premium SSD is set when you create your disk, in the form of their performance tier. When you set the provisioned size of your disk, a performance tier is automatically selected. The performance tier determines the IOPS and throughput your managed disk has. For Premium SSD disks only, the performance tier can be changed at deployment or afterwards, without changing the size of the disk, and without downtime. To learn more, see Performance tiers for managed disks.

Caching

High-scale Azure Virtual Machines that can use premium storage have a multitier caching technology called **BlobCache**. **BlobCache** is a caching solution built on RAM and SSDs that are locally attached to the server hosting the virtual machine (VM). This cache is available for Standard HDD, Standard SSD, and Premium SSD managed disks. With disk caching enabled, high-scale VMs can achieve levels of performance that exceed the underlying disk performance. To learn more, see Disk caching.

Write accelerator

Write accelerator is a disk capability for M-Series VMs on Premium SSD managed disks. Write accelerator improves the I/O latency of writes against Premium SSD disks. Write accelerator is optimized for volumes that contain the transaction log or redo logs of a DBMS, and shouldn't be used for data volumes. To learn more, see the write accelerator article.

Performance plus

Enabling performance plus increases the Input/Output Operations Per Second (IOPS) and throughput limits for Azure Premium SSDs that are 513 GiB and larger. Enabling performance plus improves the experience for workloads that require high IOPS and throughput, such as database and transactional workloads. There's no extra charge for enabling performance plus on a disk. To learn more about performance plus, see Increase IOPS and throughput limits for Azure Premium SSDs and Standard SSD/HDDs.

Standard SSD

Credit-based bursting

With credit-based bursting, a disk bursts only if it has burst credits accumulated in its credit bucket. This model doesn't incur extra charges when the disk bursts. For Standard SSDs, credit-based bursting is available for disk sizes E30 and smaller. By default, disk bursting is enabled on all new and existing deployments of supported disk sizes. For more information, see the credit-based bursting section of the disk bursting models article.

Caching

High-scale Azure Virtual Machines that can use premium storage have a multitier caching technology called **BlobCache**. **BlobCache** is a caching solution built on RAM and SSDs that are locally attached to the server hosting the virtual machine (VM). This cache is available for Standard HDD, Standard SSD, and Premium SSD managed disks. With disk caching enabled, high-scale VMs can achieve levels of performance that exceed the underlying disk performance. To learn more, see Disk caching.

Performance plus

Enabling performance plus increases the Input/Output Operations Per Second (IOPS) and throughput limits for Azure Standard SSDs that are 513 GiB and larger. Enabling performance plus improves the experience for workloads that require high IOPS and throughput, such as database and transactional workloads. There's no extra charge for enabling performance plus on a disk. To learn more about performance plus, see Increase IOPS and throughput limits for Azure Premium SSDs and Standard SSD/HDDs.

Standard HDD

Caching

High-scale Azure Virtual Machines that can use premium storage have a multitier caching technology called **BlobCache**. **BlobCache** is a caching solution built on RAM and SSDs that are locally attached to the server hosting the virtual machine (VM). This cache is available for Standard HDD, Standard SSD, and Premium SSD managed disks. With disk caching enabled, high-scale VMs can achieve levels of performance that exceed the underlying disk performance. To learn more, see Disk caching.

Performance plus

Enabling performance plus increases the Input/Output Operations Per Second (IOPS) and throughput limits for Azure Standard hard disk drives (HDD) that are 513 GiB and larger. Enabling performance plus improves the experience for workloads that require high IOPS and throughput, such as database and transactional workloads. There's no extra charge for enabling performance plus on a disk. To learn more about performance plus, see Increase IOPS and throughput limits for Azure Premium SSDs and Standard SSD/HDDs.

Next steps

- Learn about managed disk bursting
- Benchmark a disk
- Learn about disk performance metrics

Increase IOPS and throughput limits for Azure Premium SSDs and Standard SSD/HDDs

Article • 04/14/2025

The Input/Output Operations Per Second (IOPS) and throughput limits for Azure Premium solid-state drives (SSD), Standard SSDs, and Standard hard disk drives (HDD) that are 513 GiB and larger can be increased by enabling performance plus. Enabling performance plus improves the experience for workloads that require high IOPS and throughput, such as database and transactional workloads. There's no extra charge for enabling performance plus on a disk.

Once enabled, the IOPS and throughput limits for an eligible disk increase to the higher maximum limits. To see the new IOPS and throughput limits for eligible disks, consult the columns that begin with "*Expanded" in the Scalability and performance targets for VM disks article.

Limitations

- Can only be enabled on Standard HDD, Standard SSD, and Premium SSD managed disks that are 513 GiB or larger
- Can only be enabled on new disks
  - To work around this, create a snapshot of your disk, then create a new disk from the snapshot
- Not supported for disks recovered with Azure Site Recovery

Prerequisites

Either use the Azure Cloud Shell to run your commands or install a version of the Azure PowerShell module 9.5 or newer, or a version of the Azure CLI that is 2.44.0 or newer.

Enable performance plus

You need to create a new disk to use performance plus. The following scripts show how to create a disk with performance plus enabled and, if desired, attach it to a VM. The commands have been organized into self-contained steps for reliability.

Azure CLI

Create a resource group

This step creates a resource group with a unique name.

Azure CLI

```
export RANDOM_SUFFIX=$(openssl rand -hex 3)
export MY_RG="PerfPlusRG$RANDOM_SUFFIX"
export REGION="WestUS2"
az group create -g $MY_RG -l $REGION
```

Results:

JSON

```
{
  "id": "/subscriptions/xxxxx/resourceGroups/PerfPlusRGxxx",
  "location": "WestUS2",
  "name": "PerfPlusRGxxx",
  "properties": {
    "provisioningState": "Succeeded"
  }
}
```

Create a new disk with performance plus enabled

This step creates a new disk of 513 GiB (or larger) with performance plus enabled using a valid SKU value.

Azure CLI

```
export MY_DISK="PerfPlusDisk$RANDOM_SUFFIX"
export SKU="Premium_LRS"
export DISK_SIZE=513
az disk create -g $MY_RG -n $MY_DISK --size-gb $DISK_SIZE --sku $SKU -l
$REGION --performance-plus true
```

Results:

JSON

```
{
  "id":
"/subscriptions/xxxxx/resourceGroups/PerfPlusRGxxx/providers/Microsoft.Compute
/disks/PerfPlusDiskxxx",
```

```
  "location": "WestUS2",
  "name": "PerfPlusDiskxxx",
  "properties": {
    "provisioningState": "Succeeded",
    "diskSizeGb": 513,
    "sku": "Premium_LRS",
    "performancePlus": true
  },
  "type": "Microsoft.Compute/disks"
}
```


Attempt to attach the disk to a VM

This optional step attempts to attach the disk to an existing VM. It first checks if the VM exists and then proceeds accordingly.

Azure CLI

```
export MY_VM="NonExistentVM"
if az vm show -g $MY_RG -n $MY_VM --query "name" --output tsv >/dev/null 2>&1;
then
    az vm disk attach --vm-name $MY_VM --name $MY_DISK --resource-group $MY_RG
else
    echo "VM $MY_VM not found. Skipping disk attachment."
fi
```

Results:

text

```
VM NonExistentVM not found. Skipping disk attachment.
```

Create a new disk from an existing disk or snapshot with performance plus enabled

This series of steps creates a separate resource group and then creates a new disk from an existing disk or snapshot. Replace the SOURCE_URI with a valid source blob URI that belongs to the same region (WestUS2) as the disk.

Create a resource group for migration

Azure CLI

```
export RANDOM_SUFFIX=$(openssl rand -hex 3)
export MY_MIG_RG="PerfPlusMigrRG$RANDOM_SUFFIX"
```

```
export REGION="WestUS2"
az group create -g $MY_MIG_RG -l $REGION
```

Results:

JSON


```
{
  "id": "/subscriptions/xxxxx/resourceGroups/PerfPlusMigrRGxxx",
  "location": "WestUS2",
  "name": "PerfPlusMigrRGxxx",
  "properties": {
    "provisioningState": "Succeeded"
  }
}
```


Create the disk from an existing snapshot or disk

Azure CLI

```
# Create a snapshot from the original disk
export MY_SNAPSHOT_NAME="PerfPlusSnapshot$RANDOM_SUFFIX"
echo "Creating snapshot from original disk..."
az snapshot create \
  --name $MY_SNAPSHOT_NAME \
  --resource-group $MY_RG \
  --source $MY_DISK

# Get the snapshot ID for use as source
SNAPSHOT_ID=$(az snapshot show \
  --name $MY_SNAPSHOT_NAME \
  --resource-group $MY_RG \
  --query id \
  --output tsv)

echo "Using snapshot ID: $SNAPSHOT_ID"

# Create the new disk using the snapshot as source
export MY_MIG_DISK="PerfPlusMigrDisk$RANDOM_SUFFIX"
export SKU="Premium_LRS"
export DISK_SIZE=513

az disk create \
  --name $MY_MIG_DISK \
  --resource-group $MY_MIG_RG \
  --size-gb $DISK_SIZE \
  --performance-plus true \
  --sku $SKU \
  --source $SNAPSHOT_ID \
  --location $REGION
```

Results:

JSON


```
{
  "id":
"/subscriptions/xxxxx/resourceGroups/PerfPlusMigrRGxxx/providers/Microsoft.Com
pute/disks/PerfPlusMigrDiskxxx",
  "location": "WestUS2",
  "name": "PerfPlusMigrDiskxxx",
  "properties": {
    "provisioningState": "Succeeded",
    "diskSizeGb": 513,
    "sku": "Premium_LRS",
    "performancePlus": true,
    "source":
"https://examplestorageaccount.blob.core.windows.net/snapshots/sample-
westus2.vhd"
  },
  "type": "Microsoft.Compute/disks"
}
```

Prevent torn writes with Azure managed disks

Azure managed disks have native protection against torn writes for 8-KiB and 16-KiB blocks of data to ensure data integrity, which you can use to reduce your performance overhead.

A torn write (or a partial write) can occur when a power loss or system crash interrupts a disk write, leaving a data block only partially updated. A partially updated data block results in an inconsistent page containing a mix of old and new data, essentially a torn page. Torn writes compromise data integrity, and data integrity is critical for applications like databases. Databases must detect and resolve torn writes to avoid corrupt records or indexes

Applications can use managed disk's native torn write prevention through atomic write operations for 8-KiB and 16-KiB blocks of data if the issued write aligns with the respective block offset. For example, a 16 KiB I/O issued to the disk at an offset that is a multiple of 16-KiB guarantees atomicity. With atomic writes, managed disks persist the entire block or nothing, so there's never a partial update. Using atomic writes with managed disk's protection against torn writes allows applications to improve transaction throughput and reduce write latency without traditional software overhead for preventing torn writes.

Traditional torn write protection

PostgreSQL uses full page writes, so the entire page is logged to the write ahead log before updating data files. MySQL (InnoDB) uses a doublewrite buffer, writing each page twice (first to a protected area, then to main storage) to ensure at least one intact copy. Both of these methods are effective but, introduce performance overhead due to extra I/Os. With managed disk's torn write prevention, there's no extra performance overhead.

Limitations

- Your filesystem and your operating system must guarantee large atomic write support
  - Only supported for Linux VMs using version 6.13 or newer of the Linux kernel
    - As of 6.13, the Linux kernel introduced support for large atomic writes with direct IO for XFS and Ext4
- Atomic write operations are only available when using managed disks with NVMe controllers

Prerequisites

- Deploy a Linux VM with Linux kernel 6.13 or newer with a managed disk that's using an NVMe controller
- Install the **nvme-cli** package on your VM
  - If your distribution uses the Advanced Package Tool (APT), use `sudo apt install nvme-cli`
- 8-KiB and 16-KiB issued writes must align with their respective block offsets to ensure torn writes are prevented

Confirm your VM supports atomic writes

Access your VM and run `lsblk`, your disk should show up under the NVMe namespace (like nvme0n2).

Check the namespace parameters of the NVMe from the `lsblk` output with `sudo nvme id-ns <your-nvme-here>` (an example would be /dev/nvme0n2). Verify that the values of `nawun`, `nawupf`, and `nabspf` report values that indicate 16-KiB atomic wrote support. The output of these values should be either `31 (lbads:9)` or `3 (lbads:12)`.

Verify kernel atomic write support

Next, inspect atomic write parameters for 16-KiB atomic write size and alignment boundaries.

Bash

```
cat /sys/class/block/nvme0n2/queue/atomic_write_unit_max_bytes
cat /sys/class/block/nvme0n2/queue/atomic_write_unit_min_bytes
cat /sys/class/block/nvme0n2/queue/atomic_write_boundary_bytes
cat /sys/class/block/nvme0n2/queue/atomic_write_max_bytes
```

Set up and use the filesystem

Once you confirm that both the VM and the kernel support atomic writes, create an XFS filesystem with 16-KiB block size. You can create an XFS filesystem with the `mkfs.xfs` command. Include the following parameters `-b size=16384 <your-nvme-here>` (an example value for your NVMe is /dev/nvme0n2).

Now that you have setup the filesystem. Configure your application to write atomically with direct IO and the same block size formatted. One way of doing this is with the `fio` command. Include the following parameters: `direct=1 atomic=1 bs=16k`.

Last updated on 11/18/2025

Virtual machine and disk performance

Article • 04/01/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article helps clarify disk performance and how it works when you combine Azure Virtual Machines and Azure disks. It also describes how you can diagnose bottlenecks for your disk IO and the changes you can make to optimize for performance.

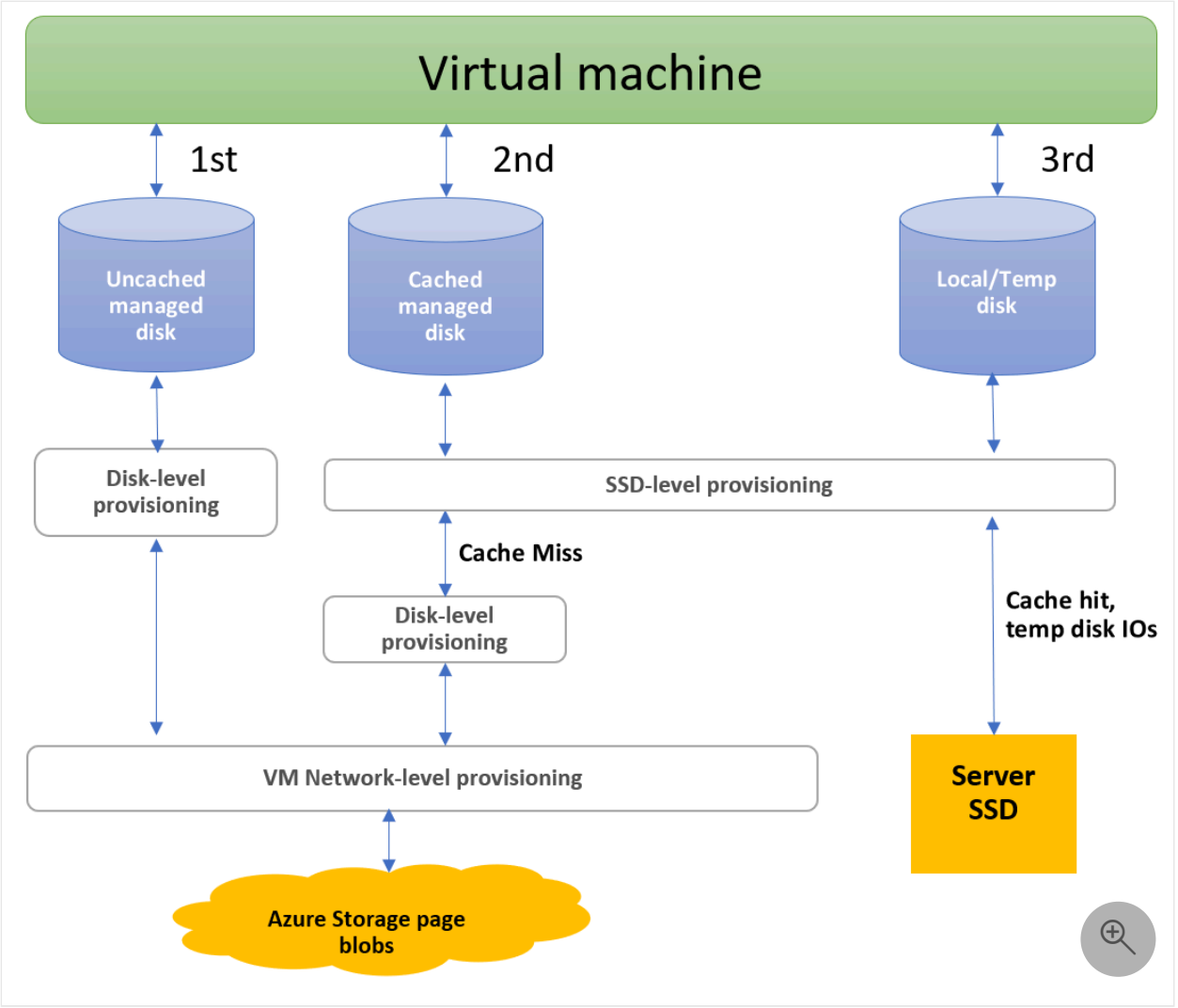
How does disk performance work?

Azure virtual machines (VM) have input/output operations per second (IOPS) and throughput performance limits based on the VM type and size. OS disks and data disks can be attached to virtual machines. The disks have their own IOPS and throughput limits.

Disk allocation and performance

There are three paths that bandwidth and I/O operations per second (IOPS) can take to disks attached to a VM in Azure. The following diagram depicts real-time allocation of these paths.

The following diagram depicts real-time allocation of bandwidth and I/O operations per second (IOPS) for disks, with three paths that I/O can take.



The first I/O path is the uncached managed disk path. I/O operations use this path when you're using a managed disk and you set the host caching to `none`. I/O operations that use this path run based on disk-level provisioning and then VM network-level provisioning for IOPS and throughput.

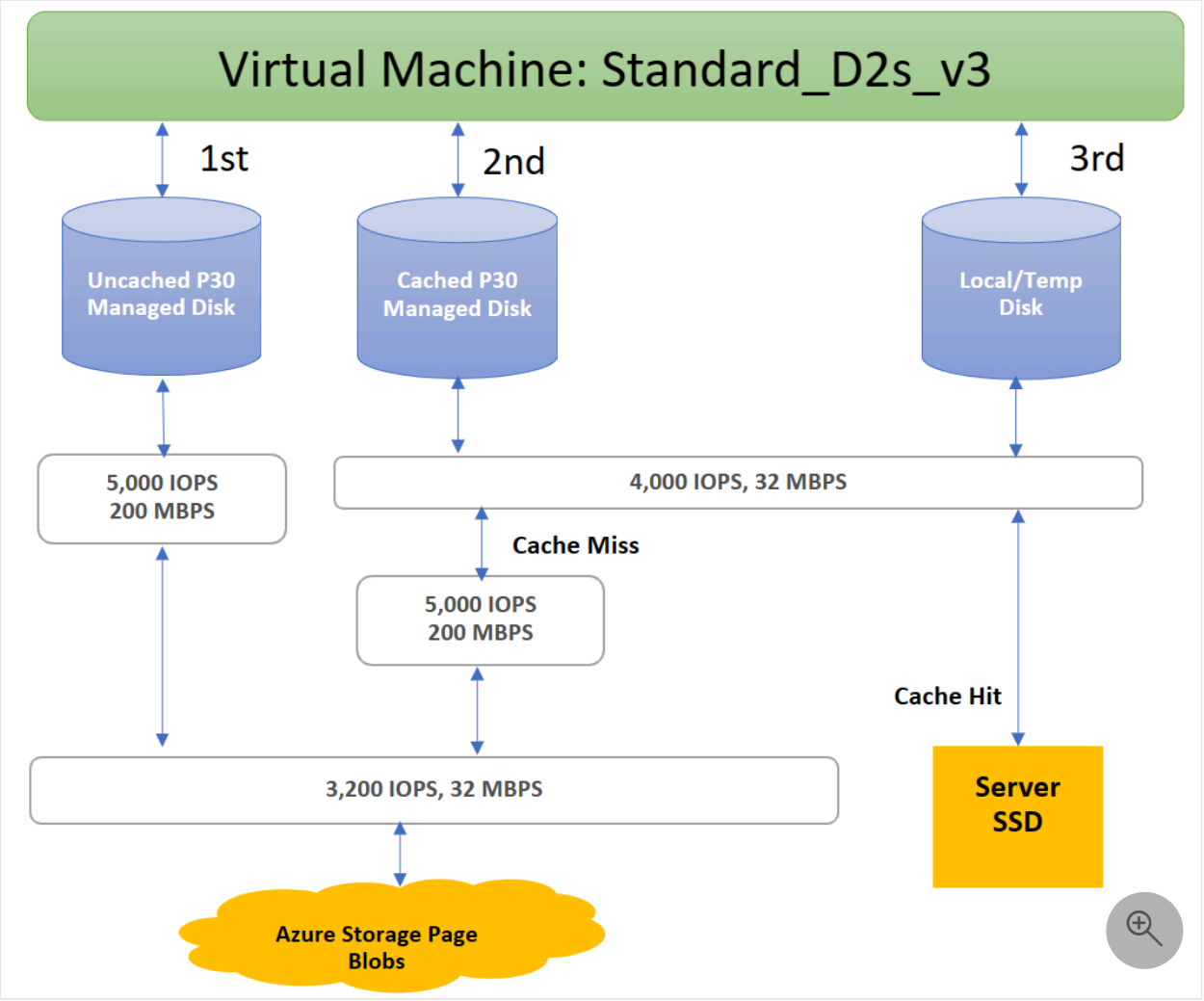
The second I/O path is the cached managed disk path. Cached managed disk I/O uses an SSD that's close to the VM. This SSD has its own IOPS and throughput provisioned, and it appears as "SSD-level provisioning" in the diagram.

When a cached managed disk initiates a read, the request first checks to see if the data is in the server SSD. If the data isn't present, a cached miss occurs. Then the I/O runs based on SSD-level provisioning, disk-level provisioning, and then VM network-level provisioning for IOPS and throughput.

When the server SSD initiates reads on cached I/O that are present on the server SSD, a cache hit occurs. The I/O then runs based on the SSD-level provisioning. Writes that a cached managed disk initiates always follow the path of a cached miss. They go through SSD-level, disk-level, and VM network-level provisioning.

The third path is for the *Local/Temp* disk. It's only available on VMs that support *Local/Temp* disks. An I/O operation that uses this path runs based on SSD-level provisioning for IOPS and throughput.

The following diagram depicts an example of these limitations. The system prevents a Standard_D2s_v3 VM from achieving the 5,000 IOPS potential of a P30 disk, whether it's cached or not, because of limits at the SSD and network levels.



Azure uses a prioritized network channel for disk traffic. Disk traffic takes precedence over low-priority network traffic. This prioritization helps disks maintain their expected performance if there's network contention.

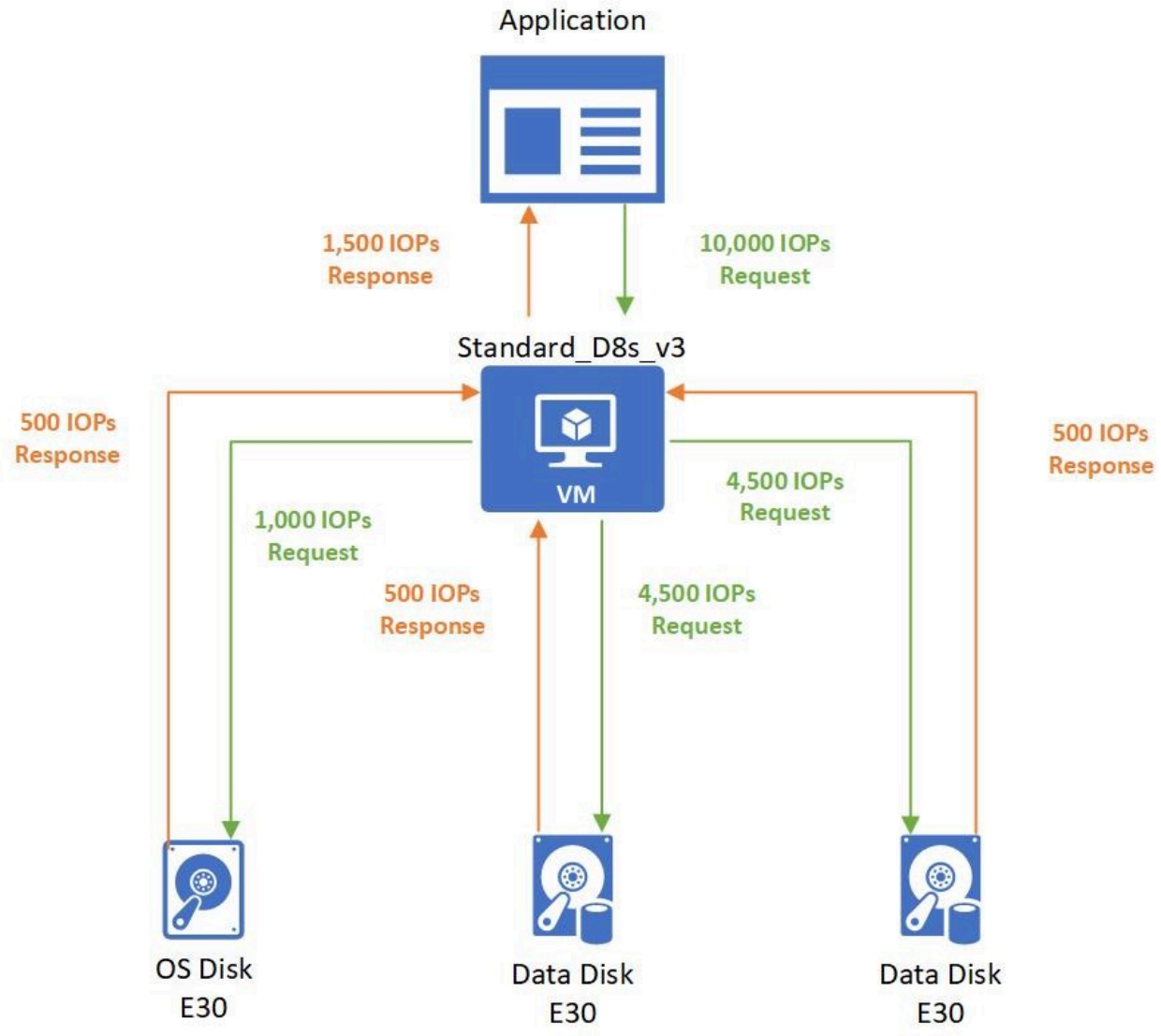
Similarly, Azure Storage handles resource contentions and other issues in the background with automatic load balancing. Azure Storage allocates required resources when you create a disk, and it applies proactive and reactive balancing of resources to handle the traffic level. This behavior further ensures that disks can sustain their expected IOPS and throughput targets. Use VM-level and disk-level metrics to track the performance and set up alerts as needed.

Disk IO capping

Your application's performance gets capped when it requests more IOPS or throughput than what is allotted for the virtual machines or attached disks. When capped, the application experiences suboptimal performance. This can lead to negative consequences like increased latency. Let's run through a couple of examples to clarify this concept. To make these examples easy to follow, we'll only look at IOPS. But, the same logic applies to throughput.

Setup:

- Standard_D8s_v3
  - Uncached IOPS: 12,800
- E30 OS disk
  - IOPS: 500
- Two E30 data disks × 2
  - IOPS: 500



The application running on the virtual machine makes a request that requires 10,000 IOPS to the virtual machine. All of which are allowed by the VM because the Standard_D8s_v3 virtual machine can execute up to 12,800 IOPS.

The 10,000 IOPS requests are broken down into three different requests to the different disks:

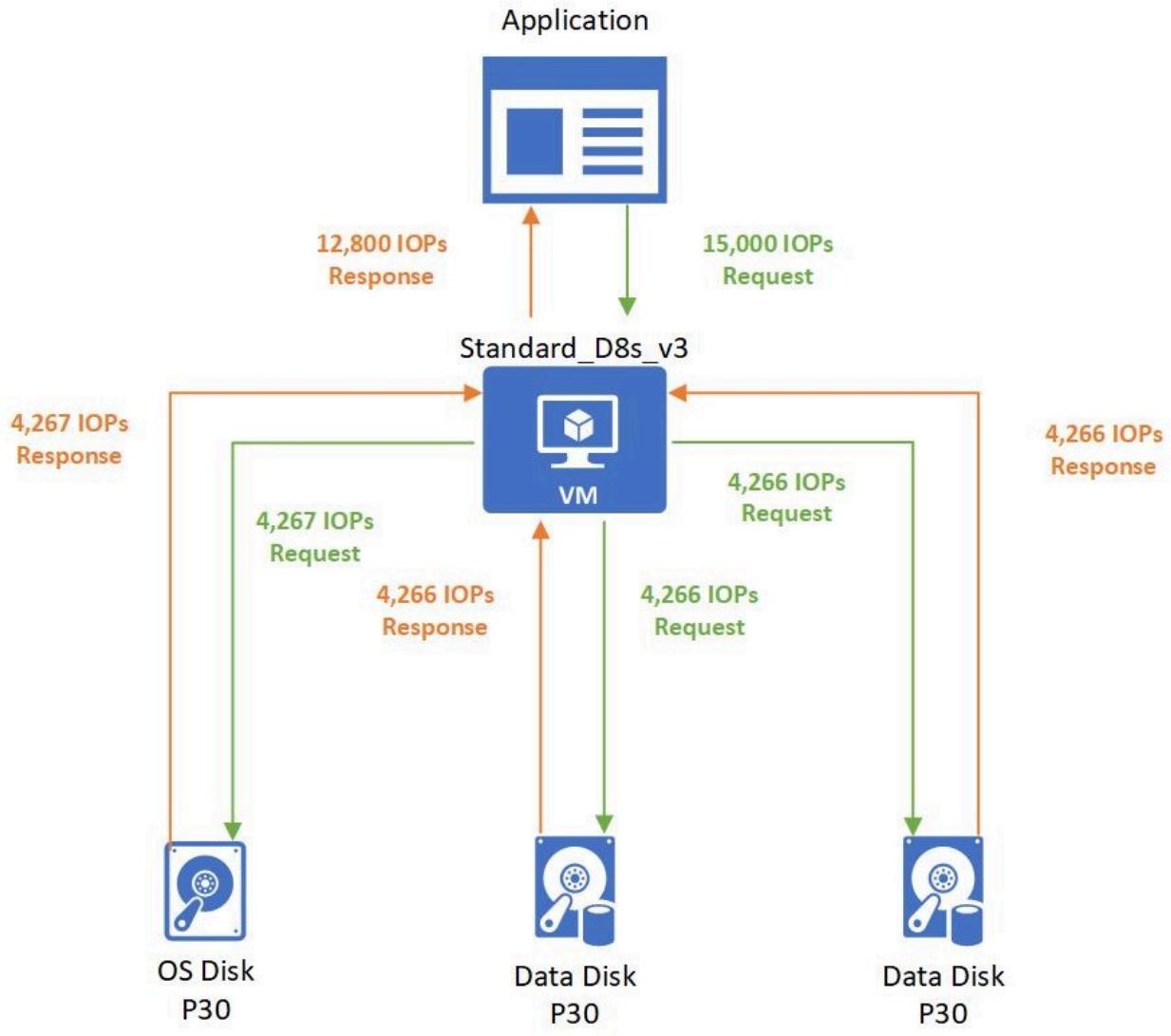
- 1,000 IOPS are requested to the operating system disk.
- 4,500 IOPS are requested to each data disk.

All attached disks are E30 disks and can only handle 500 IOPS. So, they respond back with 500 IOPS each. The application's performance is capped by the attached disks, and it can only process 1,500 IOPS. The application could work at peak performance at 10,000 IOPS if better-performing disks are used, such as Premium SSD P30 disks.

Virtual machine IO capping

Setup:

- Standard_D8s_v3
  - Uncached IOPS: 12,800
- P30 OS disk
  - IOPS: 5,000
- Two P30 data disks × 2
  - IOPS: 5,000



The application running on the virtual machine makes a request that requires 15,000 IOPS. Unfortunately, the Standard_D8s_v3 virtual machine is only provisioned to handle 12,800 IOPS. The application is capped by the virtual machine limits and must allocate the allotted 12,800 IOPS.

Those 12,800 IOPS requested are broken down into three different requests to the different disks:

- 4,267 IOPS are requested to the operating system disk.
- 4,266 IOPS are requested to each data disk.

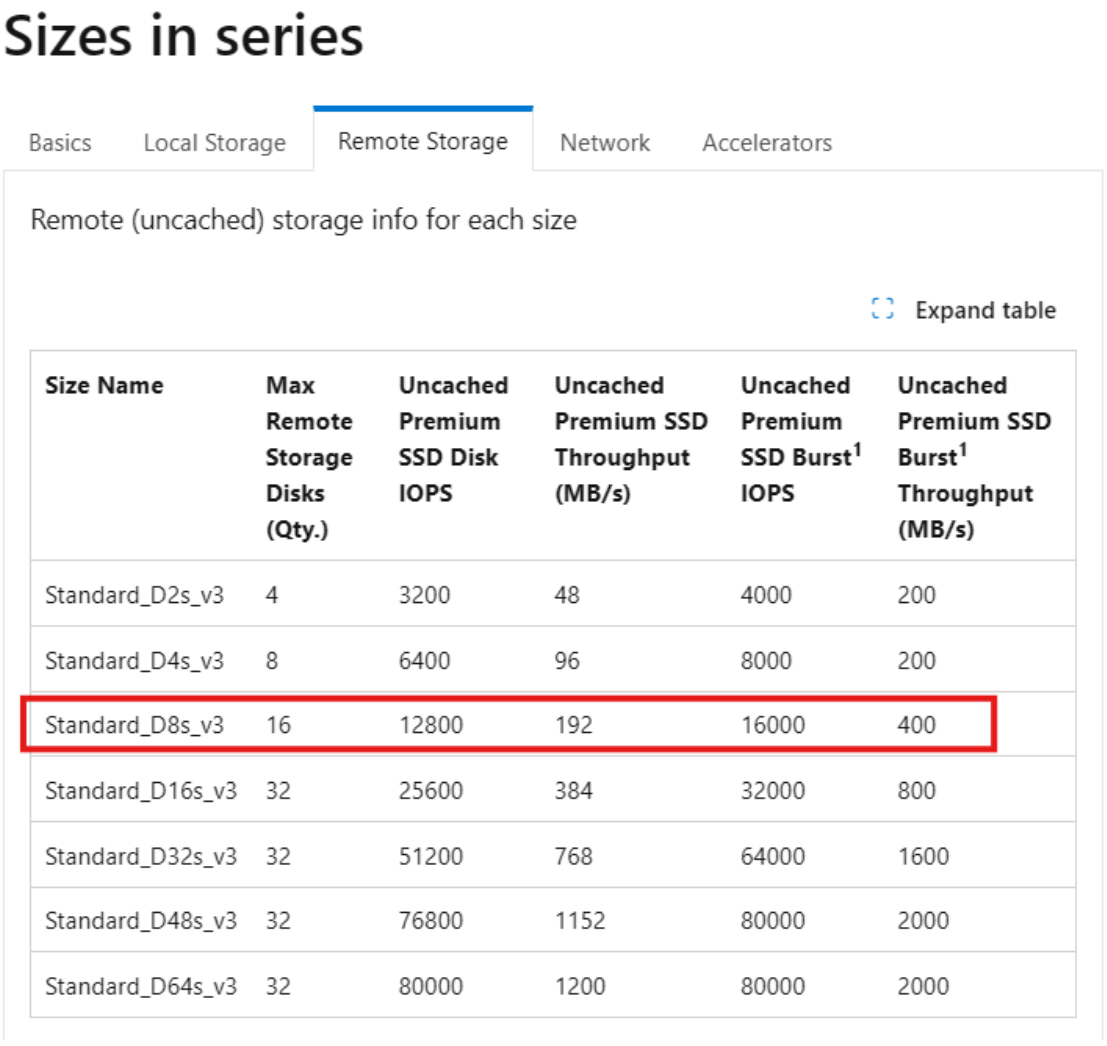
All attached disks are P30 disks that can handle 5,000 IOPS. So, they respond back with their requested amounts.

Virtual machine uncached vs cached limits

Virtual machines that are enabled for both premium storage and premium storage caching have two different storage bandwidth limits. Let's look at the Standard_D8s_v3

virtual machine as an example. Here is the documentation on the Dsv3-series and the Standard_D8s_v3:

- The "*Uncached*" disk data under the **Remote Storage** are the default storage maximum limits that the virtual machine can handle.



Sizes in series

Basics | Local Storage | Remote Storage | Network | Accelerators

Remote (uncached) storage info for each size

Expand table

Size Name	Max Remote Storage Disks (Qty.)	Uncached Premium SSD Disk IOPS	Uncached Premium SSD Throughput (MB/s)	Uncached Premium SSD Burst[1] IOPS	Uncached Premium SSD Burst[1] Throughput (MB/s)
Standard_D2s_v3	4	3200	48	4000	200
Standard_D4s_v3	8	6400	96	8000	200
Standard_D8s_v3	16	12800	192	16000	400
Standard_D16s_v3	32	25600	384	32000	800
Standard_D32s_v3	32	51200	768	64000	1600
Standard_D48s_v3	32	76800	1152	80000	2000
Standard_D64s_v3	32	80000	1200	80000	2000

- The "*Cached*" disk data under the **Local Storage** tab are separate limits when you enable host caching.

Sizes in series

Basics | Local Storage | Remote Storage | Network | Accelerators

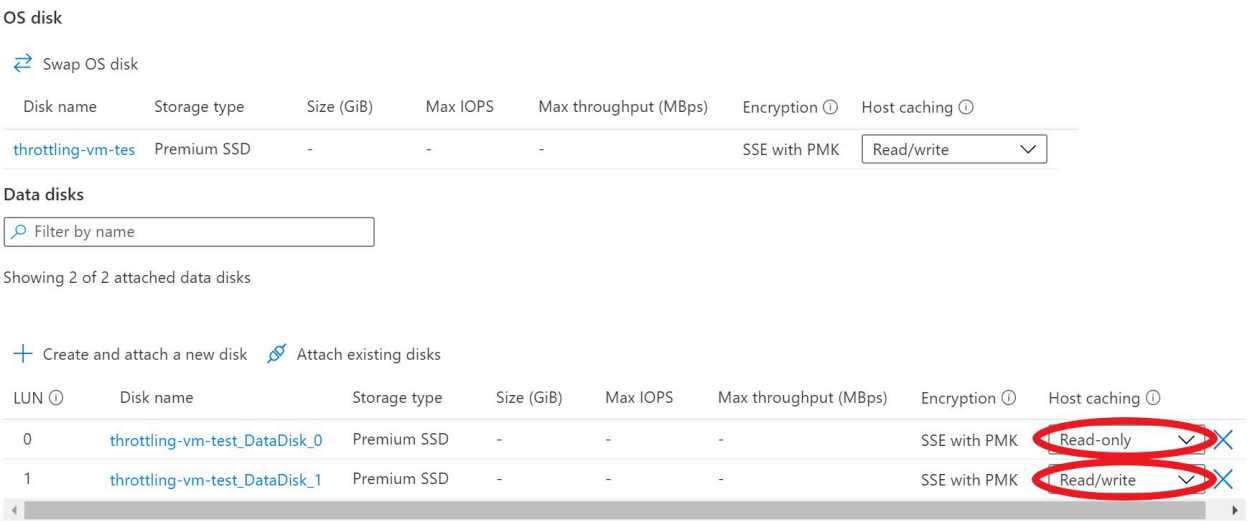
Local (cached & temp) storage info for each size

Expand table

Size Name	Max Temp Storage Disks (Qty.)	Temp Disk Size (GiB)	Disk Cache Size (GiB)	Cached Disk Random Read (RR)[1] IOPS	Cached Disk Random Read (RR)[1] Throughput (MB/s)
Standard_D2s_v3	1	16	50	4000	32
Standard_D4s_v3	1	32	100	8000	64
Standard_D8s_v3	1	64	200	16000	128
Standard_D16s_v3	1	128	400	32000	256
Standard_D32s_v3	1	256	800	64000	512
Standard_D48s_v3	1	384	1200	96000	768
Standard_D64s_v3	1	512	1600	128000	1024

Host caching works by bringing storage closer to the VM that can be written or read to quickly. The amount of storage that is available to the VM for host caching is in the documentation. For example, you can see the Standard_D8s_v3 comes with 200 GiB of cache storage.

You can enable host caching when you create your virtual machine and attach disks. You can also turn on and off host caching on your disks on an existing VM. By default, cache-capable data disks do not have caching enabled. Cache-capable OS disks have read/write caching enabled.



You can adjust the host caching to match your workload requirements for each disk. You can set your host caching to be:

- **Read-only**: For workloads that only do read operations
- **Read/write**: For workloads that do a balance of read and write operations

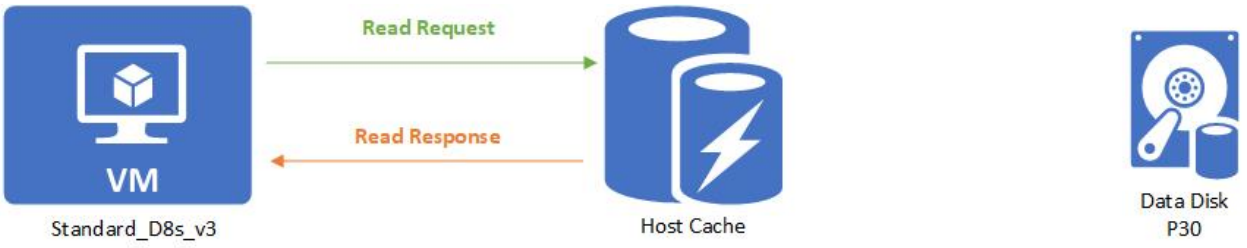
If your workload doesn't follow either of these patterns, we don't recommend that you use host caching.

Let's run through a couple examples of different host cache settings to see how it affects the data flow and performance. In this first example, we see what happens with IO requests when the host caching setting is set to **Read-only**.

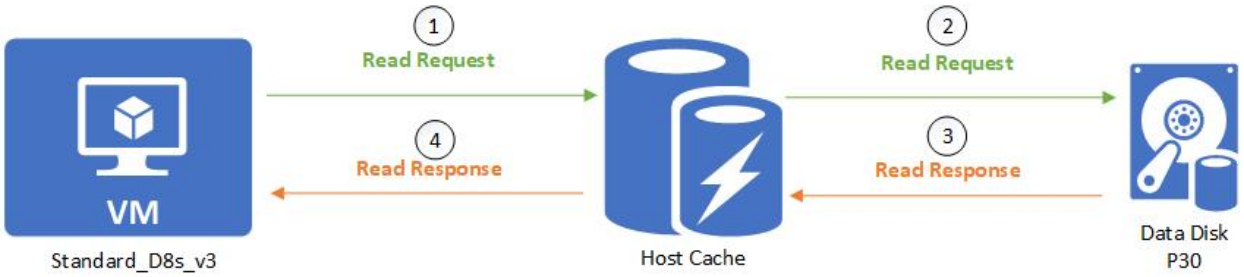
Setup:

- Standard_D8s_v3
  - Cached IOPS: 16,000
  - Uncached IOPS: 12,800
- P30 data disk
  - IOPS: 5,000
  - Host caching: **Read-only**

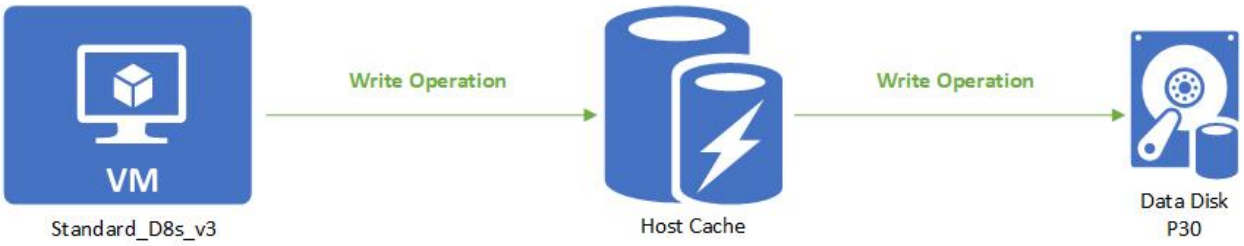
When a read is performed and the desired data is available on the cache, the cache returns the requested data. There's no need to read from the disk. This read is counted toward the VM's cached limits.



When a read is performed and the desired data *isn't* available on the cache, the read request is relayed to the disk. Then the disk surfaces it to both the cache and the VM. This read is counted toward both the VM's uncached limit and the VM's cached limit.



When a write is performed, the write has to be written to both the cache and the disk before it's considered complete. This write is counted toward the VM's uncached limit and the VM's cached limit.

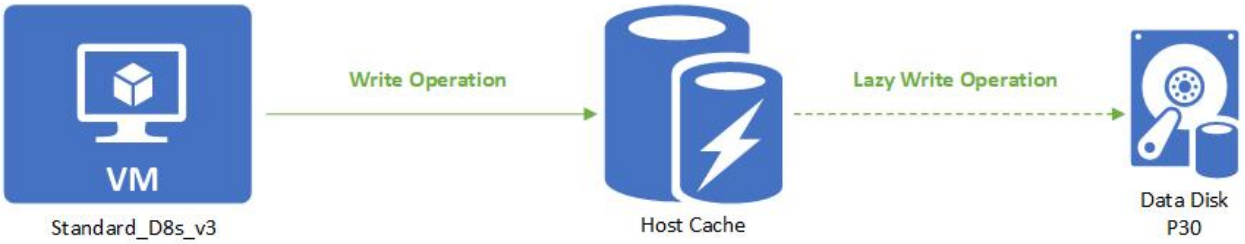


Next let's look at what happens with IO requests when the host cache setting is set to **Read/write**.

Setup:

- Standard_D8s_v3
  - Cached IOPS: 16,000
  - Uncached IOPS: 12,800
- P30 data disk
  - IOPS: 5,000
  - Host caching: **Read/write**

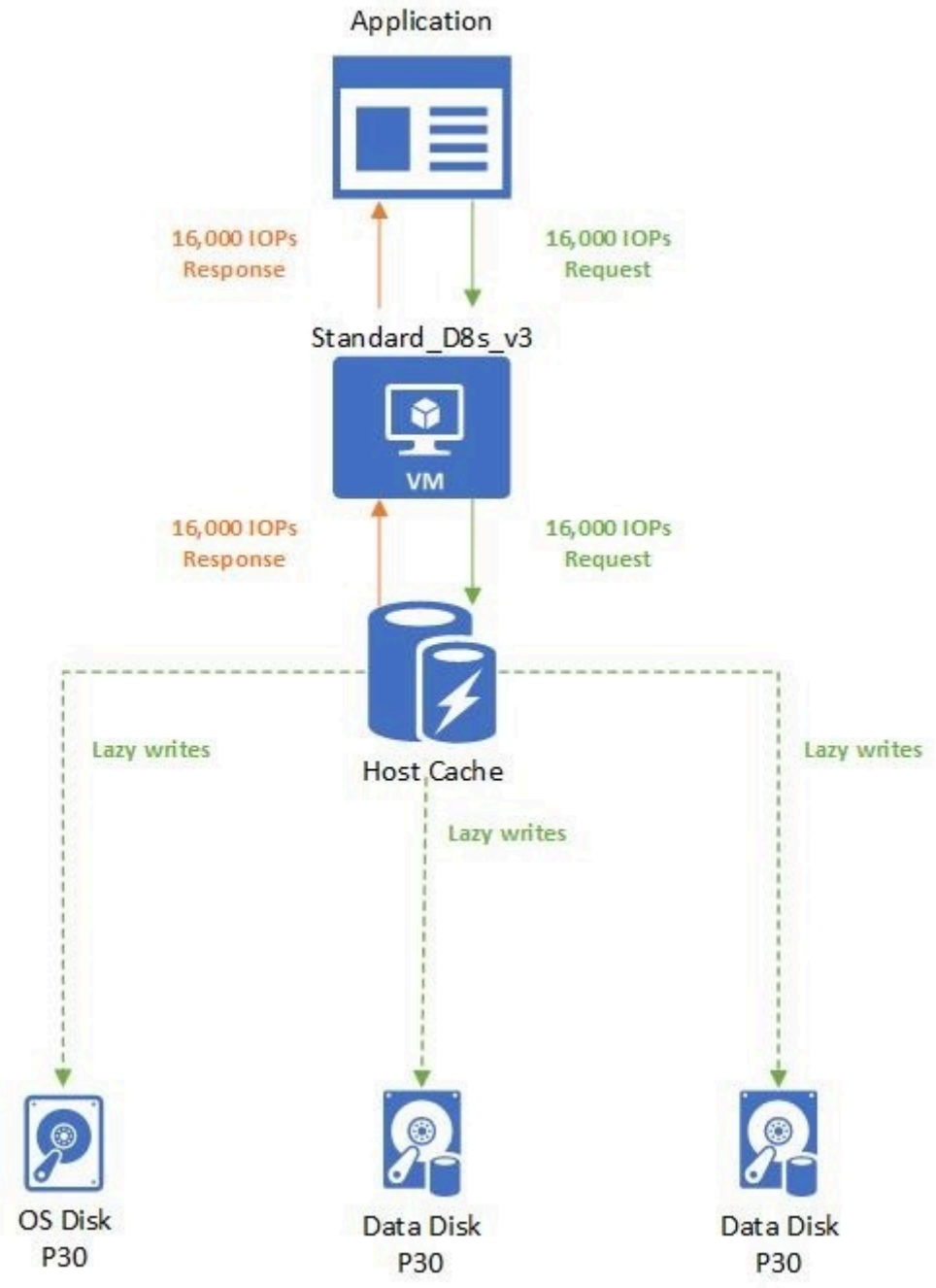
A read is handled the same way as a read-only. Writes are the only thing that's different with read/write caching. When writing with host caching is set to **Read/write**, the write only needs to be written to the host cache to be considered complete. The write is then lazily written to the disk when the cache is flushed periodically. Customers can additionally force a flush by issuing an `f/sync` or `fua` command. This means that a write is counted toward cached IO when it's written to the cache. When it's lazily written to the disk, it counts toward the uncached IO.



Let’s continue with our Standard_D8s_v3 virtual machine. Except this time, we enable host caching on the disks. This makes the VM's IOPS limit 16,000 IOPS. Attached to the VM are three underlying P30 disks that can each handle 5,000 IOPS.

Setup:

- Standard_D8s_v3
  - Cached IOPS: 16,000
  - Uncached IOPS: 12,800
- P30 OS disk
  - IOPS: 5,000
  - Host caching: **Read/write**
- Two P30 data disks × 2
  - IOPS: 5,000
  - Host caching: **Read/write**



The application uses a Standard_D8s_v3 virtual machine with caching enabled. It makes a request for 16,000 IOPS. The requests are completed as soon as they're read or written to the cache. Writes are then lazily written to the attached Disks.

Combined uncached and cached limits

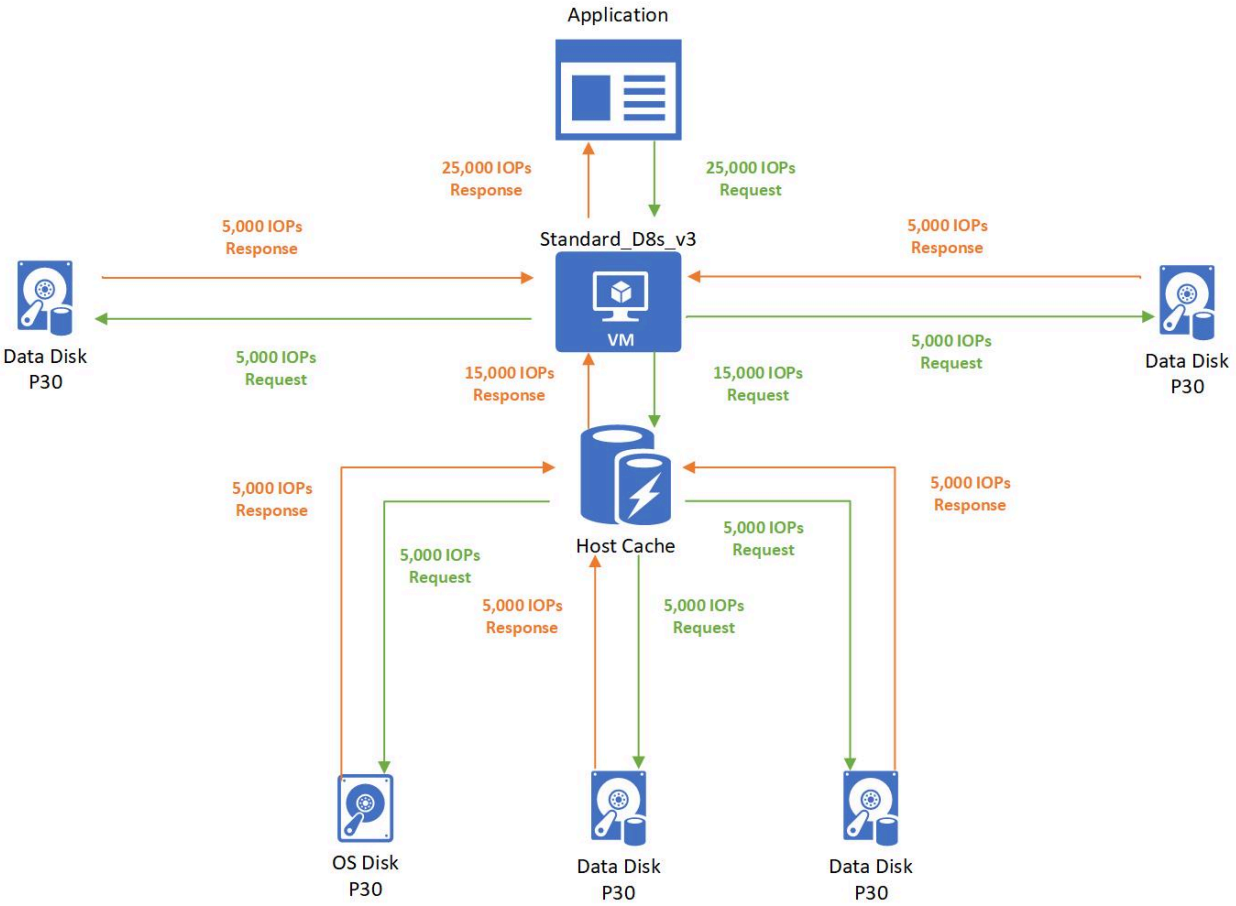
A virtual machine's cached limits are separate from its uncached limits. This means you can enable host caching on disks attached to a VM while not enabling host caching on other disks. This configuration allows your virtual machines to get a total storage IO of the cached limit plus the uncached limit.

Let's run through an example to help you understand how these limits work together. We continue with the Standard_D8s_v3 virtual machine and premium disks attached

configuration.

Setup:

- Standard_D8s_v3
  - Cached IOPS: 16,000
  - Uncached IOPS: 12,800
- P30 OS disk
  - IOPS: 5,000
  - Host caching: **Read/write**
- Two P30 data disks × 2
  - IOPS: 5,000
  - Host caching: **Read/write**
- Two P30 data disks × 2
  - IOPS: 5,000
  - Host caching: **Disabled**



In this case, the application running on a Standard_D8s_v3 virtual machine makes a request for 25,000 IOPS. The request is broken down as 5,000 IOPS to each of the attached disks. Three disks use host caching and two disks don't use host caching.

- Since the three disks that use host caching are within the cached limits of 16,000, those requests are successfully completed. No storage performance capping

occurs.

- Since the two disks that don't use host caching are within the uncached limits of 12,800, those requests are also successfully completed. No capping occurs.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Understand how your reservation discount is applied to Azure disk storage

After you purchase Azure disk reserved capacity, a reservation discount is automatically applied to disk resources that match the terms of your reservation. The reservation discount only applies to managed disks, disk snapshots are charged at pay-as-you-go rates.

For more information about Azure disk reservation, see Save costs with Azure disk reservation. For information about pricing for Azure disk reservation, see Azure Managed Disks pricing.

How the reservation discount is applied

The Azure disk reservation discount is a use-it-or-lose-it discount that applies to managed disk resources hourly. For a given hour, if you have no managed disk resources that meet the reservation terms, you lose a reservation quantity for that hour. Unused hours don't carry forward.

When you delete a resource, the reservation discount automatically applies to another matching resource in the specified scope. If no matching resource is found, the reserved hours are lost.

Stopped resources are billed and continue to use reservation hours. Deallocate or delete resources or scale-in other resources to use your available reservation hours with other workloads.

Discount examples

The following examples show how the Azure disk reservation discount applies depending on your deployment.

Suppose you purchase and reserve 100 P30 disks in the US West 2 region for a one-year term. Each disk has approximately 1 TiB of storage. Assume the cost of this sample reservation is $140,100. You can choose to pay either the full amount up front or fixed monthly installments of $11,675 for the next 12 months.

The following scenarios describe what happens if you underuse, overuse, or tier your reserved capacity. For these examples, assume you signed up for a monthly reservation-payment plan.

Underusing your capacity

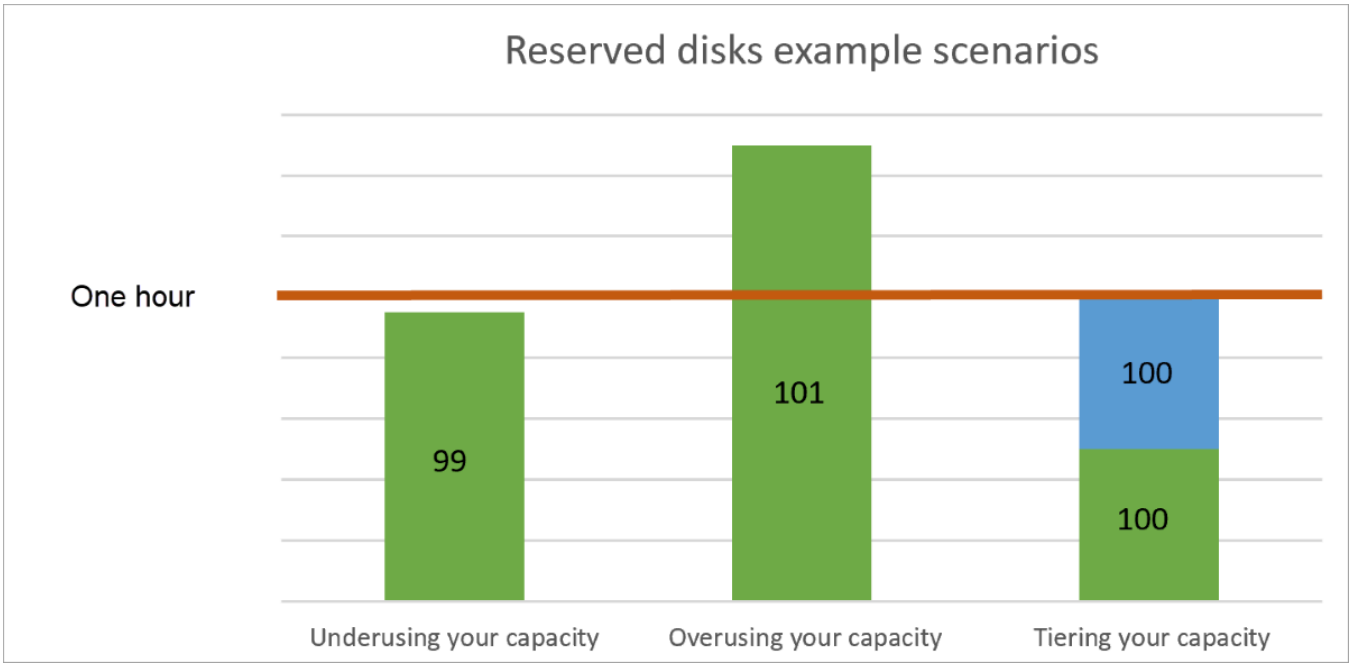
Suppose you deploy only 99 of your 100 reserved Azure Premium SSD P30 disks for an hour within the reservation period. The remaining P30 disk isn't applied during that hour. It also doesn't carry over.

Overusing your capacity

Suppose that for an hour within the reservation period, you use 101 Premium SSD P30 disks. The reservation discount applies only to 100 P30 disks. The remaining P30 disk is charged at pay-as-you-go rates for that hour. For the next hour, if your usage goes down to 100 P30 disks, all usage is covered by the reservation.

Tiering your capacity

Suppose that in a given hour within your reservation period, you want to use a total of 200 P30 Premium SSDs. Also suppose you use only 100 for the first 30 minutes. During this period, your use is fully covered because you made a reservation for 100 P30 disks. If you then discontinue the use of the first 100 (so that you're using zero) and then begin to use the other 100 for the remaining 30 minutes, that usage is also covered under your reservation.



Need help? Contact us

If you have questions or need help, create a support request.

Related content

- Reduce costs with Azure Disks Reservation

- What are Azure Reservations?

Last updated on 03/19/2026

Reduce costs with Azure Disks Reservation

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Save on your Azure Disk Storage usage with reserved capacity. Azure Disk Storage reservations combined with Azure Reserved Virtual Machine Instances let you lower your total virtual machine (VM) costs. The reservation discount is applied automatically to the matching disks in the selected reservation scope. Because of this automatic application, you don't need to assign a reservation to a managed disk to get the discounts.

Discounts are applied hourly depending on the disk usage. Unused reserved capacity doesn't carry over. Azure Disk Storage reservation discounts don't apply to unmanaged disks, ultra disks, or page blob consumption.

Determine your storage needs

Before you purchase a reservation, determine your storage needs. Currently, Azure Disk Storage reservations are available only for select Azure premium SSD SKUs. The SKU of a premium SSD determines the disk's size and performance.

When determining your storage needs, don't think of disks based on just capacity. For example, you can't have a reservation for a P40 disk and use that to pay for two smaller P30 disks. When purchasing a reservation, you're only purchasing a reservation for the total number of disks per SKU.

A disk reservation is made per disk SKU. As a result, the reservation consumption is based on the unit of the disk SKUs instead of the provided size.

For example, assume you reserve one P40 disk that has 2 TiB of provisioned storage capacity. Also assume you allocate only two P30 disks. The P40 reservation in that case doesn't account for P30 consumption, and you pay the pay-as-you-go rate on the P30 disks.

Expand table

Premium SSD sizes	**P1**	**P2**	**P3**	**P4**	**P6**	**P10**	**P15**	**P20**	**P30**	**P40**	**P50**	**P60**	**P70**	**P80**
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
**Expanded provisioned IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8,000	16,000	20,000	20,000	20,000	20,000
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year

*Applies only to disks with on-demand bursting enabled.
** Only applies to disks with performance plus (preview) enabled.

Purchase considerations

We recommend the following practices when considering disk reservation purchase:

- Analyze your usage information to help determine which reservations you should purchase. Make sure you track the usage in disk SKUs instead of provisioned or used disk capacity.
- Examine your disk reservation along with your VM reservation. We highly recommend making reservations for both VM usage and disk usage for maximum savings. You can start with determining the right VM reservation and then evaluate the disk reservation. Generally, you'll have a standard configuration for each of your workloads. For example, a SQL Server server might have two P40 data disks and one P30 operating system disk.

  This kind of pattern can help you determine the reserved amount you might purchase. This approach can simplify the evaluation process and ensure that you have an aligned plan for both your VM and disks. The plan contains considerations like subscriptions or regions.

Purchase restrictions

Reservation discounts are currently unavailable for the following:

- Unmanaged disks or page blobs
- Ultra Disks
- Premium SSD v2 disks
- Standard solid-state drives (SSDs) or standard hard-disk drives (HDDs)
- Premium SSD SKUs smaller than P30: P1, P2, P3, P4, P6, P10, P15, and P20 SSD SKUs
- Disks in Azure Government, Azure Germany, or Microsoft Azure operated by 21Vianet regions

In rare circumstances, Azure limits the purchase of new reservations to a subset of disk SKUs because of low capacity in a region.

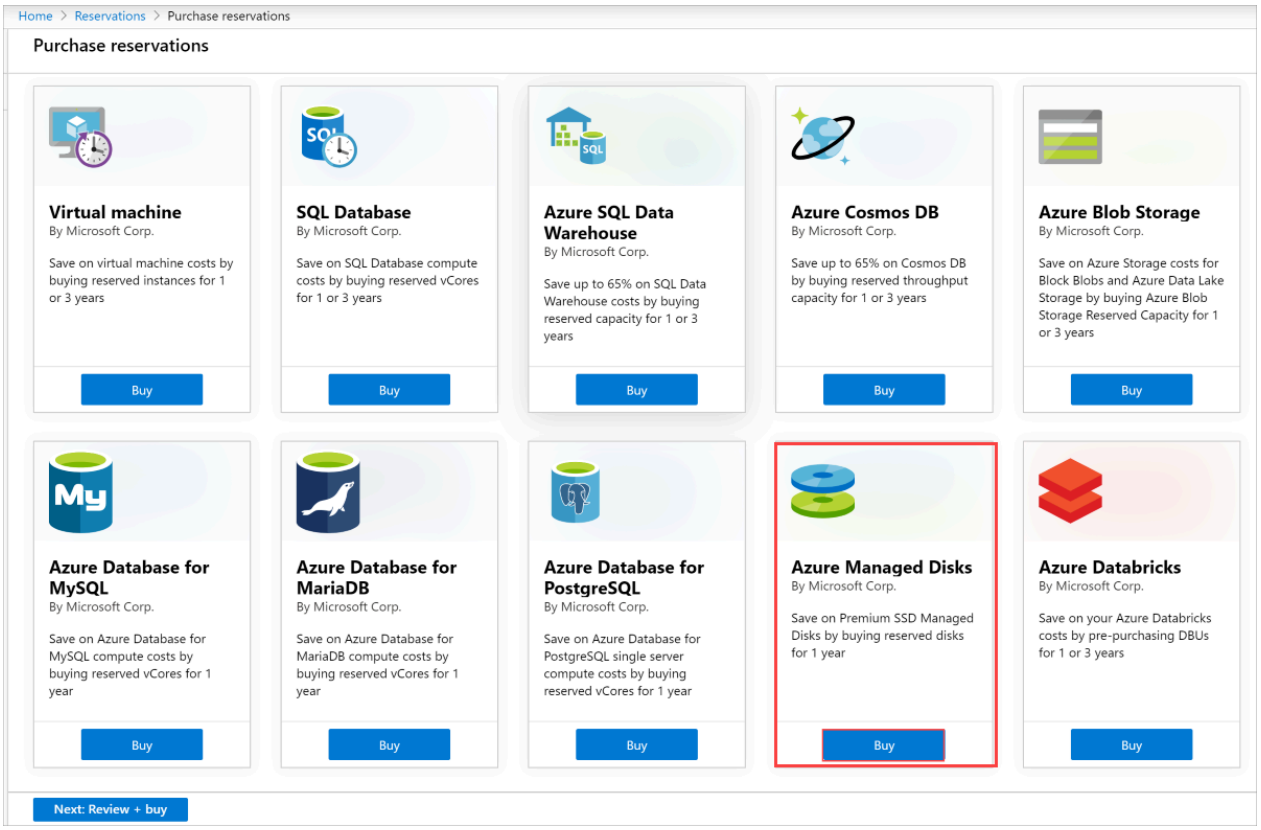
Buy a disk reservation

You can purchase Azure Disk Storage reservations through the Azure portal. You can pay for the reservation either up front or with monthly payments. For more information about purchasing with monthly payments, see Purchase reservations with monthly payments.

To buy a reservation, you must have owner role or reservation purchaser role on an Azure subscription.

Follow these steps to purchase reserved capacity:

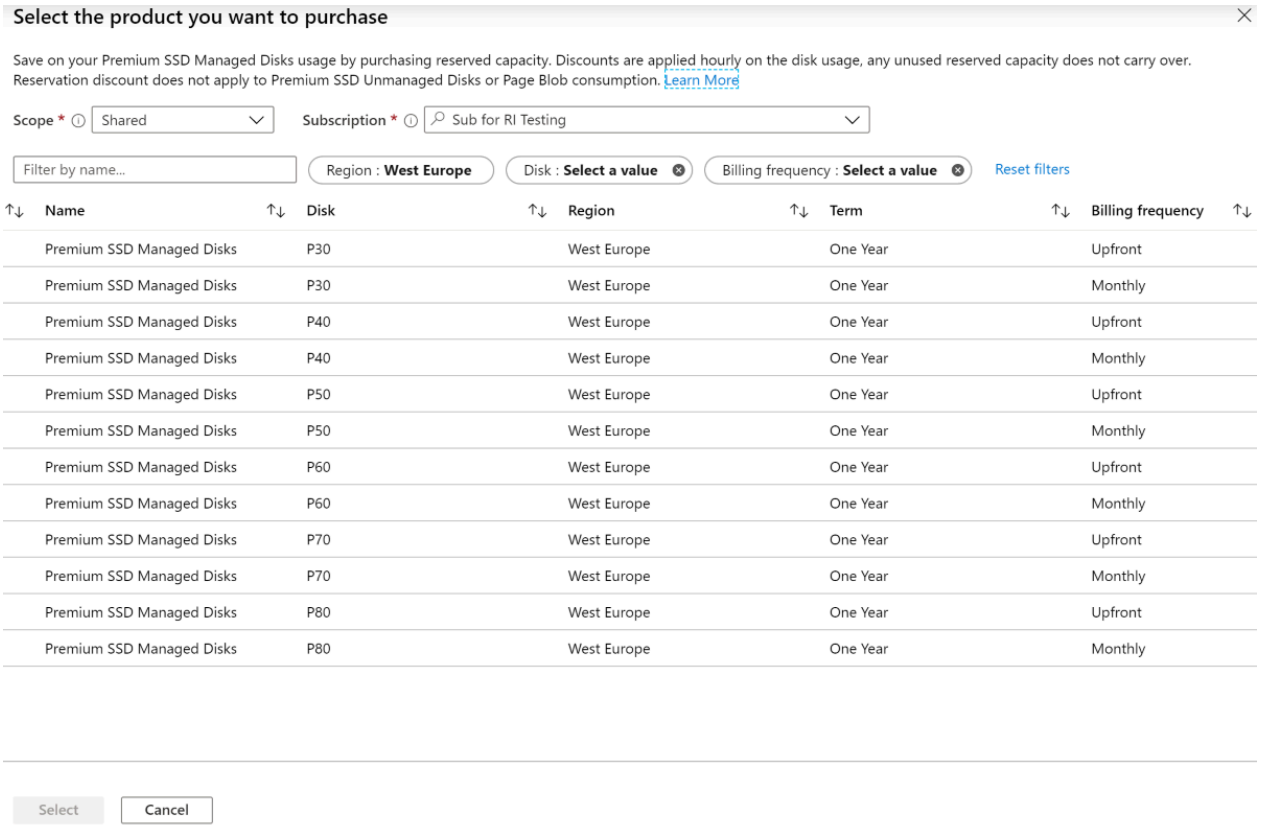
1. Go to the Purchase reservations pane in the Azure portal.
2. Select **Azure Managed Disks** to purchase a reservation.



3. Specify the required values described in the following table:

Expand table

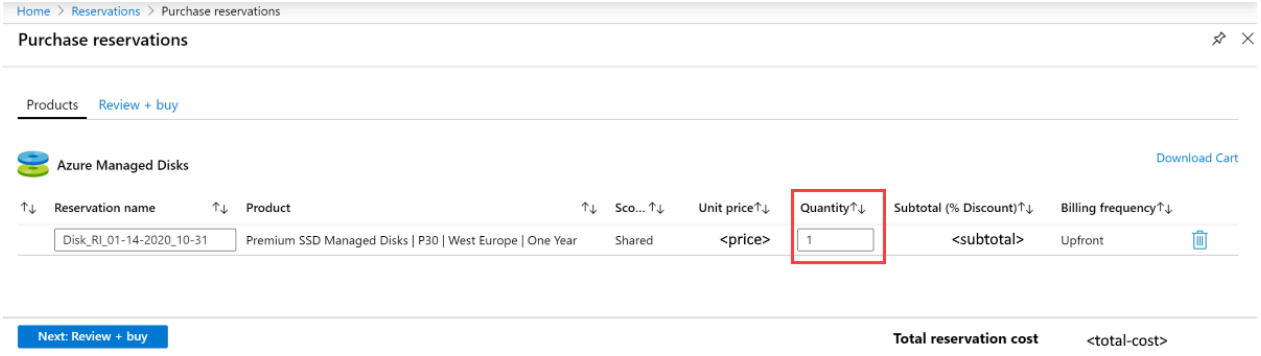
Element	Description
Scope	How many subscriptions can use the billing benefit associated with the reservation. This value also specifies how the reservation is applied to specific subscriptions. If you select **Shared**, the reservation discount is applied to Azure Storage capacity in every subscription within your billing context. The billing context is based on how you signed up for Azure. For enterprise customers, the shared scope is the enrollment and includes all subscriptions within the enrollment. For pay-as-you-go customers, the shared scope includes all individual subscriptions with pay-as-you-go rates created by the account administrator. If you select **Management group**, the reservation discount is applied to Azure Storage capacity in every subscription that is part of the management group and the billing scope. If you select **Single subscription**, the reservation discount is applied to Azure Storage capacity in the selected subscription. If you select **Single resource group**, the reservation discount is applied to Azure Storage capacity in the selected subscription and in that subscription's selected resource group. You can change the reservation scope after you purchase the reservation.
Subscription	The subscription you use to pay for the Azure Storage reservation. The payment method on the selected subscription is used in charging the costs. The subscription must be one of the following types: • Enterprise Agreement (offer numbers MS-AZR-0017P and MS-AZR-0148P). For an Enterprise subscription, the charges are deducted from the enrollment's Azure Prepayment (previously called monetary commitment) balance or charged as overage. • Individual subscription with pay-as-you-go rates (offer numbers MS-AZR-0003P and MS-AZR-0023P). For an individual subscription with pay-as-you-go rates, the charges are billed to the credit card or invoice payment method on the subscription.
Disks	The SKU you want to create.
Region	The region where the reservation is in effect.
Billing frequency	How often the account is billed for the reservation. Options include **Monthly** and **Upfront**.



4. After you specify the values for your reservation, the Azure portal displays the cost. The portal also shows the discount percentage over pay-as-you-go billing. Select **Next** to continue to the **Purchase reservations** pane.

5. On the **Purchase reservations** pane, you can name your reservation and select the total quantity of reservations you want to make. The number of reservations maps to the number of disks. For example, if you want to reserve a hundred disks, enter the **Quantity** value **100**.

6. Review the total cost of the reservation.



After you purchase a reservation, it's automatically applied to any existing Disk Storage resources that match the reservation terms. If you haven't created any Disk Storage resources yet, the reservation applies whenever you create a resource that matches the reservation terms. In either case, the reservation term begins immediately after a successful purchase.

Cancel, exchange, or refund reservations

You can cancel, exchange, or refund reservations within certain limitations. For more information, see Self-service exchanges and refunds for Azure Reservations.

Expiration of a reservation

When a reservation expires, any Azure Disk Storage capacity that you use under that reservation is billed at the pay-as-you-go rate. Reservations don't renew automatically.

You'll receive an email notification 30 days before the expiration of the reservation and again on the expiration date. To continue taking advantage of the cost savings that a reservation provides, renew it no later than the expiration date.

Need help? Contact us

If you have questions or need help, create a support request.

Next steps

- What are Azure Reservations?
- Understand how your reservation discount is applied to Azure Disk Storage

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure premium storage: Design for high performance

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article provides guidelines for building high-performance applications by using Azure premium storage. You can use the instructions provided in this document combined with performance best practices applicable to technologies used by your application. To illustrate the guidelines, we use SQL Server running on premium storage as an example throughout this document.

While we address performance scenarios for the storage layer in this article, you need to optimize the application layer. For example, if you're hosting a SharePoint Farm on premium storage, you can use the SQL Server examples from this article to optimize the database server. You can also optimize the SharePoint Farm's web server and application server to get the most performance.

This article helps to answer the following common questions about optimizing application performance on premium storage:

- How can you measure your application performance?
- Why aren't you seeing expected high performance?
- Which factors influence your application performance on premium storage?
- How do these factors influence performance of your application on premium storage?
- How can you optimize for input/output operations per second (IOPS), bandwidth, and latency?

We provide these guidelines specifically for premium storage because workloads running on premium storage are highly performance sensitive. We provide examples where appropriate. You can also apply some of these guidelines to applications running on infrastructure as a service (IaaS) VMs with standard storage disks.

> **Note**
>
> Sometimes what appears to be a disk performance issue is actually a network bottleneck. In these situations, you should optimize your network performance.
>
> If you're looking to benchmark your disk, see the following articles:
>
> - For Linux: Benchmark your application on Azure Disk Storage
> - For Windows: Benchmark a disk
>
> If your VM supports accelerated networking, make sure it's enabled. If it's not enabled, you can enable it on already deployed VMs on both Windows and Linux.

Before you begin, if you're new to premium storage, first read Select an Azure disk type for IaaS VMs and Scalability targets for premium page blob storage accounts.

Application performance indicators

We assess whether an application is performing well or not by using performance indicators like:

- How fast an application is processing a user request.
- How much data an application is processing per request.
- How many requests an application is processing in a specific period of time.
- How long a user has to wait to get a response after submitting their request.

The technical terms for these performance indicators are IOPS, throughput or bandwidth, and latency.

In this section, we discuss the common performance indicators in the context of premium storage. In the section Performance application checklist for disks, you learn how to measure these performance indicators for your application. Later in Optimize application performance, you learn about the factors that affect these performance indicators and recommendations to optimize them.

IOPS

IOPS is the number of requests that your application is sending to storage disks in one second. An input/output operation could be read or write, sequential, or random. Online transaction processing (OLTP) applications like an online retail website need to process many concurrent user requests immediately. The user requests are insert- and update-intensive database transactions, which the application must process quickly. For this reason, OLTP applications require very high IOPS.

OLTP applications handle millions of small and random I/O requests. If you have such an application, you must design the application infrastructure to optimize for IOPS. For more information on all the factors to consider to get high IOPS, see Optimize application performance.

When you attach a premium storage disk to your high-scale VM, Azure provisions for you a guaranteed number of IOPS according to the disk specification. For example, a P50 disk provisions 7,500 IOPS. Each high-scale VM size also has a specific IOPS limit that it can sustain. For example, a Standard GS5 VM has an 80,000 IOPS limit.

Throughput

Throughput, or bandwidth, is the amount of data that your application is sending to the storage disks in a specified interval. If your application is performing input/output operations with large I/O unit sizes, it requires high throughput. Data warehouse applications tend to issue scan-intensive operations that access large portions of data at a time and commonly perform bulk operations. In other words, such applications require higher throughput. If you have such an application, you must design its infrastructure to optimize for throughput. In the next section, we discuss the factors you must tune to achieve this optimization.

When you attach a premium storage disk to a high-scale VM, Azure provisions throughput according to that disk specification. For example, a P50 disk provisions 250 MB/sec disk throughput. Each high-scale VM size also has a specific throughput limit that it can sustain. For example, Standard GS5 VM has a maximum throughput of 2,000 MB/sec.

There's a relation between throughput and IOPS, as shown in the following formula.



It's important to determine the optimal throughput and IOPS values that your application requires. As you try to optimize one, the other is also affected. For more information about optimizing IOPS and throughput, see Optimize application performance.

Latency

Latency is the time it takes an application to receive a single request, send it to storage disks, and send the response to the client. Latency is a critical measure of an application's performance in addition to IOPS and throughput. The latency of a premium storage disk is the time it takes to retrieve the information for a request and communicate it back to your application. Premium storage provides consistently low latencies. Premium disks are designed to provide single-digit millisecond latencies for most I/O operations. If you enable **ReadOnly** host caching on premium storage disks, you can get much lower read latency. For more information on disk caching, see Disk caching.

When you optimize your application to get higher IOPS and throughput, it affects the latency of your application. After you tune the application performance, evaluate the latency of the application to avoid unexpected high latency behavior.

Some control plane operations on managed disks might move the disk from one storage location to another. Moving the disk between storage locations is orchestrated via a background copy of data, which can take several hours to complete. Typically, the time is less than 24 hours depending on the amount of data in the disks. During that time, your application can experience higher than usual read latency because some reads can get redirected to the original location and take longer to complete.

During a background copy, there's no effect on write latency for most disk types. For Premium SSD v2 and Ultra Disks, if the disk has a 4K sector size, it experiences higher read latency. If the disk has a 512e sector size, it experiences higher read and write latency.

The following control plane operations might move the disk between storage locations, and cause increased latency:

- Update the storage type.

- Detach and attach a disk from one VM to another.
- Create a managed disk from a VHD.
- Create a managed disk from a snapshot.
- Convert unmanaged disks to managed disks.

Performance application checklist for disks

The first step in designing high-performance applications running on premium storage is understanding the performance requirements of your application. After you gather performance requirements, you can optimize your application to achieve the most optimal performance.

In the previous section, we explained the common performance indicators: IOPS, throughput, and latency. You must identify which of these performance indicators are critical to your application to deliver the desired user experience. For example, high IOPS matters most to OLTP applications processing millions of transactions in a second. High throughput is critical for data warehouse applications processing large amounts of data in a second. Extremely low latency is crucial for real-time applications like live video-streaming websites.

Next, measure the maximum performance requirements of your application throughout its lifetime. Use the following sample checklist as a start. Record the maximum performance requirements during normal, peak, and off-hour workload periods. By identifying requirements for all workload levels, you can determine the overall performance requirement of your application.

For example, the normal workload of an e-commerce website is the transactions it serves during most days in a year. The peak workload of the website is the transactions it serves during holiday seasons or special sale events. The peak workload is typically experienced for a limited period but can require your application to scale two or more times its normal operation. Find out the 50 percentile, 90 percentile, and 99 percentile requirements. This information helps filter out any outliers in the performance requirements, and you can focus your efforts on optimizing for the right values.

Application performance requirements checklist

Expand table

Performance requirements	50 percentile	90 percentile	99 percentile
Maximum transactions per second			
% Read operations			
% Write operations			
% Random operations			
% Sequential operations			
I/O request size			
Average throughput			
Maximum throughput			
Minimum latency			
Average latency			
Maximum CPU			
Average CPU			
Maximum memory			
Average memory			
Queue depth			

Note

Consider scaling these numbers based on expected future growth of your application. It's a good idea to plan for growth ahead of time because it could be harder to change the infrastructure for improving performance later.

If you have an existing application and want to move to premium storage, first build the preceding checklist for the existing application. Then, build a prototype of your application on premium storage and design the application based on guidelines described in Optimize application performance. The next article describes the tools you can use to gather the performance measurements.

Counters to measure application performance requirements

The best way to measure performance requirements of your application is to use `PerfMon`-monitoring tools provided by the operating system of the server. You can use `PerfMon` for Windows and `iostat` for Linux. These tools capture counters corresponding to each measure explained in the preceding section. You must capture the values of these counters when your application is running its normal, peak, and off-hour workloads.

The `PerfMon` counters are available for processor, memory, and each logical disk and physical disk of your server. When you use premium storage disks with a VM, the physical disk counters are for each premium storage disk, and logical disk counters are for each volume created on the premium storage disks. You must capture the values for the disks that host your application workload. If there's a one-to-one mapping between logical and physical disks, you can refer to physical disk counters. Otherwise, refer to the logical disk counters.

On Linux, the `iostat` command generates a CPU and disk utilization report. The disk utilization report provides statistics per physical device or partition. If you have a database server with its data and logs on separate disks, collect this data for both disks. The following table describes counters for disks, processors, and memory.

Expand table

Counter	Description	PerfMon	iostat
IOPS or transactions/sec	Number of I/O requests issued to the storage disk/sec	Disk reads/sec Disk writes/sec	tps r/s w/s
Disk reads and writes	% of read and write operations performed on the disk	% Disk read time % Disk write time	r/s w/s
Throughput	Amount of data read from or written to the disk/sec	Disk read bytes/sec Disk write bytes/sec	kB_read/s kB_wrtn/s
Latency	Total time to complete a disk I/O request	Average disk sec/read Average disk sec/write	await svctm
I/O size	The size of I/O request issues to the storage disks	Average disk bytes/read Average disk bytes/write	avgrq-sz
Queue depth	Number of outstanding I/O requests waiting to be read from or written to the storage disk	Current disk queue length	avgqu-sz
Maximum memory	Amount of memory required to run the application smoothly	% Committed bytes in use	Use vmstat
Maximum CPU	Amount of CPU required to run the application smoothly	% Processor time	%util

Learn more about iostat and PerfMon.

Optimize application performance

The main factors that influence performance of an application running on premium storage are the nature of I/O requests, VM size, disk size, number of disks, disk caching, multithreading, and queue depth. You can control some of these factors with knobs provided by the system.

Most applications might not give you an option to alter the I/O size and queue depth directly. For example, if you're using SQL Server, you can't choose the I/O size and queue depth. SQL Server chooses the optimal I/O size and queue depth values to get the

most performance. It's important to understand the effects of both types of factors on your application performance so that you can provision appropriate resources to meet performance needs.

Throughout this section, refer to the application requirements checklist that you created to identify how much you need to optimize your application performance. Based on the checklist, you can determine which factors from this section you need to tune.

To witness the effects of each factor on your application performance, run benchmarking tools on your application setup. For steps to run common benchmarking tools on Windows and Linux VMs, see the benchmarking articles at the end of this document.

Optimize IOPS, throughput, and latency at a glance

The following table summarizes performance factors and the steps necessary to optimize IOPS, throughput, and latency. The sections following this summary describe each factor in more depth.

For more information on VM sizes and on the IOPS, throughput, and latency available for each type of VM, see Sizes for virtual machines in Azure.

Expand table

Performance factors	IOPS	Throughput	Latency
Example scenario	Enterprise OLTP application requiring very high transactions per second rate.	Enterprise Data warehousing application processing large amounts of data.	Near real-time applications requiring instant responses to user requests, like online gaming.
Performance factors			
I/O size	Smaller I/O size yields higher IOPS.	Larger I/O size yields higher throughput.	
VM size	Use a VM size that offers IOPS greater than your application requirement.	Use a VM size with a throughput limit greater than your application requirement.	Use a VM size that offers scale limits greater than your application requirement.
Disk size	Use a disk size that offers IOPS greater than your application requirement.	Use a disk size with a throughput limit greater than your application requirement.	Use a disk size that offers scale limits greater than your application requirement.
VM and disk scale limits	IOPS limit of the VM size chosen should be greater than the total IOPS driven by the storage disks attached to it.	Throughput limit of the VM size chosen should be greater than the total throughput driven by the premium storage disks attached to it.	Scale limits of the VM size chosen must be greater than the total scale limits of the attached premium storage disks.
Disk caching	Enable **ReadOnly** cache on premium storage disks with read-heavy operations to get higher read IOPS.		Enable **ReadOnly** cache on premium storage disks with read-heavy operations to get very low read latencies.
Disk striping	Use multiple disks and stripe them together to get a combined higher IOPS and throughput limit. The combined limit per VM should be higher than the combined limits of attached premium disks.		
Stripe size	Smaller stripe size for random small I/O pattern seen in OLTP applications. For example, use a 64-KB stripe size for a SQL Server OLTP application.	Larger stripe size for sequential large I/O pattern seen in data warehouse applications. For example, use a 256-KB stripe size for a SQL Server data warehouse application.	
Multithreading	Use multithreading to push a higher number of requests to premium storage to lead to higher IOPS and throughput. For example, on SQL Server, set a high MAXDOP value to allocate more CPUs to SQL Server.		
Queue depth	Larger queue depth yields higher IOPS.	Larger queue depth yields higher	Smaller queue depth yields

Performance factors	IOPS	Throughput	Latency
		throughput.	lower latencies.

Nature of I/O requests

An I/O request is a unit of input/output operation that your application is performing. Identifying the nature of I/O requests, random or sequential, read or write, small or large, helps you determine the performance requirements of your application. It's important to understand the nature of I/O requests to make the right decisions when you design your application infrastructure. I/Os must be distributed evenly to achieve the best performance possible.

I/O size is one of the more important factors. The I/O size is the size of the input/output operation request generated by your application. The I/O size affects performance significantly, especially on the IOPS and bandwidth that the application can achieve. The following formula shows the relationship between IOPS, I/O size, and bandwidth/throughput.



Some applications allow you to alter their I/O size, while some applications don't. For example, SQL Server determines the optimal I/O size itself and doesn't provide users with any knobs to change it. On the other hand, Oracle provides a parameter called DB_BLOCK_SIZE, which you can use to configure the I/O request size of the database.

If you're using an application, which doesn't allow you to change the I/O size, use the guidelines in this article to optimize the performance KPI that's most relevant to your application. For example:

- An OLTP application generates millions of small and random I/O requests. To handle these types of I/O requests, you must design your application infrastructure to get higher IOPS.
- A data warehousing application generates large and sequential I/O requests. To handle these types of I/O requests, you must design your application infrastructure to get higher bandwidth or throughput.

If you're using an application that allows you to change the I/O size, use this rule of thumb for the I/O size in addition to other performance guidelines:

- Smaller I/O size to get higher IOPS. For example, 8 KB for an OLTP application.
- Larger I/O size to get higher bandwidth/throughput. For example, 1,024 KB for a data warehouse application.

Here's an example of how you can calculate the IOPS and throughput/bandwidth for your application.

Consider an application that uses a P30 disk. The maximum IOPS and throughput/bandwidth a P30 disk can achieve is 5,000 IOPS and 200 MB/sec, respectively. If your application requires the maximum IOPS from the P30 disk and you use a smaller I/O size, like 8 KB, the resulting bandwidth you can get is 40 MB/sec. If your application requires the maximum throughput/bandwidth from a P30 disk and you use a larger I/O size, like 1,024 KB, the resulting IOPS is less, such as 200 IOPS.

Tune the I/O size so that it meets both your application's IOPS and throughput/bandwidth requirement. The following table summarizes the different I/O sizes and their corresponding IOPS and throughput for a P30 disk.

Expand table

Application requirement	I/O size	IOPS	Throughput/Bandwidth
Maximum IOPS	8 KB	5,000	40 MB/sec
Maximum throughput	1,024 KB	200	200 MB/sec
Maximum throughput + high IOPS	64 KB	3,200	200 MB/sec
Maximum IOPS + high throughput	32 KB	5,000	160 MB/sec

To get IOPS and bandwidth higher than the maximum value of a single premium storage disk, use multiple premium disks striped together. For example, stripe two P30 disks to get a combined IOPS of 10,000 IOPS or a combined throughput of 400 MB/sec. As explained in the next section, you must use a VM size that supports the combined disk IOPS and throughput.

> **Note**
>
> As you increase either IOPS or throughput, the other also increases. Make sure you don't hit throughput or IOPS limits of the disk or VM when you increase either one.

To witness the effects of I/O size on application performance, you can run benchmarking tools on your VM and disks. Create multiple test runs and use different I/O size for each run to see the effect. For more information, see the benchmarking articles at the end of this document.

High-scale VM sizes

When you start designing an application, one of the first things to do is choose a VM to host your application. Premium storage comes with high-scale VM sizes that can run applications requiring higher compute power and a high local disk I/O performance. These VMs provide faster processors, a higher memory-to-core ratio, and a solid-state drive (SSD) for the local disk. Examples of high-scale VMs supporting premium storage are the DS and GS series VMs.

High-scale VMs are available in different sizes with a different number of CPU cores, memory, OS, and temporary disk size. Each VM size also has a maximum number of data disks that you can attach to the VM. The chosen VM size affects how much processing, memory, and storage capacity are available for your application. It also affects the compute and storage cost. For example, the following specifications are for the largest VM size in a DS series and a GS series.

Expand table

VM size	CPU cores	Memory	VM disk sizes	Maximum data disks	Cache size	IOPS	Bandwidth cache I/O limits
Standard_DS14	16	112 GB	OS = 1,023 GB Local SSD = 224 GB	32	576 GB	50,000 IOPS 512 MB/sec	4,000 IOPS and 33 MB/sec
Standard_GS5	32	448 GB	OS = 1,023 GB Local SSD = 896 GB	64	4224 GB	80,000 IOPS 2,000 MB/sec	5,000 IOPS and 50 MB/sec

To view a complete list of all available Azure VM sizes, see Sizes for virtual machines in Azure. Choose a VM size that can meet and scale to your desired application performance requirements. Also take into account the following important considerations when you choose VM sizes.

Scale limits

The maximum IOPS limits per VM and per disk are different and independent of each other. Make sure that the application is driving IOPS within the limits of the VM and the premium disks attached to it. Otherwise, application performance experiences throttling.

As an example, suppose an application requirement is a maximum of 4,000 IOPS. To achieve this level, you provision a P30 disk on a DS1 VM. The P30 disk can deliver up to 5,000 IOPS. However, the DS1 VM is limited to 3,200 IOPS. So, the application performance is constrained by the VM limit at 3,200 IOPS and performance is degraded. To prevent this situation, choose a VM and disk size that both meet application requirements.

Cost of operation

In many cases, it's possible that your overall cost of operation using premium storage is lower than using standard storage.

For example, consider an application requiring 16,000 IOPS. To achieve this performance, you need a Standard_D14 Azure IaaS VM, which can give a maximum IOPS of 16,000 by using 32 standard storage 1-TB disks. Each 1-TB standard storage disk can achieve a maximum of 500 IOPS.

- The estimated cost of this VM per month is $1,570.
- The monthly cost of 32 standard storage disks is $1,638.
- The estimated total monthly cost is $3,208.

If you hosted the same application on premium storage, you need a smaller VM size and fewer premium storage disks, reducing the overall cost. A Standard_DS13 VM can meet the 16,000 IOPS requirement by using four P30 disks. The DS13 VM has a maximum IOPS of 25,600, and each P30 disk has a maximum IOPS of 5,000. Overall, this configuration can achieve 5,000 x 4 = 20,000 IOPS.

- The estimated cost of this VM per month is $1,003.
- The monthly cost of four P30 premium storage disks is $544.34.
- The estimated total monthly cost is $1,544.

The following table summarizes the cost breakdown of this scenario for standard and premium storage.

Expand table

Monthly cost	Standard	Premium
Cost of VM per month	$1,570.58 (Standard_D14)	$1,003.66 (Standard_DS13)
Cost of disks per month	$1,638.40 (32 x 1-TB disks)	$544.34 (4 x P30 disks)
Overall cost per month	$3,208.98	$1,544.34

Linux distros

With premium storage, you get the same level of performance for VMs running Windows and Linux. We support many flavors of Linux distros. For more information, see Linux distributions endorsed on Azure.

Different distros are better suited for different types of workloads. You see different levels of performance depending on the distro on which your workload is running. Test the Linux distros with your application and choose the one that works best.

When you run Linux with premium storage, check the latest updates about required drivers to ensure high performance.

Premium storage disk sizes

Premium storage offers various sizes so you can choose one that best suits your needs. Each disk size has a different scale limit for IOPS, bandwidth, and storage. Choose the right premium storage disk size depending on the application requirements and the high-scale VM size. The following table shows the disks sizes and their capabilities. P4, P6, P15, P60, P70, and P80 sizes are currently only supported for managed disks.

Expand table

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
**Expanded provisioned IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8,000	16,000	20,000	20,000	20,000	20,000
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year

*Applies only to disks with on-demand bursting enabled.
** Only applies to disks with performance plus (preview) enabled.

How many disks you choose depends on the disk size chosen. You could use a single P50 disk or multiple P10 disks to meet your application requirement. Take into account considerations listed here when you're making the choice.

Scale limits (IOPS and throughput)

The IOPS and throughput limits of each premium disk size is different and independent from the VM scale limits. Make sure that the total IOPS and throughput from the disks are within scale limits of the chosen VM size.

For example, if an application requirement is a maximum of 250 MB/sec throughput and you're using a DS4 VM with a single P30 disk, the DS4 VM can give up to 256 MB/sec throughput. However, a single P30 disk has a throughput limit of 200 MB/sec. So, the application is constrained at 200 MB/sec because of the disk limit. To overcome this limit, provision more than one data disk to the VM or resize your disks to P40 or P50.

> ⓘ **Note**
>
> Reads served by the cache aren't included in the disk IOPS and throughput, so they aren't subject to disk limits. Cache has its separate IOPS and throughput limit per VM.
>
> For example, initially your reads and writes are 60 MB/sec and 40 MB/sec, respectively. Over time, the cache warms up and serves more and more of the reads from the cache. Then, you can get higher write throughput from the disk.

Number of disks

Determine the number of disks you need by assessing application requirements. Each VM size also has a limit on the number of disks that you can attach to the VM. Typically, this amount is twice the number of cores. Ensure that the VM size you choose can support the number of disks needed.

Remember, the premium storage disks have higher performance capabilities compared to standard storage disks. If you're migrating your application from an Azure IaaS VM using standard storage to premium storage, you likely need fewer premium disks to achieve the same or higher performance for your application.

Disk caching

High-scale VMs that use premium storage have a multitier caching technology called **BlobCache**. **BlobCache** uses a combination of the host RAM and local SSD for caching. This cache is available for the premium storage persistent disks and the VM local disks. By default, this cache setting is set to **ReadWrite** for OS disks and **ReadOnly** for data disks hosted on premium storage. With disk caching enabled on the premium storage disks, the high-scale VMs can achieve extremely high levels of performance that exceed the underlying disk performance.

> ⚠ **Warning**

Disk caching isn't supported for disks 4 TiB and larger. If multiple disks are attached to your VM, each disk that's smaller than 4 TiB supports caching.

Changing the cache setting of an Azure disk detaches and reattaches the target disk. If it's the operating system disk, the VM is restarted. Stop all applications and services that might be affected by this disruption before you change the disk cache setting. Not following those recommendations could lead to data corruption.

To learn more about how **BlobCache** works, see the Inside Azure premium storage blog post.

It's important to enable caching on the right set of disks. Whether you should enable disk caching on a premium disk or not depends on the workload pattern that disk is handling. The following table shows the default cache settings for OS and data disks.

Expand table

Disk type	Default cache setting
OS disk	ReadWrite
Data disk	ReadOnly

We recommend the following disk cache settings for data disks.

Expand table

Disk caching setting	Recommendation for when to use this setting
None	Configure host-cache as **None** for write-only and write-heavy disks.
ReadOnly	Configure host-cache as **ReadOnly** for read-only and read-write disks.
ReadWrite	Configure host-cache as **ReadWrite** only if your application properly handles writing cached data to persistent disks when needed.

ReadOnly

By configuring **ReadOnly** caching on premium storage data disks, you can achieve low read latency and get very high read IOPS and throughput for your application for two reasons:

1. Reads performed from cache, which is on the VM memory and local SSD, are faster than reads from the data disk, which is on Azure Blob Storage.
2. Premium storage doesn't count the reads served from the cache toward the disk IOPS and throughput. For this reason, your application can achieve higher total IOPS and throughput.

ReadWrite

By default, the OS disks have **ReadWrite** caching enabled. We recently added support for **ReadWrite** caching on data disks too. If you're using **ReadWrite** caching, you must have a proper way to write the data from cache to persistent disks. For example, SQL Server handles writing cached data to the persistent storage disks on its own. Using **ReadWrite** cache with an application that doesn't handle persisting the required data can lead to data loss, if the VM crashes.

None

Currently, **None** is only supported on data disks. It isn't supported on OS disks. If you set **None** on an OS disk, it overrides this setting internally and sets it to **ReadOnly**.

As an example, you can apply these guidelines to SQL Server running on premium storage by following these steps:

1. Configure the **ReadOnly** cache on premium storage disks hosting data files.
    a. The fast reads from cache lower the SQL Server query time because data pages are retrieved faster from the cache compared to directly from the data disks.

   b. Serving reads from cache means there's more throughput available from premium data disks. SQL Server can use this extra throughput toward retrieving more data pages and other operations like backup/restore, batch loads, and index rebuilds.
2. Configure the **None** cache on premium storage disks hosting the log files.
   a. Log files have primarily write-heavy operations, so they don't benefit from the **ReadOnly** cache.

Optimize performance on Linux VMs

For all Premium SSDs or Ultra Disks, you might be able to disable *barriers* for file systems on the disk to improve performance when it's known that there are no caches that could lose data. If Azure disk caching is set to **ReadOnly** or **None**, you can disable barriers. But if caching is set to **ReadWrite**, barriers should remain enabled to ensure write durability. Barriers are typically enabled by default, but you can disable barriers by using one of the following methods depending on the file system type:

- **reiserFS**: Use the **barrier=none** mount option to disable barriers. To explicitly enable barriers, use **barrier=flush**.
- **ext3/ext4**: Use the **barrier=0** mount option to disable barriers. To explicitly enable barriers, use **barrier=1**.
- **XFS**: Use the **nobarrier** mount option to disable barriers. To explicitly enable barriers, use **barrier**. As of version 4.10 of the mainline Linux kernel, the design of the XFS file system always ensures durability. Disabling barriers has no effect and the **nobarrier** option is deprecated. However, some Linux distributions might have backported the changes to a distribution release with an earlier kernel version. Check with your distribution vendor for the status in the distribution and version that you're running.

Disk striping

When a high-scale VM is attached with several premium storage persistent disks, the disks can be striped together to aggregate their IOPs, bandwidth, and storage capacity.

On Windows, you can use Storage Spaces to stripe disks together. You must configure one column for each disk in a pool. Otherwise, the overall performance of striped volume can be lower than expected because of uneven distribution of traffic across the disks.

By using the Server Manager UI, you can set the total number of columns up to `8` for a striped volume. When you're attaching more than eight disks, use PowerShell to create the volume. By using PowerShell, you can set the number of columns equal to the number of disks. For example, if there are 16 disks in a single stripe set, specify `16` columns in the `NumberOfColumns` parameter of the `New-VirtualDisk` PowerShell cmdlet.

On Linux, use the MDADM utility to stripe disks together. For steps on how to stripe disks on Linux, see Configure Software RAID on Linux.

Stripe size

An important configuration in disk striping is the stripe size. The stripe size or block size is the smallest chunk of data that an application can address on a striped volume. The stripe size you configure depends on the type of application and its request pattern. If you choose the wrong stripe size, it could lead to I/O misalignment, which leads to degraded performance of your application.

For example, if an I/O request generated by your application is bigger than the disk stripe size, the storage system writes it across stripe unit boundaries on more than one disk. When it's time to access that data, it has to seek across more than one stripe unit to complete the request. The cumulative effect of such behavior can lead to substantial performance degradation. On the other hand, if the I/O request size is smaller than the stripe size, and if it's random in nature, the I/O requests might add up on the same disk, causing a bottleneck and ultimately degrading the I/O performance.

Depending on the type of workload your application is running, choose an appropriate stripe size. For random small I/O requests, use a smaller stripe size. For large sequential I/O requests, use a larger stripe size. Find out the stripe size recommendations for the application you'll be running on premium storage. For SQL Server, configure a stripe size of 64 KB for OLTP workloads and 256 KB for data warehousing workloads. For more information, see Performance best practices for SQL Server on Azure VMs.

> ⓘ **Note**

You can stripe together a maximum of 32 premium storage disks on a DS series VM and 64 premium storage disks on a GS series VM.

Multithreading

Azure designed the premium storage platform to be massively parallel. For this reason, a multithreaded application achieves higher performance than a single-threaded application. A multithreaded application splits up its tasks across multiple threads and increases efficiency of its execution by utilizing the VM and disk resources to the maximum.

For example, if your application is running on a single core VM using two threads, the CPU can switch between the two threads to achieve efficiency. While one thread is waiting on a disk I/O to complete, the CPU can switch to the other thread. In this way, two threads can accomplish more than a single thread would. If the VM has more than one core, it further decreases running time because each core can run tasks in parallel.

You might not be able to change the way an off-the-shelf application implements single threading or multithreading. For example, SQL Server is capable of handling multi-CPU and multicore. However, SQL Server decides under what conditions it uses one or more threads to process a query. It can run queries and build indexes by using multithreading. For a query that involves joining large tables and sorting data before returning to the user, SQL Server likely uses multiple threads. A user can't control whether SQL Server runs a query by using a single thread or multiple threads.

There are configuration settings that you can alter to influence the multithreading or parallel processing of an application. For example, for SQL Server it's the `max degree of parallelism` configuration. This setting called MAXDOP allows you to configure the maximum number of processors SQL Server can use when parallel processing. You can configure MAXDOP for individual queries or index operations. This capability is beneficial when you want to balance resources of your system for a performance critical application.

For example, say your application that's using SQL Server is running a large query and an index operation at the same time. Let's assume that you wanted the index operation to be more performant compared to the large query. In such a case, you can set the MAXDOP value of the index operation to be higher than the MAXDOP value for the query. This way, SQL Server has more processors than it can use for the index operation compared to the number of processors it can dedicate to the large query. Remember, you don't control the number of threads that SQL Server uses for each operation. You can control the maximum number of processors being dedicated for multithreading.

Learn more about degrees of parallelism in SQL Server. Find out how such settings influence multithreading in your application and their configurations to optimize performance.

Queue depth

The queue depth or queue length or queue size is the number of pending I/O requests in the system. The value of queue depth determines how many I/O operations your application can line up, which the storage disks process. It affects all three application performance indicators discussed in this article: IOPS, throughput, and latency.

Queue depth and multithreading are closely related. The queue depth value indicates how much multithreading can be achieved by the application. If the queue depth is large, the application can run more operations concurrently, in other words, more multithreading. If the queue depth is small, even though the application is multithreaded, it won't have enough requests lined up for concurrent execution.

Typically, off-the-shelf applications don't allow you to change queue depth, because if it's set incorrectly, it does more harm than good. Applications set the right value of queue depth to get the optimal performance. It's important to understand this concept so that you can troubleshoot performance issues with your application. You can also observe the effects of queue depth by running benchmarking tools on your system.

Some applications provide settings to influence the queue depth. For example, the MAXDOP setting in SQL Server explained in the previous section. MAXDOP is a way to influence queue depth and multithreading, although it doesn't directly change the queue depth value of SQL Server.

High queue depth

A high queue depth lines up more operations on the disk. The disk knows the next request in its queue ahead of time. So, the disk can schedule operations ahead of time and process them in an optimal sequence. Because the application is sending more requests to the disk, the disk can process more parallel I/Os. Ultimately, the application can achieve higher IOPS. Because the application is processing more requests, the total throughput of the application also increases.

Typically, an application can achieve maximum throughput with 8 to 16+ outstanding I/Os per attached disk. If a queue depth is one, the application isn't pushing enough I/Os to the system, and it processes a smaller amount in a given period. In other words, less throughput.

For example, in SQL Server, setting the MAXDOP value for a query to `4` informs SQL Server that it can use up to four cores to run the query. SQL Server determines the best queue depth value and the number of cores for the query execution.

Optimal queue depth

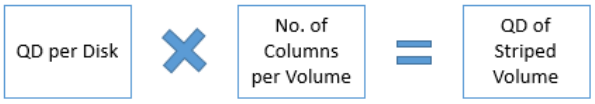
A very high queue depth value also has its drawbacks. If the queue depth value is too high, the application tries to drive very high IOPS. Unless the application has persistent disks with sufficient provisioned IOPS, a very high queue depth value can negatively affect application latencies. The following formula shows the relationship between IOPS, latency, and queue depth.



You shouldn't configure queue depth to any high value, but to an optimal value, which can deliver enough IOPS for the application without affecting latencies. For example, if the application latency needs to be 1 millisecond, the queue depth required to achieve 5,000 IOPS is QD = 5,000 x 0.001 = 5.

Queue depth for striped volume

For a striped volume, maintain a high-enough queue depth so that every disk has a peak queue depth individually. For example, consider an application that pushes a queue depth of `2` and there are four disks in the stripe. The two I/O requests go to two disks and the remaining two disks are idle. Therefore, configure the queue depth so that all the disks can be busy. The following formula shows how to determine the queue depth of striped volumes.



Throttling

Premium storage provisions a specified number of IOPS and throughput depending on the VM sizes and disk sizes you choose. Anytime your application tries to drive IOPS or throughput above these limits of what the VM or disk can handle, premium storage throttles it. The result is degraded performance in your application, which can mean higher latency, lower throughput, or lower IOPS.

If premium storage doesn't throttle, your application could completely fail by exceeding what its resources are capable of achieving. To avoid performance issues because of throttling, always provision sufficient resources for your application. Take into consideration what we discussed in the previous VM sizes and disk sizes sections. Benchmarking is the best way to figure out what resources you need to host your application.

Next steps

If you're looking to benchmark your disk, see the following articles:

- For Linux: Benchmark your application on Azure Disk Storage
- For Windows: Benchmark a disk

Learn more about the available disk types:

- For Linux: Select a disk type

- For Windows: Select a disk type

For SQL Server users, see the articles on performance best practices for SQL Server:

- Performance best practices for SQL Server in Azure VMs
- Azure premium storage provides highest performance for SQL Server in Azure VM

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Disk performance metrics

Article • 08/29/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure offers metrics in the Azure portal that provide insight on how your virtual machines (VM) and disks perform. The metrics can also be retrieved through an API call. This article is broken into 3 subsections:

- **Disk IO, throughput, queue depth and latency metrics** - These metrics allow you to see the storage performance from the perspective of a disk and a virtual machine.
- **Disk bursting metrics** - These are the metrics provide observability into our bursting feature on our premium disks.
- **Storage IO utilization metrics** - These metrics help diagnose bottlenecks in your storage performance with disks.

All metrics are emitted every minute, except for the bursting credit percentage metric, which is emitted every 5 minutes.

Disk IO, throughput, queue depth and latency metrics

The following metrics are available to get insight on VM and disk IO, throughput, and queue depth performance:

- **OS Disk Latency (Preview)**: The average time to complete IOs during the monitoring for the OS disk. Metric is only available for disks attached to VMs using SCSI disk controller and not with disk attached to VMs using the NVMe disk controller. Values are in milliseconds.
- **OS Disk Queue Depth**: The number of current outstanding IO requests that are waiting to be read from or written to the OS disk.
- **OS Disk Read Bytes/Sec**: The number of bytes that are read in a second from the OS disk. If Read-only or Read/write disk caching is enabled, this metric is inclusive of bytes read from the cache.
- **OS Disk Read Operations/Sec**: The number of input operations that are read in a second from the OS disk. If Read-only or Read/write disk caching is enabled, this metric is inclusive of IOPs read from the cache.

- **OS Disk Write Bytes/Sec**: The number of bytes that are written in a second from the OS disk.
- **OS Disk Write Operations/Sec**: The number of output operations that are written in a second from the OS disk.
- **Data Disk Latency (Preview)**: The average time to complete IOs during the monitoring for the data disk. Metric is only available for disks attached to VMs using SCSI disk controller and not with disk attached to VMs using the NVMe disk controller. Values are in milliseconds.
- **Data Disk Queue Depth**: The number of current outstanding IO requests that are waiting to be read from or written to the data disk(s).
- **Data Disk Read Bytes/Sec**: The number of bytes that are read in a second from the data disk(s). If Read-only or Read/write disk caching is enabled, this metric is inclusive of bytes read from the cache.
- **Data Disk Read Operations/Sec**: The number of input operations that are read in a second from data disk(s). If Read-only or Read/write disk caching is enabled, this metric is inclusive of IOPs read from the cache.
- **Data Disk Write Bytes/Sec**: The number of bytes that are written in a second from the data disk(s).
- **Data Disk Write Operations/Sec**: The number of output operations that are written in a second from data disk(s).
- **Disk Read Bytes**: The number of total bytes that are read in a minute from all disks attached to a VM. If Read-only or Read/write disk caching is enabled, this metric is inclusive of bytes read from the cache.
- **Disk Read Operations/Sec**: The number of input operations that are read in a second from all disks attached to a VM. If Read-only or Read/write disk caching is enabled, this metric is inclusive of IOPs read from the cache.
- **Disk Write Bytes**: The number of bytes that are written in a minute from all disks attached to a VM.
- **Disk Write Operations/Sec**: The number of output operations that are written in a second from all disks attached to a VM.
- **Temp Disk Latency (Preview)**: The average time to complete IOs during the monitoring for the temporary disk. This metric is not available for NVMe temporary storage disks. Values are in milliseconds.
- **Temp Disk Queue Depth**: The number of current outstanding IO requests that are waiting to be read from or written to the temporary disk. This metric is not available for NVMe temporary storage disks.
- **Temp Disk Read Bytes/Sec**: The number of bytes that are read in a second from the temporary disk. This metric is not available for NVMe temporary storage disks.
- **Temp Disk Read Operations/Sec**: The number of input operations that are read in a second from the temporary disk. This metric is not available for NVMe temporary storage disks.

- **Temp Disk Write Bytes/Sec**: The number of bytes that are written in a second from the temporary disk. This metric is not available for NVMe temporary storage disks.
- **Temp Disk Write Operations/Sec**: The number of output operations that are written in a second from the temporary disk. This metric is not available for NVMe temporary storage disks.

> ⓘ **Note**
>
> Disk metrics can't log CRUD (Create, Read, Update, Delete) operations inside managed disks.

Bursting metrics

The following metrics help with observability into our bursting feature on our premium disks:

- **Data Disk Max Burst Bandwidth**: The throughput limit that the data disk(s) can burst up to.
- **OS Disk Max Burst Bandwidth**: The throughput limit that the OS disk can burst up to.
- **Data Disk Max Burst IOPS**: the IOPS limit that the data disk(s) can burst up to.
- **OS Disk Max Burst IOPS**: The IOPS limit that the OS disk can burst up to.
- **Data Disk Target Bandwidth**: The throughput limit that the data(s) disk can achieve without bursting.
- **OS Disk Target Bandwidth**: The throughput limit that the OS disk can achieve without bursting.
- **Data Disk Target IOPS**: The IOPS limit that the data disk(s) can achieve without bursting.
- **OS Disk Target IOPS**: The IOPS limit that the data disk(s) can achieve without bursting.
- **Data Disk Used Burst BPS Credits Percentage**: The accumulated percentage of the throughput burst used for the data disk(s). Emitted on a 5 minute interval.
- **OS Disk Used Burst BPS Credits Percentage**: The accumulated percentage of the throughput burst used for the OS disk. Emitted on a 5 minute interval.
- **Data Disk Used Burst IO Credits Percentage**: The accumulated percentage of the IOPS burst used for the data disk(s). Emitted on a 5 minute interval.
- **OS Disk Used Burst IO Credits Percentage**: The accumulated percentage of the IOPS burst used for the OS disk. Emitted on a 5 minute interval.
- **Disk On-demand Burst Operations**: The accumulated operations of burst transactions used for disks with on-demand bursting enabled. Emitted on an hour

interval.

VM Bursting metrics

The following metrics provide insight on VM-level bursting:

- **VM Uncached Used Burst IO Credits Percentage**: The accumulated percentage of the VM's uncached IOPS burst used. Emitted on a 5 minute interval.
- **VM Uncached Used Burst BPS Credits Percentage**: The accumulated percentage of the VM's uncached throughput burst used. Emitted on a 5 minute interval.
- **VM Cached Used Burst IO Credits Percentage**: The accumulated percentage of the VM's cached IOPS burst used. Emitted on a 5 minute interval.
- **VM Cached Used Burst BPS Credits Percentage**: The accumulated percentage of the VM's cached throughput burst used. Emitted on a 5 minute interval.

Storage IO utilization metrics

The following metrics help diagnose bottleneck in your Virtual Machine and Disk combination. These metrics are only available on VM series that support premium storage.

Metrics that help diagnose disk IO capping:

- **Data Disk IOPS Consumed Percentage**: The percentage calculated by dividing the actual data disk IOPS completed by the provisioned data disk IOPS. If this amount is at 100%, your application running is IO capped from your data disk's IOPS limit.
- **Data Disk Bandwidth Consumed Percentage**: The percentage calculated by dividing the actual data disk throughput completed by the provisioned data disk throughput. If this amount is at 100%, your application running is IO capped from your data disk's bandwidth limit.
- **OS Disk IOPS Consumed Percentage**: The percentage calculated by dividing the actual OS disk IOPS completed by the provisioned OS disk IOPS. If this amount is at 100%, your application running is IO capped from your OS disk's IOPS limit.
- **OS Disk Bandwidth Consumed Percentage**: The percentage calculated by dividing the actual OS disk throughput completed by the provisioned OS disk throughput. If this amount is at 100%, your application running is IO capped from your OS disk's bandwidth limit.

Metrics that help diagnose VM IO capping:

- **VM Cached IOPS Consumed Percentage**: The percentage calculated by dividing the total actual cached IOPS completed by the max cached virtual machine IOPS

limit. If this amount is at 100%, your application running is IO capped from your VM's cached IOPS limit.

- **VM Cached Bandwidth Consumed Percentage**: The percentage calculated by dividing the total actual cached throughput completed by the max cached virtual machine throughput. If this amount is at 100%, your application running is IO capped from your VM's cached bandwidth limit.
- **VM uncached IOPS Consumed Percentage**: The percentage calculated by dividing the total actual uncached IOPS on a virtual machine completed by the max uncached virtual machine IOPS limit. If this amount is at 100%, your application running is IO capped from your VM's uncached IOPS limit.
- **VM Uncached Bandwidth Consumed Percentage**: The percentage calculated by dividing the total actual uncached throughput on a virtual machine completed over the max provisioned virtual machine throughput. If this amount is at 100%, your application running is IO capped from your VM's uncached bandwidth limit.

Storage IO metrics example

Let's run through an example of how to use these new Storage IO utilization metrics to help us debug where a bottleneck is in our system. The system setup is the same as the previous example, except this time the attached OS disk is *not* cached.

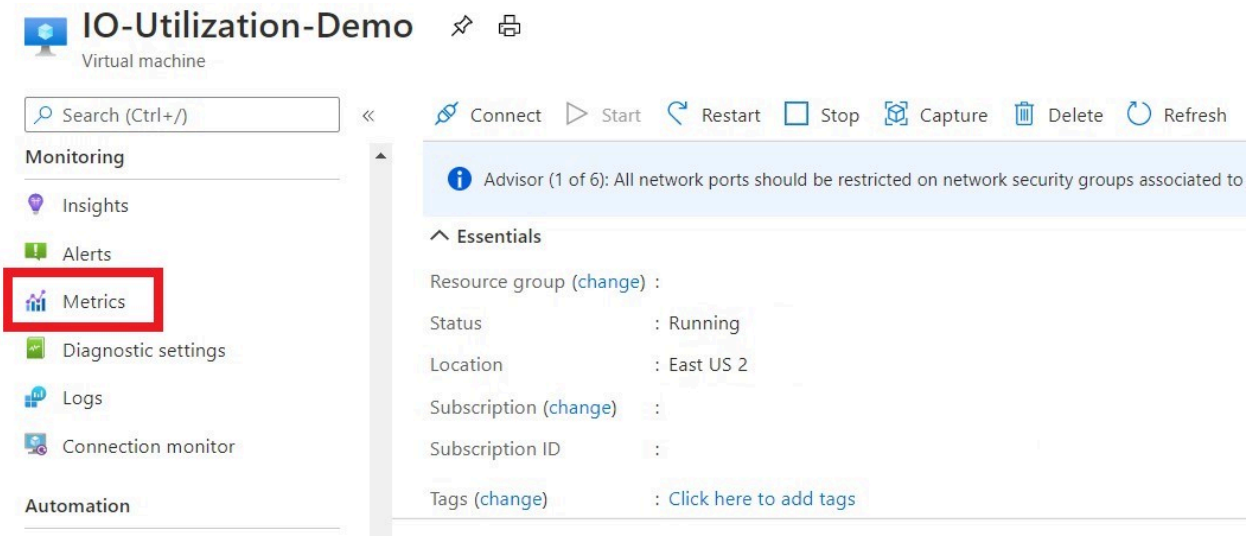
Setup:

- Standard_D8s_v3
  - Cached IOPS: 16,000
  - Uncached IOPS: 12,800
- P30 OS disk
  - IOPS: 5,000
  - Host caching: **Disabled**
- Two P30 data disks × 2
  - IOPS: 5,000
  - Host caching: **Read/write**
- Two P30 data disks × 2
  - IOPS: 5,000
  - Host caching: **Disabled**

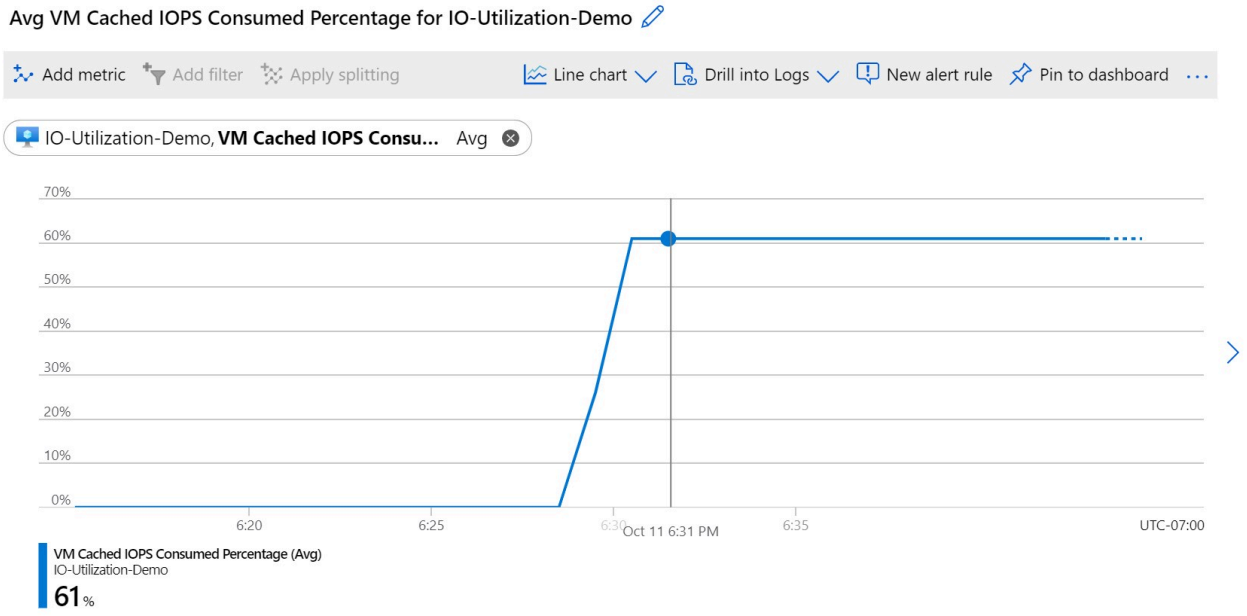
Let's run a benchmarking test on this virtual machine and disk combination that creates IO activity. To learn how to benchmark storage IO on Azure, see Benchmark your application on Azure Disk Storage. From the benchmarking tool, you can see that the VM and disk combination can achieve 22,800 IOPS:

```
demouser@IO-Utilization-Demo:~$ sudo fio fioread.ini
reader1: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=libaio, iodepth=256
reader3: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=libaio, iodepth=256
reader5: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=libaio, iodepth=256
reader10: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=libaio, iodepth=256
reader12: (g=0): rw=randread, bs=(R) 8192B-8192B, (W) 8192B-8192B, (T) 8192B-8192B, ioengine=libaio, iodepth=256
fio-3.1
Starting 5 processes
Jobs: 5 (f=5), 0-22800 IOPS: [r(5)][0.4%][r=178MiB/s,w=0KiB/s] [r=22.8k,w=0 IOPS][eta 27m:06s]
```

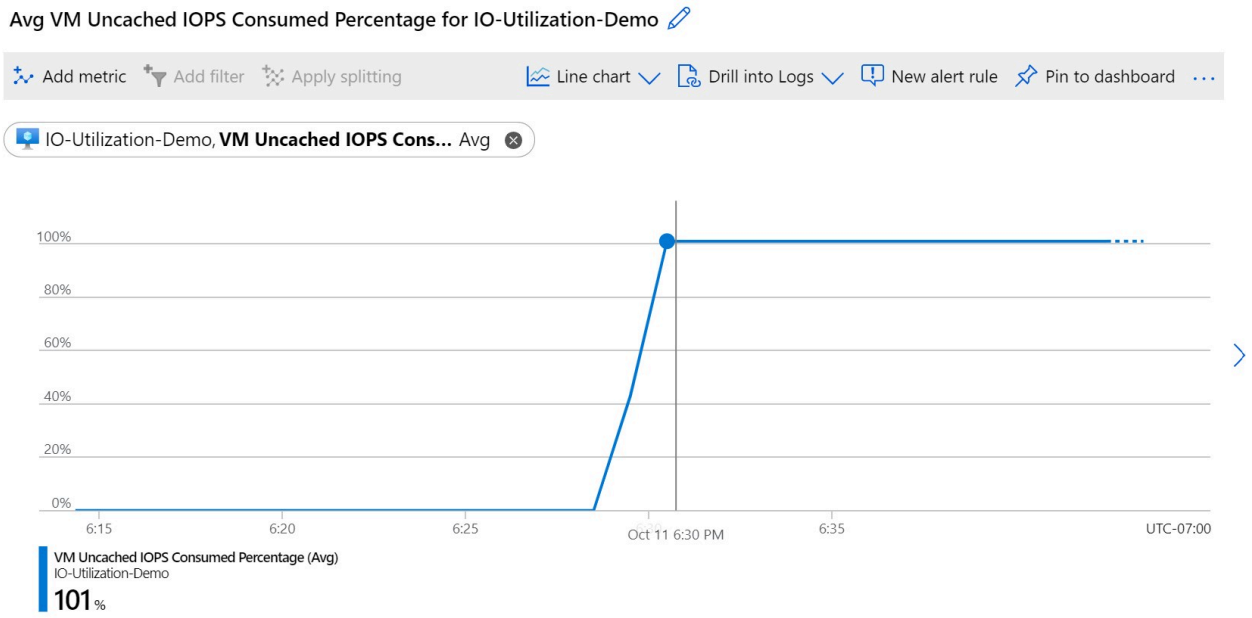
The Standard_D8s_v3 can achieve a total of 28,600 IOPS. Using the metrics, let's investigate what's going on and identify our storage IO bottleneck. On the left pane, select **Metrics**:



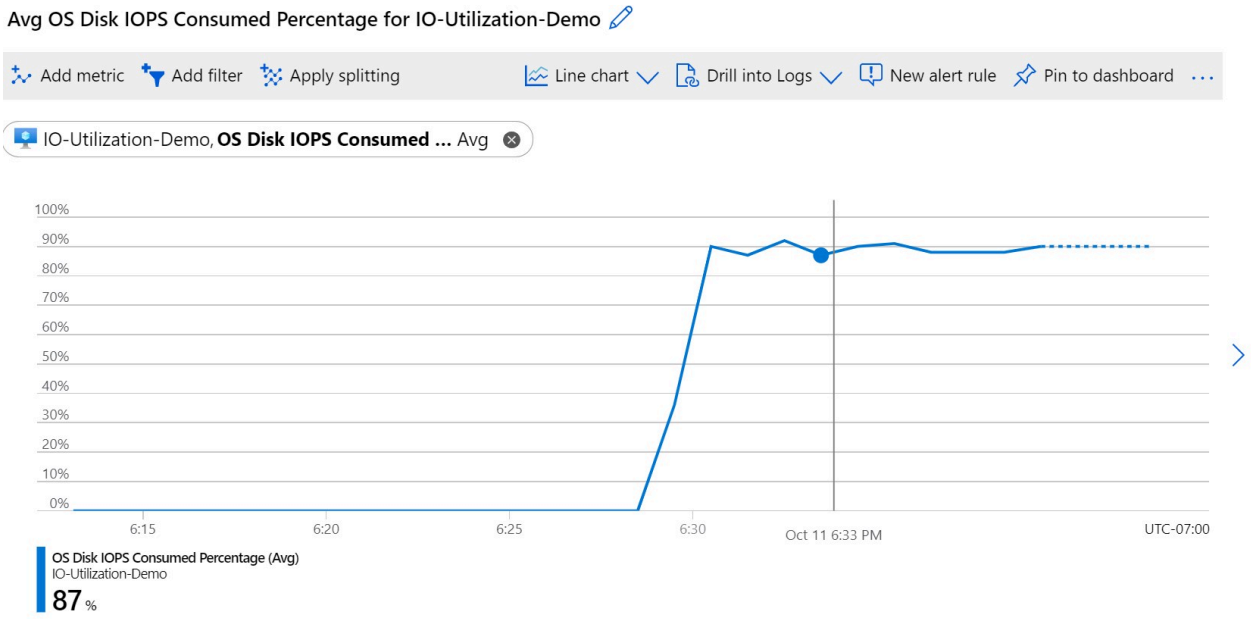
Let's first take a look at our **VM Cached IOPS Consumed Percentage** metric:



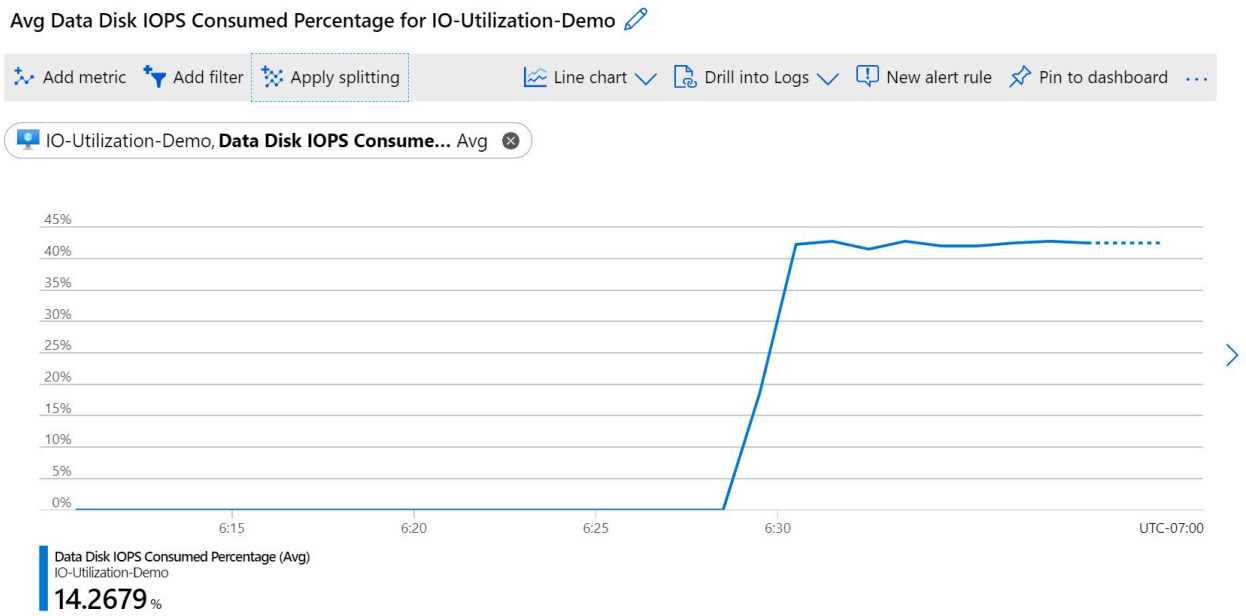
This metric tells us that 61% of the 16,000 IOPS allotted to the cached IOPS on the VM is being used. This percentage means that the storage IO bottleneck isn't with the disks that are cached because it isn't at 100%. Now let's look at our **VM Uncached IOPS Consumed Percentage** metric:



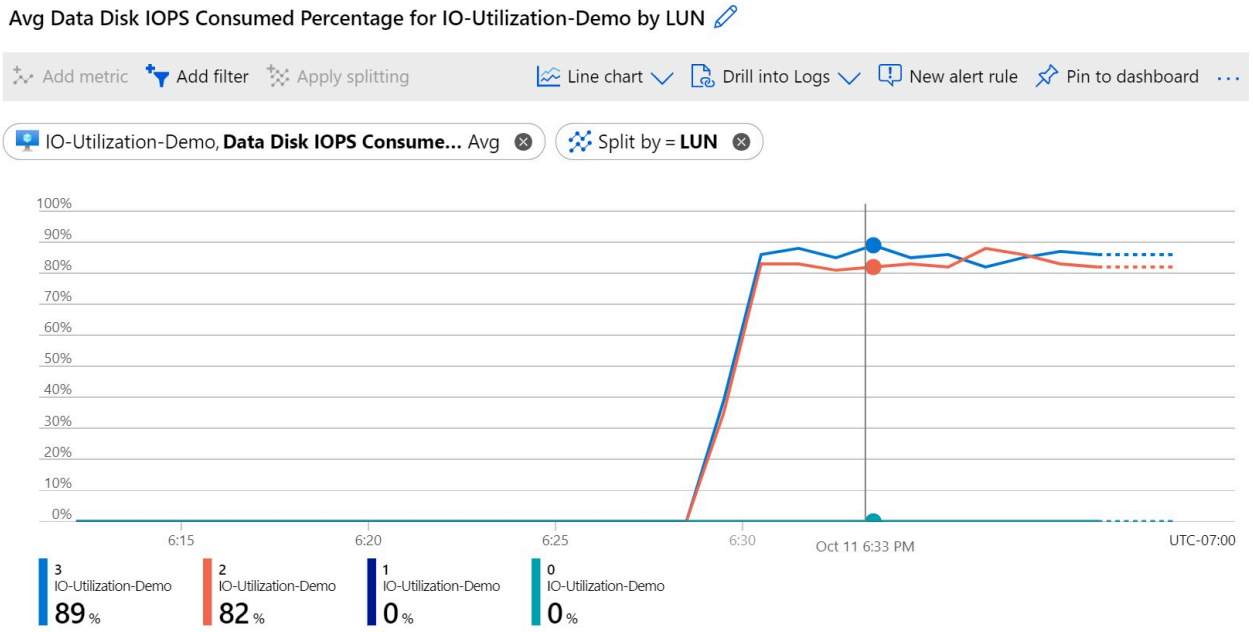
This metric is at 100%. It tells us that all of the 12,800 IOPS allotted to the uncached IOPS on the VM are being used. One way we can remediate this issue is to change the size of our VM to a larger size that can handle the additional IO. But before we do that, let's look at the attached disk to find out how many IOPS they are seeing. Check the OS Disk by looking at the **OS Disk IOPS Consumed Percentage**:



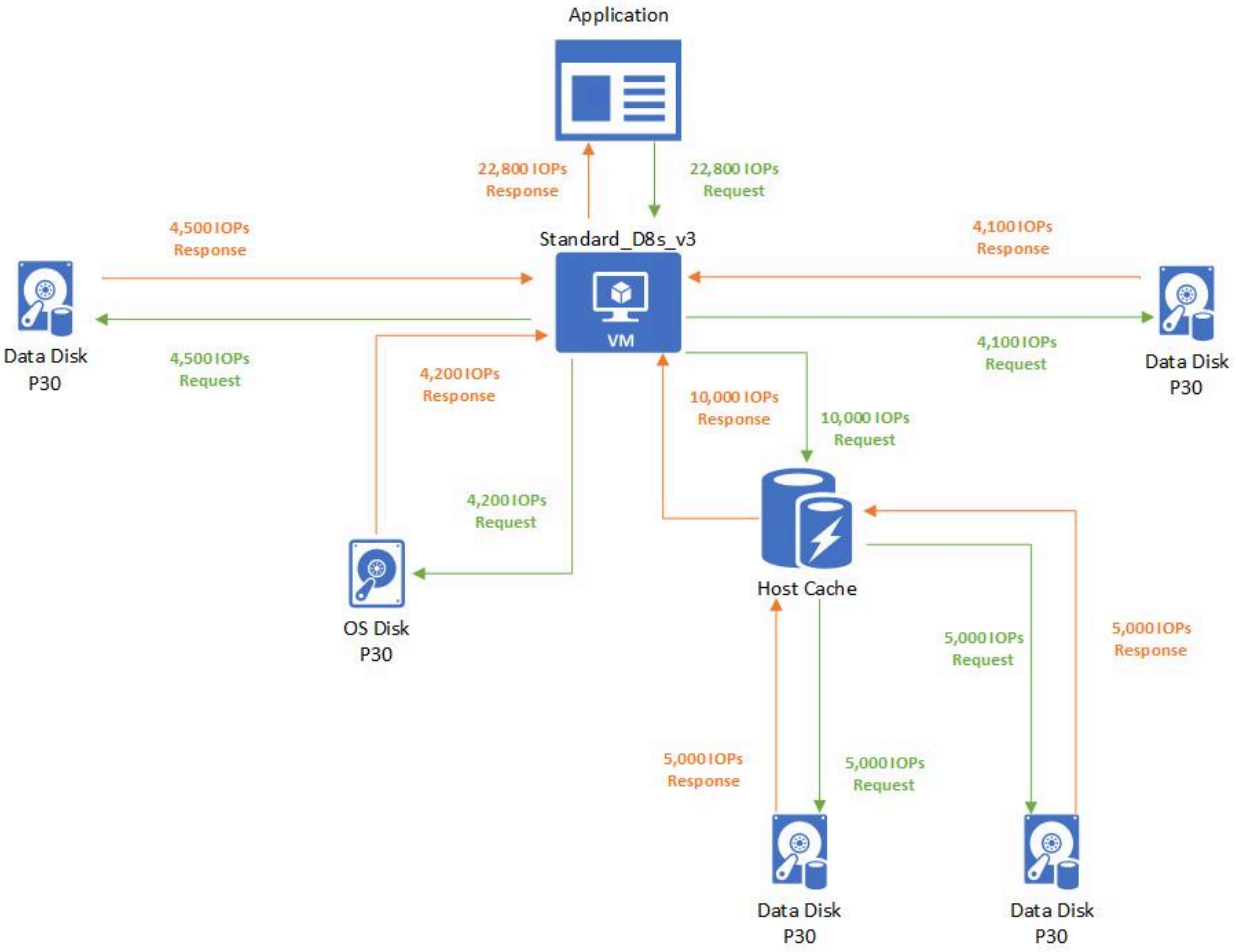
This metric tells us that around 90% of the 5,000 IOPS provisioned for this P30 OS disk is being used. This percentage means there's no bottleneck at the OS disk. Now let's check the data disks that are attached to the VM by looking at the **Data Disk IOPS Consumed Percentage**:



This metric tells us that the average IOPS consumed percentage across all the disks attached is around 42%. This percentage is calculated based on the IOPS that are used by the disks, and aren't being served from the host cache. Let's drill deeper into this metric by applying *splitting* on these metrics and splitting by the LUN value:



This metric tells us the data disks attached on LUN 3 and 2 are using around 85% of their provisioned IOPS. Here is a diagram of what the IO looks like from the VM and disks architecture:



Next steps

- Azure Monitor Metrics overview
- Metrics aggregation explained
- Create, view, and manage metric alerts using Azure Monitor

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Managed disk bursting

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure offers the ability to boost disk storage IOPS and MB/s performance, this is referred to as bursting for both virtual machines (VM) and disks. You can effectively use VM and disk bursting to achieve better bursting performance on both your VMs and disk.

Bursting for Azure VMs and disk resources aren't dependent on each other. You don't need a burst-capable VM for an attached burst-capable disk to burst. Similarly, you don't need a burst-capable disk attached to your burst-capable VM for the VM to burst.

Common scenarios

The following scenarios can benefit greatly from bursting:

- **Improve startup times** – With bursting, your instance starts up faster. For example, the default OS disk for premium enabled VMs is the P4 disk, which is a provisioned performance of up to 120 IOPS and 25 MB/s. With bursting, the P4 can go up to 3,500 IOPS and 170 MB/s, so startup accelerates by up to 6x.
- **Handle batch jobs** – Some application workloads are cyclical in nature. They require a baseline performance most of the time, and higher performance for short periods of time. An example of this nature is an accounting program that processes daily transactions that require a small amount of disk traffic. At the end of the month, this program completes reconciling reports that need a much higher amount of disk traffic.
- **Traffic spikes** – Web servers and their applications can experience traffic surges at any time. If your web server is backed by VMs or disks that use bursting, the servers are better equipped to handle traffic spikes.

Disk-level bursting

Currently, two managed disk types support bursting: Premium SSD managed disks and Standard SSDs. Other disk types don't support bursting. There are two models of bursting for disks:

- An on-demand bursting model, where the disk bursts whenever its needs exceed its current capacity. This model incurs extra charges anytime the disk bursts. On-demand bursting is only available for Premium SSDs larger than 512 GiB.

- A credit-based model, where the disk bursts only if it has burst credits accumulated in its credit bucket. This model doesn't incur extra charges when the disk bursts. Credit-based bursting operates on a best effort basis and isn't guaranteed. Credit-based bursting is only available for Premium SSD managed disks 512 GiB and smaller, and Standard SSDs 1,024 GiB and smaller.

Azure Premium SSD managed disks can use either bursting model, but Standard SSDs currently only offer credit-based bursting.

You can also change the performance tier of managed disks, which could be ideal if your workload would otherwise be running in burst.

Expand table

	Credit-based bursting	On-demand bursting	Changing performance tier
Scenarios	Ideal for short-term scaling (30 minutes or less).	Ideal for short-term scaling(Not time restricted).	Ideal if your workload would otherwise continually be running in burst.
Cost	Free	Cost is variable, see the Billing section for details.	The cost of each performance tier is fixed, see managed disk pricing for details.
Availability	Only available for Premium SSD managed disks 512 GiB and smaller, and Standard SSDs 1,024 GiB and smaller.	Only available for Premium SSD managed disks larger than 512 GiB.	Available to all Premium SSD sizes.
Enablement	Enabled by default on eligible disks.	Must be enabled by user.	User must manually change their tier.

On-demand bursting

Premium SSD managed disks using the on-demand bursting model of disk bursting can burst beyond original provisioned targets, as often as needed by their workload, up to the max burst target. For example, on a 1-TiB P30 disk, the provisioned IOPS is 5000 IOPS. When disk bursting is enabled on this disk, your workloads can issue IOs to this disk up to the max burst performance of 30,000 IOPS and 1,000 MBps. For the max burst targets on each supported disk, see Scalability and performance targets for VM disks.

If you expect your workloads to frequently run beyond the provisioned perf target, disk bursting won't be cost-effective. In this case, we recommend that you change your disk's

performance tier to a higher tier instead, for better baseline performance. Review your billing details and assess that against the traffic pattern of your workloads.

Before you enable on-demand bursting, understand the following:

- On-demand bursting cannot be enabled on a Premium SSD that has less than or equal to 512 GiB. Premium SSDs less than or equal to 512 GiB will always use credit-based bursting.
- On-demand bursting is only supported on Premium SSDs. If a Premium SSD with on-demand bursting enabled is switched to another disk type, then disk bursting is disabled.
- On-demand bursting doesn't automatically disable itself when the performance tier is changed. If you want to change your performance tier but do not want to keep disk bursting, you must disable it.
- On-demand bursting can only be enabled when the disk is detached from a VM or when the VM is stopped. On-demand bursting can be disabled 12 hours after it has been enabled.

Billing

Premium SSD managed disks using the on-demand bursting model are charged an hourly burst enablement flat fee and transaction costs apply to any burst transactions beyond the provisioned target. Transaction costs are charged using the pay-as-you go model, based on uncached disk IOs, including both reads and writes that exceed provisioned targets. The following is an example of disk traffic patterns over a billing hour:

Disk configuration: Premium SSD – 1 TiB (P30), Disk bursting enabled.

- 00:00:00 – 00:10:00 Disk IOPS below provisioned target of 5,000 IOPS
- 00:10:01 – 00:10:10 Application issued a batch job causing the disk IOPS to burst at 6,000 IOPS for 10 seconds
- 00:10:11 – 00:59:00 Disk IOPS below provisioned target of 5,000 IOPS
- 00:59:01 – 01:00:00 Application issued another batch job causing the disk IOPS to burst at 7,000 IOPS for 60 seconds

In this billing hour, the cost of bursting consists of two charges:

The first charge is the burst enablement flat fee of $X (determined by your region). This flat fee is always charged on the disk disregard of the attach status until it's disabled.

Second is the burst transaction cost. Disk bursting occurred in two time slots. From 00:10:01 – 00:10:10, the accumulated burst transaction is (6,000 – 5,000) X 10 = 10,000. From 00:59:01 –

01:00:00, the accumulated burst transaction is (7,000 – 5,000) X 60 = 120,000. The total burst transactions are 10,000 + 120,000 = 130,000. Burst transaction cost is charged at $Y based on 13 units of 10,000 transactions (based on regional pricing).

With that, the total cost on disk bursting of this billing hour equals to $X + $Y. The same calculation would apply for bursting over provisioned target of MBps. We translate the overage of MB to transactions with IO size of 256 KB. If your disk traffic exceed both provisioned IOPS and MBps target, you can refer to the example below to calculate the burst transactions.

Disk configuration: Premium SSD – 1 TB (P30), Disk bursting enabled.

- 00:00:01 – 00:00:05 Application issued a batch job causing the disk IOPS to burst at 10,000 IOPS and 300 MBps for five seconds.
- 00:00:06 – 00:00:10 Application issued a recovery job causing the disk IOPS to burst at 6,000 IOPS and 600 MBps for five seconds.

The burst transaction is accounted as the max number of transactions from either IOPS or MBps bursting. From 00:00:01 – 00:00:05, the accumulated burst transaction is Max((10,000 – 5,000), (300 - 200) * 1024 / 256)) * 5 = 25,000 transactions. From 00:00:06 – 00:00:10, the accumulated burst transaction is Max((6,000 – 5,000), (600 - 200) * 1024 / 256)) * 5 = 8,000 transactions. On top of that, you include the burst enablement flat fee to get the total cost for enabling on-demand based disk bursting.

You can refer to the managed disks pricing page for details on pricing and use Azure Pricing Calculator to make the assessment for your workload.

To enable on-demand bursting, see Enable on-demand bursting.

Credit-based bursting

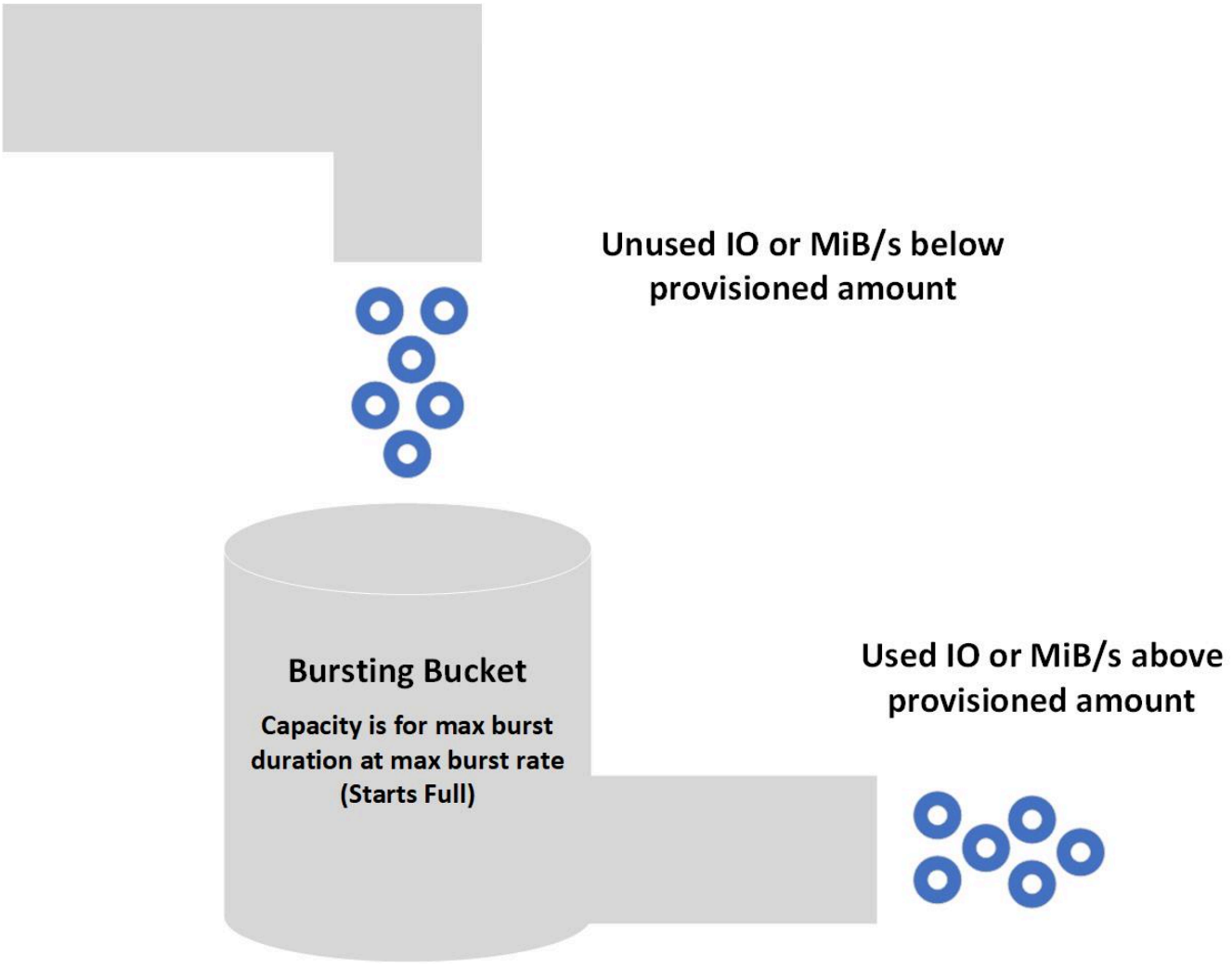
For Premium SSD managed disks, credit-based bursting is available for disk sizes P20 and smaller. For Standard SSDs, credit-based bursting is available for disk sizes E30 and smaller. For both standard and Premium SSD managed disks, credit-based bursting is available in all regions in Azure Public, Government, and China Clouds. By default, disk bursting is enabled on all new and existing deployments of supported disk sizes. VM-level bursting only uses credit-based bursting.

Virtual machine-level bursting

VM-level bursting only uses the credit-based model for bursting, it's enabled by default for most Premium Storage supported VMs.

Bursting flow

The bursting credit system applies in the same manner at both the VM level and disk level. Your resource, either a VM or disk, will start with fully stocked credits in its own burst bucket. These credits allow you to burst for up to 30 minutes at the maximum burst rate. You accumulate credits whenever the resource's IOPS or MB/s are being utilized below the resource's performance target. If your resource has accrued bursting credits and your workload needs the extra performance, your resource can use those credits to go above its performance limits on a best-effort basis to help meet workload demands.



How you spend your available credits is up to you. You can use your 30 minutes of burst credits consecutively or sporadically throughout the day. When resources are deployed, they come with a full allocation of credits. When those deplete, it takes less than a day to restock. Credits can be spent at your discretion. The burst bucket doesn't need to be full in order for resources to burst. Burst accumulation varies depending on each resource, since it's based on unused

IOPS and MB/s below their performance targets. Higher baseline performance resources can accrue their bursting credits faster than lower baseline performing resources. For example, a P1 disk idling accrues 120 IOPS per second, whereas an idling P20 disk would accrue 2,300 IOPS per second.

Bursting states

There are three states your resource can be in with bursting enabled:

- **Accruing** – The resource's IO traffic is using less than the performance target. Accumulating bursting credits for IOPS and MB/s are done separate from one another. Your resource can be accruing IOPS credits and spending MB/s credits or vice versa.
- **Bursting** – The resource's traffic is using more than the performance target. The burst traffic will independently consume credits from IOPS or bandwidth.
- **Constant** – The resource's traffic is exactly at the performance target.

Bursting examples

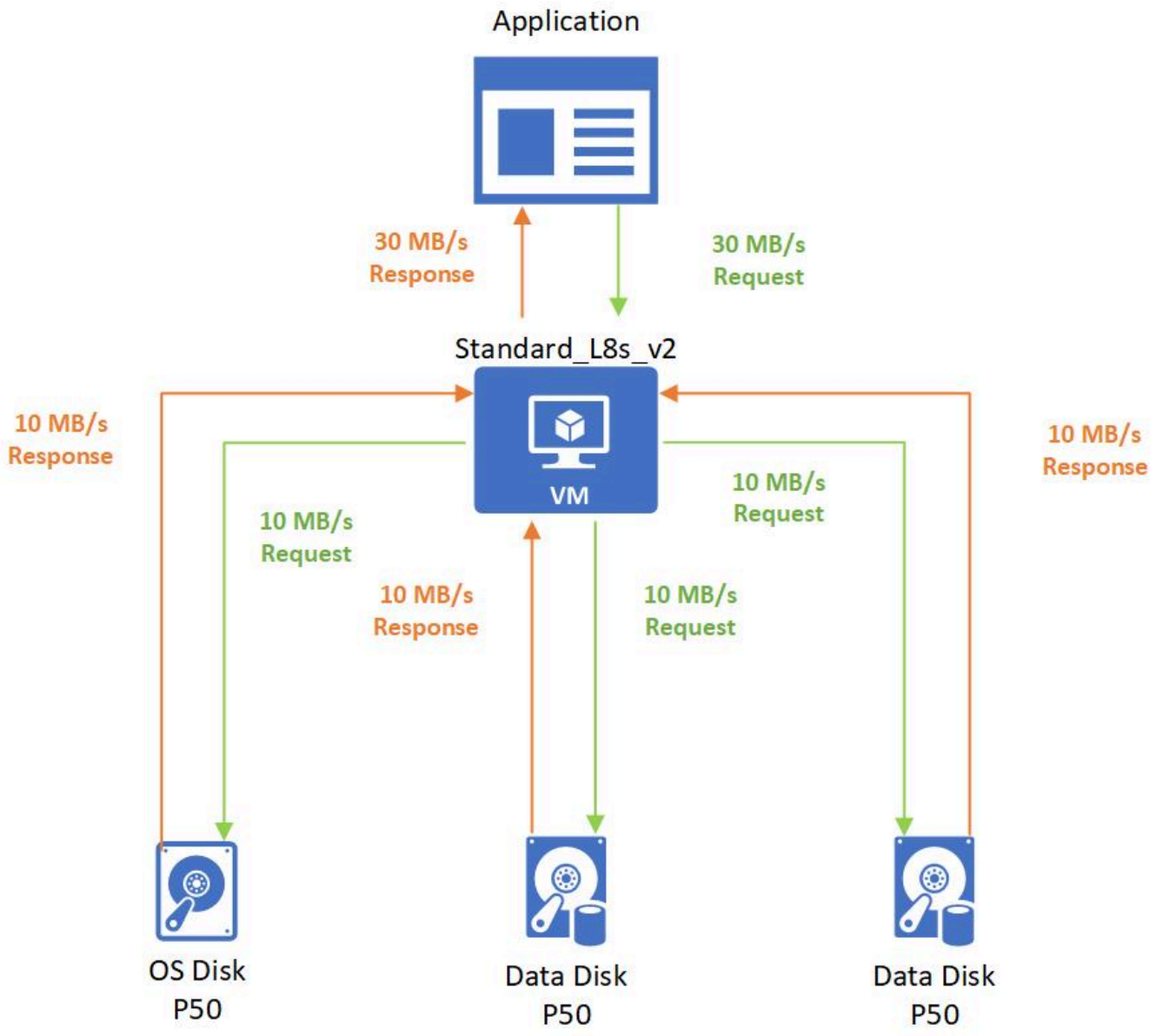
The following examples show how bursting works with various VM and disk combinations. To make the examples easy to follow, we focus on MB/s, but the same logic is applied independently to IOPS.

Burstable virtual machine with nonburstable disks

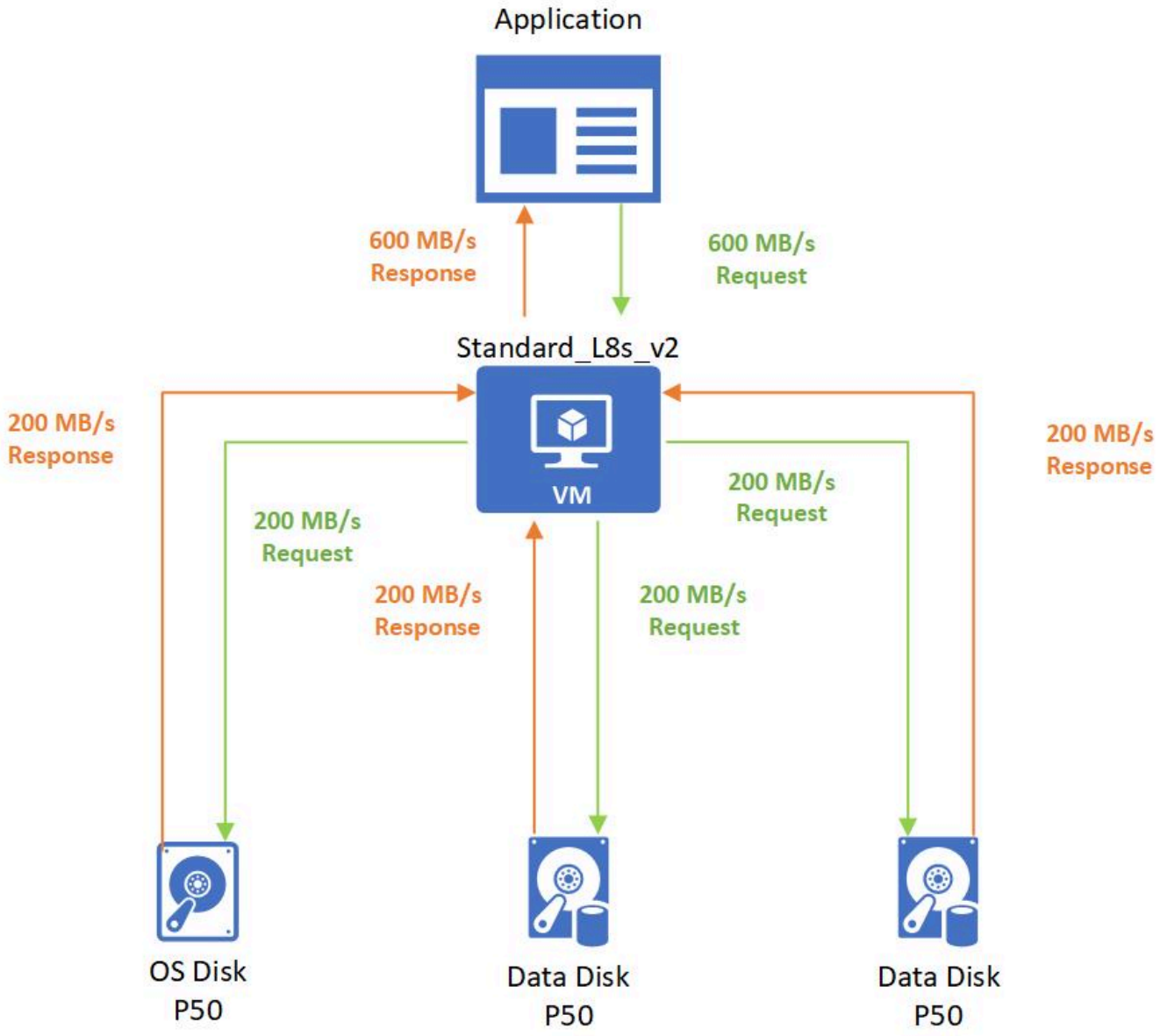
VM and disk combination:

- Standard_L8s_v2
  - Uncached MB/s: 160
  - Max burst MB/s: 1,280
- P50 OS Disk
  - Provisioned MB/s: 250
  - On-Demand Bursting: **not enabled**
- 2 P50 Data Disks
  - Provisioned MB/s: 250
  - On-Demand Bursting: **not enabled**

After the initial boot up, an application is run on the VM and has a noncritical workload. This workload requires 30 MB/s that gets spread evenly across all the disks.



Then the application needs to process a batched job that requires 600 MB/s. The Standard_L8s_v2 bursts to meet this demand and then requests to the disks get evenly spread out to P50 disks.

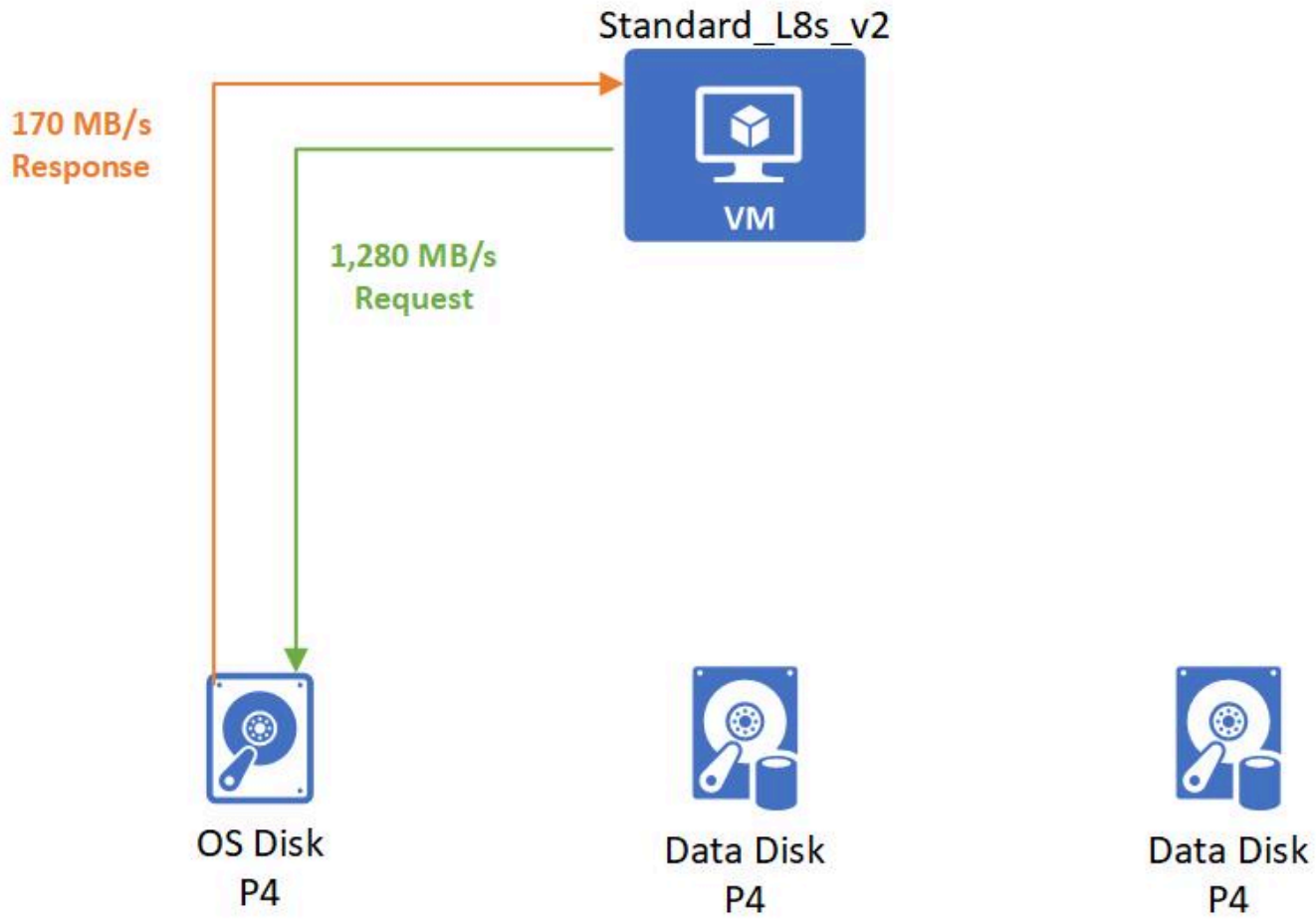


Burstable virtual machine with burstable disks

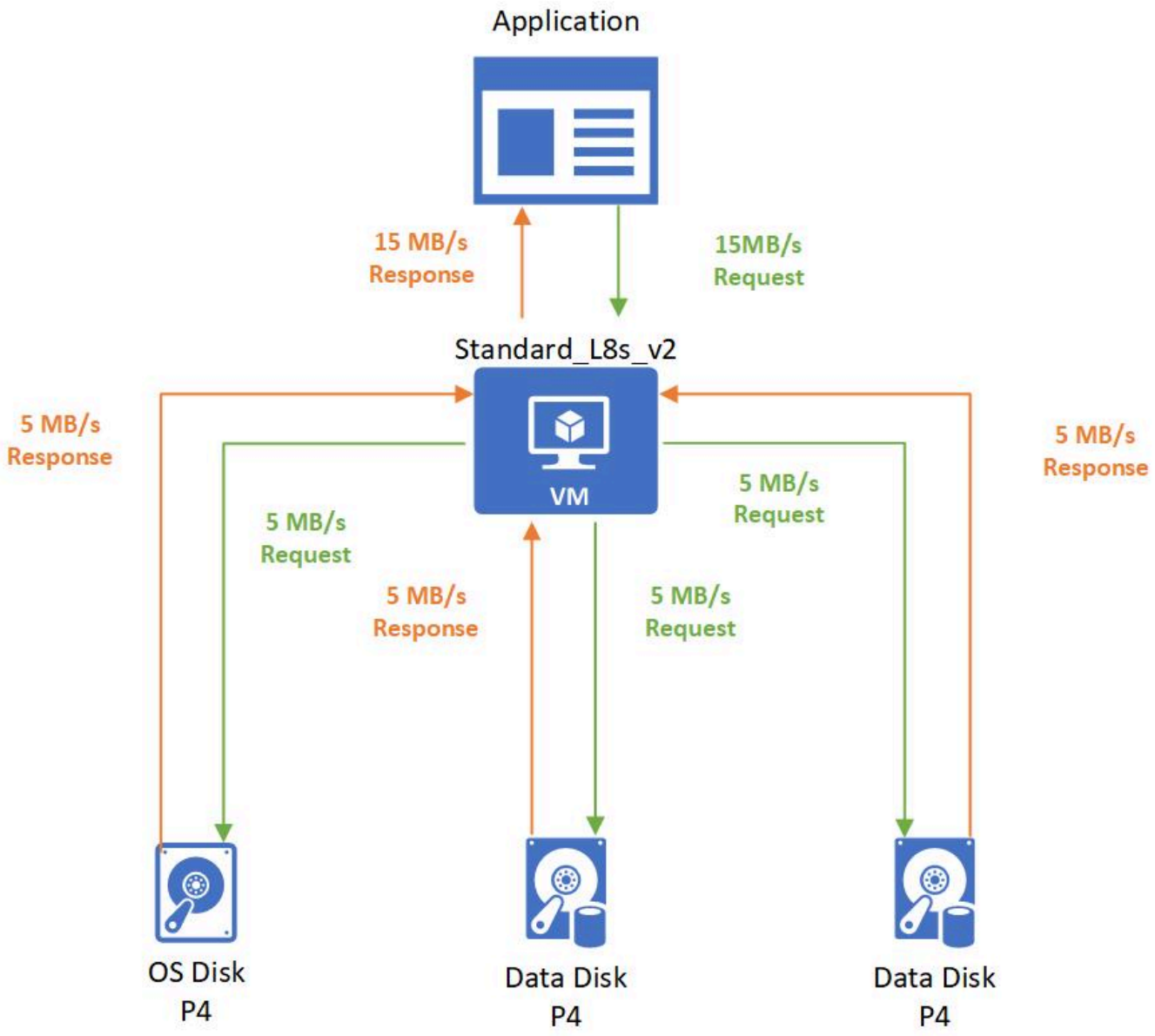
VM and disk combination:

- Standard_L8s_v2
  - Uncached MB/s: 160
  - Max burst MB/s: 1,280
- P4 OS Disk
  - Provisioned MB/s: 25
  - Max burst MB/s: 170
- 2 P4 Data Disks
  - Provisioned MB/s: 25
  - Max burst MB/s: 170

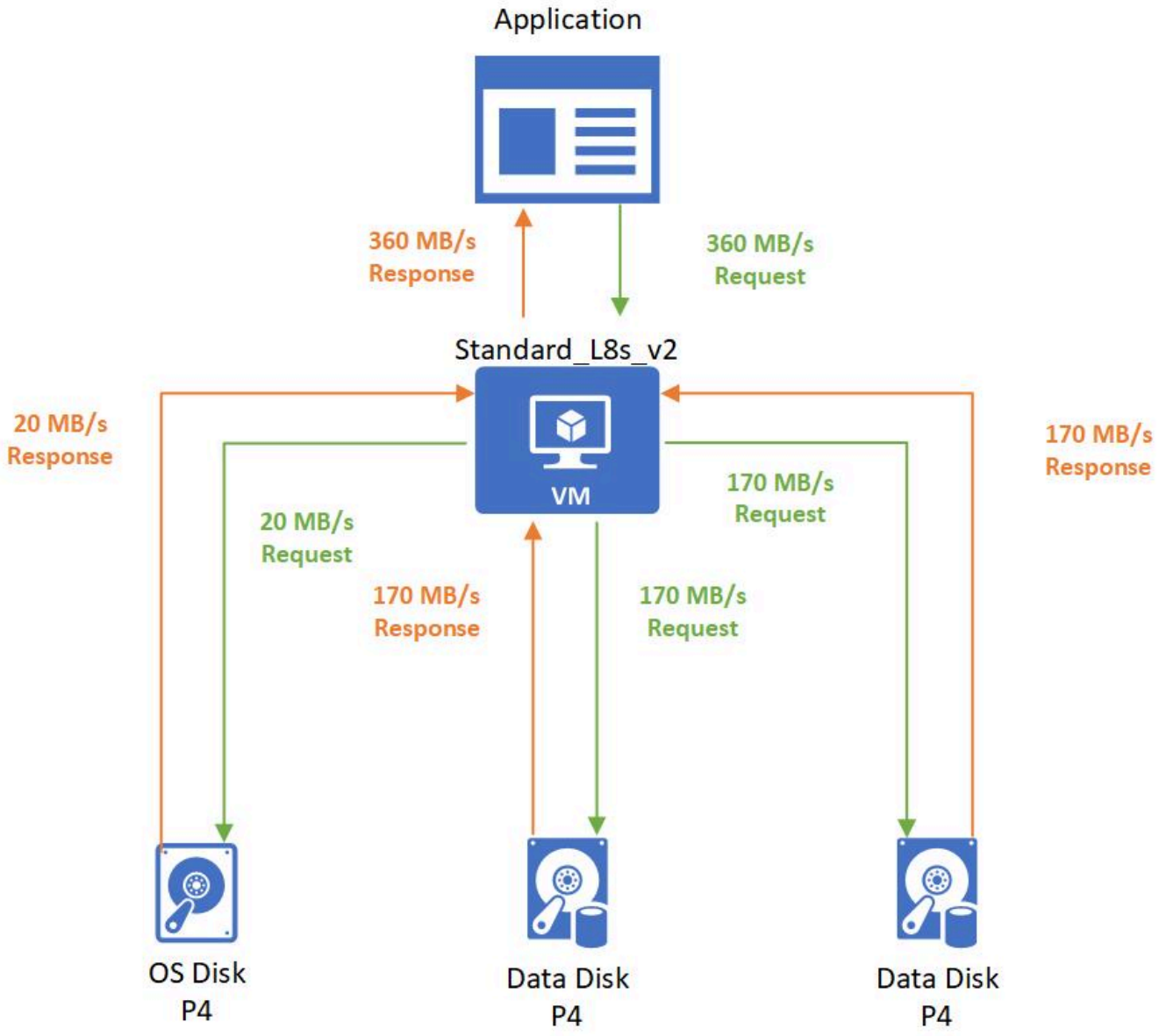
When the VM starts, it will burst to request its burst limit of 1,280 MB/s from the OS disk, and the OS disk will respond with its burst performance of 170 MB/s.



After startup, you start an application that has a noncritical workload. This application requires 15 MB/s that gets spread evenly across all the disks.



Then the application needs to process a batched job that requires 360 MB/s. The Standard_L8s_v2 bursts to meet this demand and then requests. Only 20 MB/s are needed by the OS disk. The remaining 340 MB/s are handled by the bursting P4 data disks.



Next steps

- To enable on-demand bursting, see Enable on-demand bursting.
- To learn how to gain insight into your bursting resources, see Disk bursting metrics.
- To see exactly how much each applicable disk size can burst, see Scalability and performance targets for VM disks.

Last updated on 04/15/2026

Enable on-demand bursting

Article • 01/14/2025

Premium solid-state drives (SSD) have two available bursting models; credit-based bursting and on-demand bursting. This article covers how to switch to on-demand bursting. Disks that use the on-demand model can burst beyond their original provisioned targets. On-demand bursting occurs as often as needed by the workload, up to the maximum burst target. On-demand bursting incurs additional charges.

For details on disk bursting, see Managed disk bursting.

For the max burst targets on each supported disk, see Scalability and performance targets for VM disks.

> **Important**
>
> You don't need to follow the steps in this article to use credit-based bursting. By default, credit-based bursting is enabled on all eligible disks.

Before you enable on-demand bursting, understand the following:

- On-demand bursting cannot be enabled on a premium SSD that has less than or equal to 512 GiB. Premium SSDs less than or equal to 512 GiB will always use credit-based bursting.
- On-demand bursting is only supported on premium SSDs. If a premium SSD with on-demand bursting enabled is switched to another disk type, then disk bursting is disabled.
- On-demand bursting doesn't automatically disable itself when the performance tier is changed. If you want to change your performance tier but do not want to keep disk bursting, you must disable it.
- On-demand bursting can only be enabled when the disk is detached from a VM or when the VM is stopped. On-demand bursting can be disabled 12 hours after it has been enabled.

Get started

On-demand bursting can be enabled with either the Azure portal, the Azure PowerShell module, the Azure CLI, or Azure Resource Manager templates. The following examples cover how to create a new disk with on-demand bursting enabled and enabling on-demand bursting on existing disks.

Portal

A managed disk must be larger than 512 GiB to enable on-demand bursting.

To enable on-demand bursting for an existing disk:

1. Sign in to the Azure portal and navigate to your disk.
2. Select **Configuration** and select **Enable on-demand bursting**.
3. Select **Save**.

Next steps

To learn how to gain insight into your bursting resources, see Disk bursting metrics.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Performance tiers for managed disks

Article • 01/23/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

> **Note**
>
> This article covers what performance tiers are, conceptually. If you want to learn how to change the performance of disks that don't use performance tiers, like Ultra Disks or Premium SSD v2, see either Adjust the performance of an ultra disk or Adjust disk performance of a Premium SSD v2

When you set the provisioned size of a Premium solid-state drive (SSD), a performance tier is automatically selected based on the size you set. The performance tier determines the IOPS and throughput your managed disk has. For Premium SSD disks only, the performance tier can be changed at deployment or afterwards, without changing the size of the disk, and without downtime.

Changing the performance tier allows you to prepare to meet higher demand without using your disk's bursting capability. It can be more cost-effective to change your performance tier rather than rely on bursting, depending on how long the extra performance is necessary. This is ideal for events that temporarily require a consistently higher level of performance, like holiday shopping, performance testing, or running a training environment. To handle these events, you can switch a disk to a higher performance tier without downtime, for as long as you need the extra performance. You can then return to the original tier without downtime when the extra performance is no longer necessary.

To learn more about how the performance of a disk works with the performance of a virtual machine, see Virtual machine and disk performance.

Restrictions

- Changing the performance tier is currently only supported for Premium SSD managed disks.
- Performance tiers of shared disks can't be changed while attached to running VMs.
  - To change the performance tier of a shared disk, stop all the VMs the disk is attached to.
- Only disks larger than 4,096 GiB can use the P60, P70, and P80 performance tiers.

- A disk's performance tier can be downgraded only once every 12 hours.
- The system doesn't return `Performance Tier` for disks created before June 2020. You can take advantage of `Performance Tier` for an older disk by updating it with the baseline Tier.

How it works

When you first deploy or provision a disk, the baseline performance tier for that disk is set based on the provisioned disk size. You can use a performance tier higher than the original baseline to meet higher demand. When you no longer need that performance level, you can return to the initial baseline performance tier.

Billing impact

Disk billing changes as its performance tier changes. For example, if you provision a P10 disk (128 GiB), your baseline performance tier is set as P10 (500 IOPS and 100 MBps). Your disk is billed at the P10 rate. You can set the disk's performance tier to P50 (7,500 IOPS and 250 MBps) without increasing the disk size. While the disk's performance tier is set to P50, your disk is billed at the P50 rate. When you no longer need the higher performance, you can set the performance tier of the disk back to the P10 tier and your disk's billing will return to the P10 rate.

For billing information, see Managed disk pricing.

What tiers can be changed

The following table depicts which tiers each baseline performance tier can upgrade to.

Expand table

Disk size	Baseline performance tier	Can be upgraded to
4 GiB	P1	P2, P3, P4, P6, P10, P15, P20, P30, P40, P50
8 GiB	P2	P3, P4, P6, P10, P15, P20, P30, P40, P50
16 GiB	P3	P4, P6, P10, P15, P20, P30, P40, P50
32 GiB	P4	P6, P10, P15, P20, P30, P40, P50
64 GiB	P6	P10, P15, P20, P30, P40, P50
128 GiB	P10	P15, P20, P30, P40, P50

Disk size	Baseline performance tier	Can be upgraded to
256 GiB	P15	P20, P30, P40, P50
512 GiB	P20	P30, P40, P50
1 TiB	P30	P40, P50
2 TiB	P40	P50
4 TiB	P50	None
8 TiB	P60	P70, P80
16 TiB	P70	P80
32 TiB	P80	None

Next steps

To learn how to change your performance tier, see Change your performance tier without downtime.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Change your performance tier without downtime

Article • 01/16/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

> **Note**
>
> This article focuses on how to change performance tiers. To learn how to change the performance of disks that don't use performance tiers, like Ultra Disks or Premium SSD v2, see either Adjust the performance of an ultra disk or Adjust disk performance of a Premium SSD v2

The performance of your Azure managed disk is set when you create your disk, in the form of its performance tier. The performance tier determines the IOPS and throughput your managed disk has. When you set the provisioned size of your disk, a performance tier is automatically selected. The performance tier can be changed at deployment or afterwards, without changing the size of the disk and without downtime. To learn more about performance tiers, see Performance tiers for managed disks.

Changing your performance tier has billing implications. See Billing impact for details.

Restrictions

- Changing the performance tier is currently only supported for Premium SSD managed disks.
- Performance tiers of shared disks can't be changed while attached to running VMs.
  - To change the performance tier of a shared disk, stop all the VMs the disk is attached to.
- Only disks larger than 4,096 GiB can use the P60, P70, and P80 performance tiers.
- A disk's performance tier can be downgraded only once every 12 hours.
- The system doesn't return `Performance Tier` for disks created before June 2020. You can take advantage of `Performance Tier` for an older disk by updating it with the baseline Tier.

Prerequisites

Azure CLI

Install the latest Azure CLI and sign in to an Azure account with az login.

Create an empty data disk with a tier higher than the baseline tier

Azure CLI

Azure CLI

```
subscriptionId=<yourSubscriptionIDHere>
resourceGroupName=<yourResourceGroupNameHere>
diskName=<yourDiskNameHere>
diskSize=<yourDiskSizeHere>
performanceTier=<yourDesiredPerformanceTier>
region=westcentralus

az account set --subscription $subscriptionId

az disk create -n $diskName -g $resourceGroupName -l $region --sku
Premium_LRS --size-gb $diskSize --tier $performanceTier
```

Create an OS disk with a tier higher than the baseline tier from an Azure Marketplace image

Azure CLI

```
resourceGroupName=<yourResourceGroupNameHere>
diskName=<yourDiskNameHere>
performanceTier=<yourDesiredPerformanceTier>
region=westcentralus
image=Canonical:UbuntuServer:18.04-LTS:18.04.202002180

az disk create -n $diskName -g $resourceGroupName -l $region --image-
reference $image --sku Premium_LRS --tier $performanceTier
```

Update the tier of a disk without downtime

A disk's performance tier can be changed without downtime, so you don't have to deallocate your VM or detach your disk to change the tier.

Azure CLI

1. Update the tier of a disk even when it is attached to a running VM

```
Azure CLI
```

```
resourceGroupName=<yourResourceGroupNameHere>
diskName=<yourDiskNameHere>
performanceTier=<yourDesiredPerformanceTier>

az disk update -n $diskName -g $resourceGroupName --set
tier=$performanceTier
```

Show the tier of a disk

Azure CLI

```
Azure CLI
```

```
az disk show -n $diskName -g $resourceGroupName --query [tier] -o tsv
```

Next steps

If you need to resize a disk to take advantage of the higher performance tiers, see these articles:

- Expand virtual hard disks on a Linux VM with the Azure CLI
- Expand a managed disk attached to a Windows virtual machine

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Enable Write Accelerator

Article • 01/14/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Write Accelerator is a disk capability for M-Series Virtual Machines (VMs) on Premium SSD managed disks. As the name states, the purpose of the functionality is to improve the I/O latency of writes against Premium SSD disks. Write Accelerator is ideally suited where log file updates are required to persist to disk in a highly performant manner for modern databases.

Write Accelerator is generally available for M-series VMs in the Public Cloud.

Prerequisites

- Only supported with Premium SSD managed disks
- Only supported by M-series VMs

Planning for using Write Accelerator

Write Accelerator should be used for the volumes that contain the transaction log or redo logs of a DBMS. You shouldn't use Write Accelerator for the data volumes of a DBMS as the feature is optimized to be used against log disks.

> **ⓘ Important**
>
> Enabling Write Accelerator for the operating system disk of the VM reboots the VM.
>
> To enable Write Accelerator to an existing Azure disk that isn't part of a volume build out of multiple disks with Windows disk or volume managers, Windows Storage Spaces, Windows Scale-out file server (SOFS), Linux LVM, or MDADM, the workload accessing the Azure disk needs to be shut down. Database applications using the Azure disk must be shut down.
>
> To enable or disable Write Accelerator for an existing volume that is built out of multiple Azure Premium SSD disks and striped using Windows disk or volume managers, Windows Storage Spaces, Windows Scale-out file server (SOFS), Linux LVM or MDADM, all disks building the volume must be enabled or disabled for

Write Accelerator in separate steps. Shut down the VM Before enabling or disabling Write Accelerator in such a configuration.

Enabling Write Accelerator for OS disks shouldn't be necessary for SAP-related VM configurations.

Restrictions when using Write Accelerator

When using Write Accelerator for an Azure disk/VHD, these restrictions apply:

- The Premium disk caching must be set to 'None' or 'Read Only'. All other caching modes aren't supported.
- Snapshots are currently supported for only Write Accelerator-enabled data disks, and not the OS disk. During backup, the Azure Backup service automatically backs up and protects Write Accelerator-enabled data disks attached to the VM.
- Only smaller I/O sizes (<=64 KiB) are taking the accelerated path. In workload situations where data is getting bulk loaded or where the transaction log buffers of the different DBMS are filled to a larger degree before getting persisted to the storage, chances are that the I/O written to disk is not taking the accelerated path.

There are limits of Azure Premium SSD disks per VM that can be supported by Write Accelerator. The current limits are:

Expand table

VM SKU	Number of Write Accelerator disks	Write Accelerator Disk IOPS per VM
M416ms_v2, M416s_8_v2, M416s_v2	16	20000
M208ms_v2, M208s_v2	8	10000
M192ids_v2, M192idms_v2, M192is_v2, M192ims_v2,	16	20000
M128ms, M128s, M128ds_v2, M128dms_v2, M128s_v2, M128ms_v2	16	20000
M64ms, M64ls, M64s, M64ds_v2, M64dms_v2, M64s_v2, M64ms_v2	8	10000
M32ms, M32ls, M32ts, M32s, M32dms_v2, M32ms_v2	4	5000
M16ms, M16s	2	2500

VM SKU	Number of Write Accelerator disks	Write Accelerator Disk IOPS per VM
M8ms, M8s	1	1250
Standard_M12s_v3, Standard_M12ds_v3	1	5000
Standard_M24s_v3, Standard_M24ds_v3	2	5000
Standard_M48s_1_v3, Standard_M48ds_1_v3	4	5000
Standard_M96s_1_v3, Standard_M96ds_1_v3, Standard_M96s_2_v3, Standard_M96ds_2_v3	8	10000
Standard_M176s_3_v3, Standard_M176ds_3_v3, Standard_M176s_4_v3, Standard_M176ds_4_v3	16	20000

The IOPS limits are per VM and not per disk. All Write Accelerator disks share the same IOPS limit per VM. Attached disks can't exceed the write accelerator IOPS limit for a VM. For an example, even though the attached disks can do 30,000 IOPS, the system doesn't allow the disks to go above 20,000 IOPS for M416ms_v2.

Enabling Write Accelerator on a specific disk

The next few sections describe how Write Accelerator can be enabled on Azure Premium SSD VHDs.

Enabling Azure Write Accelerator using Azure PowerShell

The Azure PowerShell module from version 5.5.0 include the changes to the relevant cmdlets to enable or disable Write Accelerator for specific Azure Premium SSD disks. In order to enable or deploy disks supported by Write Accelerator, the following PowerShell commands got changed, and extended to accept a parameter for Write Accelerator.

A new switch parameter, **-WriteAccelerator** has been added to the following cmdlets:

- Set-AzVMOsDisk
- Add-AzVMDataDisk
- Set-AzVMDataDisk
- Add-AzVmssDataDisk

> **Note**
>
> If enabling Write Accelerator on Virtual Machine Scale Sets using Flexible Orchestration Mode, you need to enable it on each individual instance.

Not giving the parameter sets the property to false and will deploy disks that have no support by Write Accelerator.

A new switch parameter, **-OsDiskWriteAccelerator** was added to the following cmdlets:

- Set-AzVmssStorageProfile

Not specifying the parameter sets the property to false by default, returning disks that don't use Write Accelerator.

A new optional Boolean (non-nullable) parameter, **-OsDiskWriteAccelerator** was added to the following cmdlets:

- Update-AzVM
- Update-AzVmss

Specify either `$true` or `$false` to control support of Azure Write Accelerator with the disks.

Examples of commands could look like:

Azure PowerShell

```
New-AzVMConfig | Set-AzVMOsDisk | Add-AzVMDataDisk -Name "datadisk1" | Add-
AzVMDataDisk -Name "logdisk1" -WriteAccelerator | New-AzVM

Get-AzVM | Update-AzVM -OsDiskWriteAccelerator $true

New-AzVmssConfig | Set-AzVmssStorageProfile -OsDiskWriteAccelerator | Add-
AzVmssDataDisk -Name "datadisk1" -WriteAccelerator:$false | Add-
AzVmssDataDisk -Name "logdisk1" -WriteAccelerator | New-AzVmss

Get-AzVmss | Update-AzVmss -OsDiskWriteAccelerator:$false
```

Two main scenarios can be scripted as shown in the following sections.

Adding a new disk supported by Write Accelerator using PowerShell

You can use this script to add a new disk to your VM. The disk created with this script uses Write Accelerator.

Replace `myVM`, `myWAVMs`, `log001`, size of the disk, and LunID of the disk with values appropriate for your specific deployment.

Azure PowerShell

```
# Specify your VM Name
$vmName="myVM"
#Specify your Resource Group
$rgName = "myWAVMs"
#data disk name
$datadiskname = "log001"
#LUN Id
$lunid=8
#size
$size=1023
#Pulls the VM info for later
$vm=Get-AzVM -ResourceGroupName $rgname -Name $vmname
#add a new VM data disk
Add-AzVMDataDisk -CreateOption empty -DiskSizeInGB $size -Name
$vmname-$datadiskname -VM $vm -Caching None -WriteAccelerator:$true -lun
$lunid
#Updates the VM with the disk config - does not require a reboot
Update-AzVM -ResourceGroupName $rgname -VM $vm
```

Enabling Write Accelerator on an existing Azure disk using PowerShell

You can use this script to enable Write Accelerator on an existing disk. Replace `myVM`, `myWAVMs`, and `test-log001` with values appropriate for your specific deployment. The script adds Write Accelerator to an existing disk where the value for **$newstatus** is set to '$true'. Using the value '$false' will disable Write Accelerator on a given disk.

Azure PowerShell

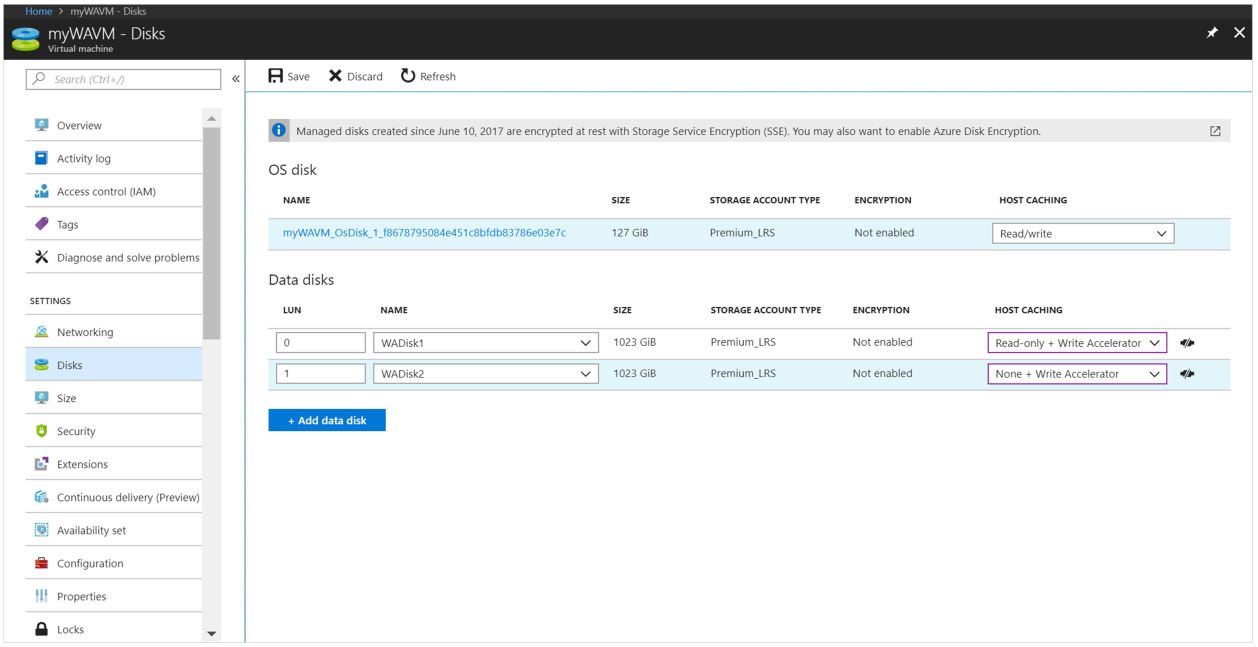
```
#Specify your VM Name
$vmName="myVM"
#Specify your Resource Group
$rgName = "myWAVMs"
#data disk name
$datadiskname = "test-log001"
#new Write Accelerator status ($true for enabled, $false for disabled)
$newstatus = $true
#Pulls the VM info for later
$vm=Get-AzVM -ResourceGroupName $rgname -Name $vmname
#add a new VM data disk
```

```
Set-AzVMDataDisk -VM $vm -Name $datadiskname -Caching None -
WriteAccelerator:$newstatus
#Updates the VM with the disk config - does not require a reboot
Update-AzVM -ResourceGroupName $rgname -VM $vm
```

> **Note**
>
> Executing the script above will detach the disk specified, enable Write Accelerator against the disk, and then attach the disk again

Enabling Write Accelerator using the Azure portal

You can enable Write Accelerator via the portal where you specify your disk caching settings:



Enabling Write Accelerator using the Azure CLI

You can use the Azure CLI to enable Write Accelerator.

To enable Write Accelerator on an existing disk, use az vm update, you can use the following examples if you replace the `diskName`, `VMName`, and `ResourceGroup` with your own values: `az vm update -g group1 -n vm1 -write-accelerator 1=true`

To attach a disk with Write Accelerator enabled use az vm disk attach, you can use the following example if you substitute in your own values: `az vm disk attach -g group1 -`

`vm-name vm1 -disk d1 --enable-write-accelerator`

To disable Write Accelerator, use az vm update, setting the properties to false: `az vm update -g group1 -n vm1 -write-accelerator 0=false 1=false`

Enabling Write Accelerator using REST APIs

To deploy through Azure REST API, you need to install the Azure armclient.

Install armclient

To run armclient, you need to install it through Chocolatey. You can install it through cmd.exe or PowerShell. Use elevated rights for these commands ("Run as Administrator").

Using cmd.exe, run the following command: `@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"`

Using PowerShell, run the following command: `Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))`

Now you can install the armclient by using the following command in either cmd.exe or PowerShell `choco install armclient`

Getting your current VM configuration

To change the attributes of your disk configuration, you first need to get the current configuration in a JSON file. You can get the current configuration by executing the following command: `armclient GET /subscriptions/<<subscription-ID<</resourceGroups/<<ResourceGroup>>/providers/Microsoft.Compute/virtualMachines/<<virtualmachinename>>?api-version=2017-12-01 > <<filename.json>>`

Replace the terms within '<< >>' with your data, including the file name the JSON file should have.

The output could look like:

```
```

Output


```
{
  "properties": {
    "vmId": "2444c93e-f8bb-4a20-af2d-1658d9dbbbcb",
    "hardwareProfile": {
      "vmSize": "Standard_M64s"
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "SUSE",
        "offer": "SLES-SAP",
        "sku": "12-SP3",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Linux",
        "name": "mylittlesap_OsDisk_1_754a1b8bb390468e9b4c429b81cc5f5a",
        "createOption": "FromImage",
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS",
          "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/disks/mylittlesap_OsDisk_1_754a1b8bb390468e9
b4c429b81cc5f5a"
        },
        "diskSizeGB": 30
      },
      "dataDisks": [
        {
          "lun": 0,
          "name": "data1",
          "createOption": "Attach",
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Premium_LRS",
            "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/disks/data1"
          },
          "diskSizeGB": 1023
        },
        {
          "lun": 1,
          "name": "log1",
          "createOption": "Attach",
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Premium_LRS",
            "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/disks/data2"
          },
```

```
          "diskSizeGB": 1023
        }
      ]
    },
    "osProfile": {
      "computerName": "mylittlesapVM",
      "adminUsername": "pl",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      },
      "secrets": []
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Network/networkInterfaces/mylittlesap518"
        }
      ]
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "enabled": true,
        "storageUri": "https://mylittlesapdiag895.blob.core.windows.net/"
      }
    },
    "provisioningState": "Succeeded"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "location": "westeurope",
  "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/virtualMachines/mylittlesapVM",
  "name": "mylittlesapVM"
```

Next, update the JSON file and to enable Write Accelerator on the disk called 'log1'. This can be accomplished by adding this attribute into the JSON file after the cache entry of the disk.

JSON

```
        {
          "lun": 1,
          "name": "log1",
          "createOption": "Attach",
          "caching": "None",
          "writeAcceleratorEnabled": true,
          "managedDisk": {
            "storageAccountType": "Premium_LRS",
            "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
```

```
sap/providers/Microsoft.Compute/disks/data2"
          },
          "diskSizeGB": 1023
        }
```

Then update the existing deployment with this command: `armclient PUT /subscriptions/<<subscription-ID<</resourceGroups/<<ResourceGroup>>/providers/Microsoft.Compute/virtualMachines/<<virtualmachinename>>?api-version=2017-12-01 @<<filename.json>>`

The output should look like the one below. You can see that Write Accelerator enabled for one disk.

Output

```
{
  "properties": {
    "vmId": "2444c93e-f8bb-4a20-af2d-1658d9dbbbcb",
    "hardwareProfile": {
      "vmSize": "Standard_M64s"
    },
    "storageProfile": {
      "imageReference": {
        "publisher": "SUSE",
        "offer": "SLES-SAP",
        "sku": "12-SP3",
        "version": "latest"
      },
      "osDisk": {
        "osType": "Linux",
        "name": "mylittlesap_OsDisk_1_754a1b8bb390468e9b4c429b81cc5f5a",
        "createOption": "FromImage",
        "caching": "ReadWrite",
        "managedDisk": {
          "storageAccountType": "Premium_LRS",
          "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/disks/mylittlesap_OsDisk_1_754a1b8bb390468e9
b4c429b81cc5f5a"
        },
        "diskSizeGB": 30
      },
      "dataDisks": [
        {
          "lun": 0,
          "name": "data1",
          "createOption": "Attach",
          "caching": "None",
          "managedDisk": {
            "storageAccountType": "Premium_LRS",
            "id":
```

```
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/disks/data1"
          },
          "diskSizeGB": 1023
        },
        {
          "lun": 1,
          "name": "log1",
          "createOption": "Attach",
          "caching": "None",
          "writeAcceleratorEnabled": true,
          "managedDisk": {
            "storageAccountType": "Premium_LRS",
            "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/disks/data2"
          },
          "diskSizeGB": 1023
        }
      ]
    },
    "osProfile": {
      "computerName": "mylittlesapVM",
      "adminUsername": "pl",
      "linuxConfiguration": {
        "disablePasswordAuthentication": false
      },
      "secrets": []
    },
    "networkProfile": {
      "networkInterfaces": [
        {
          "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Network/networkInterfaces/mylittlesap518"
        }
      ]
    },
    "diagnosticsProfile": {
      "bootDiagnostics": {
        "enabled": true,
        "storageUri": "https://mylittlesapdiag895.blob.core.windows.net/"
      }
    },
    "provisioningState": "Succeeded"
  },
  "type": "Microsoft.Compute/virtualMachines",
  "location": "westeurope",
  "id":
"/subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/resourceGroups/mylittle
sap/providers/Microsoft.Compute/virtualMachines/mylittlesapVM",
  "name": "mylittlesapVM"
```


Once you've made this change, the drive should be supported by Write Accelerator.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Benchmark a disk

Article • 01/07/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Benchmarking is the process of simulating different workloads on your application and measuring the application performance for each workload. Using the steps described in the designing for high performance article, you have gathered the application performance requirements. By running benchmarking tools on the VMs hosting the application, you can determine the performance levels that your application can achieve with premium SSDs. In this article, we provide you examples of benchmarking a Standard_D8ds_v4 VM provisioned with Azure premium SSDs.

We have used common benchmarking tools DiskSpd and FIO, for Windows and Linux respectively. These tools spawn multiple threads simulating a production like workload, and measure the system performance. Using the tools you can also configure parameters like block size and queue depth, which you normally cannot change for an application. This gives you more flexibility to drive the maximum performance on a high scale VM provisioned with premium SSDs for different types of application workloads. To learn more about each benchmarking tool visit DiskSpd and FIO.

To follow the examples below, create a Standard_D8ds_v4 and attach four premium SSDs to the VM. Of the four disks, configure three with host caching as "None" and stripe them into a volume called NoCacheWrites. Configure host caching as "ReadOnly" on the remaining disk and create a volume called CacheReads with this disk. Using this setup, you are able to see the maximum Read and Write performance from a Standard_D8ds_v4 VM. For detailed steps about creating a Standard_D8ds_v4 with premium SSDs, see Designing for high performance.

Warm up the Cache

The disk with ReadOnly host caching is able to give higher IOPS than the disk limit. To get this maximum read performance from the host cache, first you must warm up the cache of this disk. This ensures that the Read IOs that the benchmarking tool will drive on CacheReads volume, actually hits the cache, and not the disk directly. The cache hits result in more IOPS from the single cache enabled disk.

> **Important**

You must warm up the cache before running benchmarks every time VM is rebooted.

DISKSPD

Download the DISKSP tool on the VM. DISKSPD is a tool that you can customize to create your own synthetic workloads. We will use the same setup described above to run benchmarking tests. You can change the specifications to test different workloads.

In this example, we use the following set of baseline parameters:

- -c200G: Creates (or recreates) the sample file used in the test. It can be set in bytes, KiB, MiB, GiB, or blocks. In this case, a large file of 200-GiB target file is used to minimize memory caching.
- -w100: Specifies the percentage of operations that are write requests (-w0 is equivalent to 100% read).
- -b4K: Indicates the block size in bytes, KiB, MiB, or GiB. In this case, 4K block size is used to simulate a random I/O test.
- -F4: Sets a total of four threads.
- -r: Indicates the random I/O test (overrides the -s parameter).
- -o128: Indicates the number of outstanding I/O requests per target per thread. This is also known as the queue depth. In this case, 128 is used to stress the CPU.
- -W7200: Specifies the duration of the warm-up time before measurements start.
- -d30: Specifies the duration of the test, not including warm-up.
- -Sh: Disable software and hardware write caching (equivalent to -Suw).

For a complete list of parameters, see the GitHub repository.

Maximum write IOPS

We use a high queue depth of 128, a small block size of 8 KB, and four worker threads for driving Write operations. The write workers are driving traffic on the "NoCacheWrites" volume, which has three disks with cache set to "None".

Run the following command for 30 seconds of warm-up and 30 seconds of measurement:

```
diskspd -c200G -w100 -b8K -F4 -r -o128 -W30 -d30 -Sh testfile.dat
```

Results show that the Standard_D8ds_v4 VM is delivering its maximum write IOPS limit of 12,800.

```
Total IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  file
-----------------------------------------------------------------------------
     0 |       802029568 |        97904 |      25.49 |    3262.63 | G:\testfile.dat (200GiB)
     1 |       801382400 |        97825 |      25.47 |    3260.00 | G:\testfile.dat (200GiB)
     2 |       802217984 |        97927 |      25.50 |    3263.39 | G:\testfile.dat (200GiB)
     3 |       803012608 |        98024 |      25.52 |    3266.63 | G:\testfile.dat (200GiB)
-----------------------------------------------------------------------------
total:        3208642560 |       391680 |     101.97 |   13052.65

Read IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  file
-----------------------------------------------------------------------------
     0 |               0 |            0 |       0.00 |       0.00 | G:\testfile.dat (200GiB)
     1 |               0 |            0 |       0.00 |       0.00 | G:\testfile.dat (200GiB)
     2 |               0 |            0 |       0.00 |       0.00 | G:\testfile.dat (200GiB)
     3 |               0 |            0 |       0.00 |       0.00 | G:\testfile.dat (200GiB)
-----------------------------------------------------------------------------
total:                 0 |            0 |       0.00 |       0.00

Write IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  file
-----------------------------------------------------------------------------
     0 |       802029568 |        97904 |      25.49 |    3262.63 | G:\testfile.dat (200GiB)
     1 |       801382400 |        97825 |      25.47 |    3260.00 | G:\testfile.dat (200GiB)
     2 |       802217984 |        97927 |      25.50 |    3263.39 | G:\testfile.dat (200GiB)
     3 |       803012608 |        98024 |      25.52 |    3266.63 | G:\testfile.dat (200GiB)
-----------------------------------------------------------------------------
total:        3208642560 |       391680 |     101.97 |   13052.65
```

Maximum read IOPS

We use a high queue depth of 128, a small block size of four KB, and four worker threads for driving Read operations. The read workers are driving traffic on the "CacheReads" volume, which has one disk with cache set to "ReadOnly".

Run the following command for two hours of warm-up and 30 seconds of measurement:

```
diskspd -c200G -b4K -F4 -r -o128 -W7200 -d30 -Sh testfile.dat
```

Results show that the Standard_D8ds_v4 VM is delivering its maximum read IOPS limit of 77,000.

```
Total IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  file
------------------------------------------------------------------------------
     0 |      2632953856 |       642811 |      83.66 |   21417.94 | F:\testfile.dat (200GiB)
     1 |      2468286464 |       602609 |      78.43 |   20078.44 | F:\testfile.dat (200GiB)
     2 |      2044755968 |       499208 |      64.97 |   16633.21 | F:\testfile.dat (200GiB)
     3 |      2506788864 |       612009 |      79.65 |   20391.64 | F:\testfile.dat (200GiB)
------------------------------------------------------------------------------
total:        9652785152 |      2356637 |     306.72 |   78521.23

Read IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  file
------------------------------------------------------------------------------
     0 |      2632953856 |       642811 |      83.66 |   21417.94 | F:\testfile.dat (200GiB)
     1 |      2468286464 |       602609 |      78.43 |   20078.44 | F:\testfile.dat (200GiB)
     2 |      2044755968 |       499208 |      64.97 |   16633.21 | F:\testfile.dat (200GiB)
     3 |      2506788864 |       612009 |      79.65 |   20391.64 | F:\testfile.dat (200GiB)
------------------------------------------------------------------------------
total:        9652785152 |      2356637 |     306.72 |   78521.23

Write IO
thread |       bytes     |     I/Os     |    MiB/s   |  I/O per s |  file
------------------------------------------------------------------------------
     0 |               0 |            0 |       0.00 |       0.00 | F:\testfile.dat (200GiB)
     1 |               0 |            0 |       0.00 |       0.00 | F:\testfile.dat (200GiB)
     2 |               0 |            0 |       0.00 |       0.00 | F:\testfile.dat (200GiB)
     3 |               0 |            0 |       0.00 |       0.00 | F:\testfile.dat (200GiB)
------------------------------------------------------------------------------
total:                 0 |            0 |       0.00 |       0.00
```

Maximum throughput

To get the maximum read and write throughput, you can change to a larger block size of 64 KB.

FIO

FIO is a popular tool to benchmark storage on the Linux VMs. It has the flexibility to select different IO sizes, sequential or random reads and writes. It spawns worker threads or processes to perform the specified I/O operations. You can specify the type of I/O operations each worker thread must perform using job files. We created one job file per scenario illustrated in the examples below. You can change the specifications in these job files to benchmark different workloads running on Premium Storage. In the examples, we are using a Standard_D8ds_v4 running **Ubuntu**. Use the same setup described in the beginning of the benchmark section and warm up the cache before running the benchmark tests.

Before you begin, download FIO and install it on your virtual machine.

Run the following command for Ubuntu,

```
apt-get install fio
```

We use four worker threads for driving Write operations and four worker threads for driving Read operations on the disks. The write workers are driving traffic on the "nocache" volume, which has three disks with cache set to "None". The read workers are driving traffic on the "readcache" volume, which has one disk with cache set to "ReadOnly".

Maximum write IOPS

Create the job file with following specifications to get maximum Write IOPS. Name it "fiowrite.ini".

ini

```
[global]
size=30g
direct=1
iodepth=256
ioengine=libaio
bs=4k
numjobs=4

[writer1]
rw=randwrite
directory=/mnt/nocache
```

Note the follow key things that are in line with the design guidelines discussed in previous sections. These specifications are essential to drive maximum IOPS,

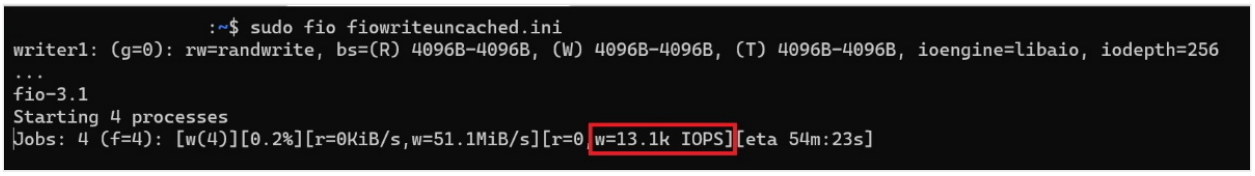
- A high queue depth of 256.
- A small block size of 4 KB.
- Multiple threads performing random writes.

Run the following command to kick off the FIO test for 30 seconds,

```
sudo fio --runtime 30 fiowrite.ini
```

While the test runs, you are able to see the number of write IOPS the VM and Premium disks are delivering. As shown in the sample below, the Standard_D8ds_v4 VM is

delivering its maximum write IOPS limit of 12,800 IOPS.



Maximum read IOPS

Create the job file with following specifications to get maximum Read IOPS. Name it "fioread.ini".

ini

```
[global]
size=30g
direct=1
iodepth=256
ioengine=libaio
bs=4k
numjobs=4

[reader1]
rw=randread
directory=/mnt/readcache
```

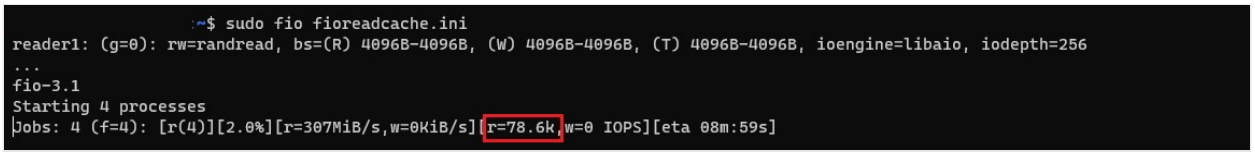
Note the follow key things that are in line with the design guidelines discussed in previous sections. These specifications are essential to drive maximum IOPS,

- A high queue depth of 256.
- A small block size of 4 KB.
- Multiple threads performing random writes.

Run the following command to kick off the FIO test for 30 seconds,

```
sudo fio --runtime 30 fioread.ini
```

While the test runs, you are able to see the number of read IOPS the VM and Premium disks are delivering. As shown in the sample below, the Standard_D8ds_v4 VM is delivering more than 77,000 Read IOPS. This is a combination of the disk and the cache performance.



Maximum read and write IOPS

Create the job file with following specifications to get maximum combined Read and Write IOPS. Name it "fioreadwrite.ini".

ini

```
[global]
size=30g
direct=1
iodepth=128
ioengine=libaio
bs=4k
numjobs=4

[reader1]
rw=randread
directory=/mnt/readcache

[writer1]
rw=randwrite
directory=/mnt/nocache
rate_iops=3200
```

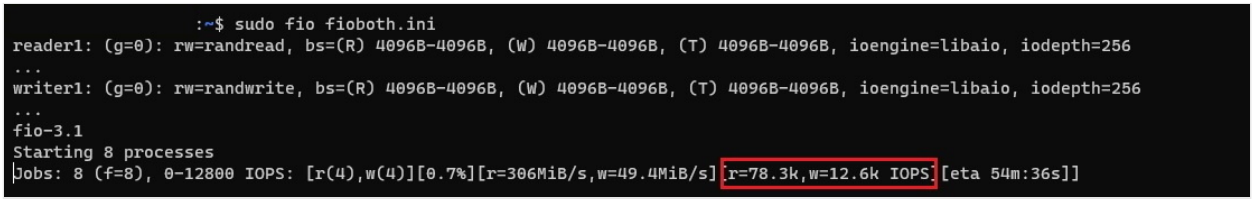
Note the follow key things that are in line with the design guidelines discussed in previous sections. These specifications are essential to drive maximum IOPS,

- A high queue depth of 128.
- A small block size of 4 KB.
- Multiple threads performing random reads and writes.

Run the following command to kick off the FIO test for 30 seconds,

```
sudo fio --runtime 30 fioreadwrite.ini
```

While the test runs, you are able to see the number of combined read and write IOPS the VM and Premium disks are delivering. As shown in the sample below, the Standard_D8ds_v4 VM is delivering more than 90,000 combined Read and Write IOPS. This is a combination of the disk and the cache performance.



Maximum combined throughput

To get the maximum combined Read and Write Throughput, use a larger block size and large queue depth with multiple threads performing reads and writes. You can use a block size of 64 KB and queue depth of 128.

Next steps

Proceed to our article on designing for high performance.

In that article, you create a checklist similar to your existing application for the prototype. Using Benchmarking tools you can simulate the workloads and measure performance on the prototype application. By doing so, you can determine which disk offering can match or surpass your application performance requirements. Then you can implement the same guidelines for your production application.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Scalability and performance targets for VM disks

Article • 12/09/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can attach a number of data disks to an Azure virtual machine (VM). Based on the scalability and performance targets for a VM's data disks, you can determine the number and type of disk that you need to meet your performance and capacity requirements.

> **Important**
>
> For optimal performance, limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all attached disks aren't highly utilized at the same time, the virtual machine can support a larger number of disks. Additionally, when creating a managed disk from an existing managed disk, only 49 disks can be created concurrently. More disks can be created after some of the initial 49 have been created.

For Azure managed disks:

The following table illustrates the default and maximum limits of the number of resources per region per subscription. The limits remain the same irrespective of disks encrypted with either platform-managed keys or customer-managed keys. There is no limit for the number of Managed Disks, snapshots and images per resource group.

Expand table

Resource	Limit
Standard managed disks	50,000
Standard SSD managed disks	50,000
Premium SSD managed disks	50,000
Premium SSD v2 managed disks	1,000
Premium SSD v2 managed disks capacity[2]	32,768
Ultra disks	1,000
Ultra disk capacity[2]	32,768
Standard_LRS snapshots[1]	75,000
Standard_ZRS snapshots[1]	75,000
Managed image	50,000

[1]An individual disk can have 500 incremental snapshots.

[2]This is the default max but higher capacities are supported by request. To request an increase in capacity, request a quota increase or contact Azure Support.

For standard storage accounts:

A Standard storage account has a maximum total request rate of 20,000 IOPS. The total IOPS across all of your virtual machine disks in a Standard storage account should not exceed this limit.

For unmanaged disks, you can roughly calculate the number of highly utilized disks supported by a single standard storage account based on the request rate limit. For example, for a Basic tier VM, the maximum number of highly utilized disks is about 66, which is 20,000/300 IOPS per disk. The maximum number of highly utilized disks for a Standard tier VM is about 40, which is 20,000/500 IOPS per disk.

For premium storage accounts:

A premium storage account has a maximum total throughput rate of 50 Gbps. The total throughput across all of your VM disks should not exceed this limit.

See VM sizes for more details.

Managed virtual machine disks

Standard HDD managed disks

Expand table

Standard Disk Type	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000
*Expanded IOPS per disk	N/A	N/A	N/A	N/A	N/A	Up to 1,500	Up to 3,000	Up to 3,000	Up to 3,000	Up to 3,000	Up to 3,000
Base throughput per disk	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 60 MB/s	Up to 300 MB/s	Up to 500 MB/s	Up to 500 MB/s
*Expanded throughput per disk	N/A	N/A	N/A	N/A	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 500 MB/s	Up to 500 MB/s	Up to 500 MB/s	Up to 500 MB/s

* Only applies to disks with performance plus (preview) enabled.

Standard SSD managed disks

Expand table

Standard SSD sizes	E1	E2	E3	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base IOPS per disk	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000
*Expanded IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Up to 1,500	Up to 3,000	Up to 6,000	Up to 6,000	Up to 6,000	Up to 6,000
Base throughput per disk	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 100 MB/s	Up to 400 MB/s	Up to 600 MB/s	Up to 750 MB/s
*Expanded throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Up to 150 MB/s	Up to 300 MB/s	Up to 600 MB/s	Up to 750 MB/s	Up to 750 MB/s	Up to 750 MB/s
Max burst IOPS per disk	600	600	600	600	600	600	600	600	1000					
Max burst throughput per disk	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	150 MB/s	250 MB/s					
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min					

* Only applies to disks with performance plus (preview) enabled.

Premium SSD managed disks: Per-disk limits

Expand table

Premium SSD sizes	P1	P2	P3	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
Base provisioned IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
**Expanded provisioned IOPS per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8,000	16,000	20,000	20,000	20,000	20,000
Base provisioned Throughput per disk	25 MB/s	25 MB/s	25 MB/s	25 MB/s	50 MB/s	100 MB/s	125 MB/s	150 MB/s	200 MB/s	250 MB/s	250 MB/s	500 MB/s	750 MB/s	900 MB/s
**Expanded provisioned throughput per disk	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	300 MB/s	600 MB/s	900 MB/s	900 MB/s	900 MB/s	900 MB/s
Max burst IOPS per disk	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500	30,000*	30,000*	30,000*	30,000*	30,000*	30,000*
Max burst throughput per disk	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	170 MB/s	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*	1,000 MB/s*
Max burst duration	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*	Unlimited*
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year

*Applies only to disks with on-demand bursting enabled.

** Only applies to disks with performance plus (preview) enabled.

Premium SSD managed disks: Per-VM limits

Expand table

Resource	Limit
Maximum IOPS Per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/s with GS5 VM

Unmanaged virtual machine disks

Standard unmanaged virtual machine disks: Per-disk limits

Expand table

VM tier	Basic tier VM	Standard tier VM
Disk size	4,095 GB	4,095 GB

VM tier	Basic tier VM	Standard tier VM
Maximum 8-KB IOPS per persistent disk	300	500
Maximum number of disks that perform the maximum IOPS	66	40

Premium unmanaged virtual machine disks: Per-account limits

Expand table

Resource	Limit
Total disk capacity per account	35 TB
Total snapshot capacity per account	10 TB
Maximum bandwidth per account (ingress + egress)[1]	<=50 Gbps

[1]*Ingress* refers to all data from requests that are sent to a storage account. *Egress* refers to all data from responses that are received from a storage account.

Premium unmanaged virtual machine disks: Per-disk limits

Expand table

Premium storage disk type	P10	P20	P30	P40	P50
Disk size	128 GiB	512 GiB	1,024 GiB (1 TB)	2,048 GiB (2 TB)	4,095 GiB (4 TB)
Maximum IOPS per disk	500	2,300	5,000	7,500	7,500
Maximum throughput per disk	100 MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec
Maximum number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: Per-VM limits

Expand table

Resource	Limit
Maximum IOPS per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/sec with GS5 VM

See also

Azure subscription and service limits, quotas, and constraints

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an incremental snapshot for managed disks

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Incremental snapshots are point-in-time backups for managed disks that, when taken, consist only of the changes since the last snapshot. The first incremental snapshot is a full copy of the disk. The subsequent incremental snapshots occupy only delta changes to disks since the last snapshot. When you restore a disk from an incremental snapshot, the system reconstructs the full disk that represents the point in time backup of the disk when the incremental snapshot was taken. This capability for managed disk snapshots potentially allows them to be more cost-effective, since, unless you choose to, you don't have to store the entire disk with each individual snapshot. Just like full snapshots, incremental snapshots can be used to either create a full managed disk or a full snapshot. Both full snapshots and incremental snapshots can be used immediately after being taken. In other words, once you take either snapshot, you can immediately read the underlying data and use it to restore disks.

There are a few differences between an incremental snapshot and a full snapshot. Incremental snapshots will always use Standard HDD storage, irrespective of the storage type of the disk, whereas full snapshots can use Premium SSDs. If you're using full snapshots on Premium Storage to scale up VM deployments, we recommend you use custom images on standard storage in the Azure Compute Gallery. It will help you achieve a more massive scale with lower cost. Additionally, incremental snapshots potentially offer better reliability with zone-redundant storage (ZRS). If ZRS is available in the selected region, an incremental snapshot will use ZRS automatically. If ZRS isn't available in the region, then the snapshot will default to locally redundant storage (LRS). You can override this behavior and select one manually but, we don't recommend that.

Incremental snapshots are billed for the used size only. You can find the used size of your snapshots by looking at the Azure usage report. For example, if the used data size of a snapshot is 10 GiB, the **daily** usage report will show 10 GiB/(31 days) = 0.3226 as the consumed quantity.

Restrictions

- Incremental snapshots currently can't be moved between subscriptions.
- You can currently only generate SAS URIs of up to five snapshots of a particular snapshot family at any given time.

- You can't create an incremental snapshot for a particular disk outside of that disk's subscription.
- Incremental snapshots can't be moved to another resource group. But, they can be copied to another resource group or region.
- Up to seven incremental snapshots per disk can be created every five minutes.
- A total of 500 incremental snapshots can be created for a single disk. The 500 quota limit isn't over the lifetime of a disk, but at any given point in time. You can always delete older snapshots of a disk to make room for newer snapshots.completion
- You can't get the changes between snapshots taken before and after you changed the size of the parent disk across 4-TB boundary. For example, You took an incremental snapshot `snapshot-a` when the size of a disk was 2 TB. Now you increased the size of the disk to 6 TB and then took another incremental snapshot `snapshot-b`. You can't get the changes between `snapshot-a` and `snapshot-b`. You have to download the full copy of `snapshot-b` created after the resize. Subsequently, you can get the changes between `snapshot-b` and snapshots created after `snapshot-b`.
- When you create a managed disk from a snapshot, it starts a background copy process. You can attach a disk to a VM while this process is running but you'll experience performance impact. You can use `CompletionPercent` property to check the status of the background copy only for Premium SSD v2 and Ultra Disk. You can't use the `CompletionPercent` property to gauge the progress of the background data copy for snapshots created from Premium SSD, Standard SSD, and Standard HDDs, or the disk hydration process from snapshots.

Incremental snapshots of Premium SSD v2 and Ultra Disks

Incremental snapshots of Premium SSD v2 and Ultra Disks have the following extra restrictions:

- Snapshots with a 512 logical sector size are stored as VHD, and can be used to create any disk type. Snapshots with a 4096 logical sector size are stored as VHDX and can only be used to create Ultra Disks and Premium SSD v2 disks, they can't be used to create other disk types. To determine which sector size your snapshot is, see check sector size.
- Up to five disks may be simultaneously created from a snapshot of a Premium SSD v2 or an Ultra Disk.
- When an incremental snapshot of either a Premium SSD v2 or an Ultra Disk is created, a background copy process for that disk is started. While a background copy is ongoing, you can have up to three total snapshots pending. The process must complete before any more snapshots of that disk can be created.
- Incremental snapshots of a Premium SSD v2 or an Ultra Disk can't be used immediately after they're created. The background copy must complete before you can create a disk from the snapshot. See Check snapshot status for details.

- When you increase the size of a Premium SSD v2 or an Ultra Disk, any incremental snapshots that are under background copy will fail.
- Encryption settings can't be updated for Premium SSD v2 and Ultra Disk snapshot where background data copy is in progress.
- When you attach a Premium SSD v2 or Ultra Disk created from snapshot to a running Virtual Machine while `CompletionPercent` property hasn't reached 100, the disk suffers performance impact. Specifically, if the disk has a 4k sector size, it may experience slower read. If the disk has a 512e sector size, it may experience slower read and write. To track the progress of this background copy process, see the check disk status section of either the Azure PowerShell sample or the Azure CLI.
- You can have up to 50 currently active background copies per subscription. The following activities count against the total number of active background copies:
  - Creating Premium SSD v2 and Ultra Disks with snapshots
  - Creating Premium SSD v2 and Ultra Disks with restore points
  - Uploading VHDX files into Premium SSD v2 and Ultra Disks
  - Converting existing Premium SSDs to Premium SSD v2 disks.

> **Note**
>
> Normally, when you take an incremental snapshot, and there aren't any changes, the size of that snapshot is 0 MiB. Currently, empty snapshots of disks with a 4096 logical sector size instead have a size of 6 MiB, when they'd normally be 0 MiB.

Create incremental snapshots

Azure CLI

You can use the Azure CLI to create an incremental snapshot. You need the latest version of the Azure CLI. See the following articles to learn how to either install or update the Azure CLI.

The following script creates an incremental snapshot of a particular disk:

Azure CLI

```
# Declare variables
diskName="yourDiskNameHere"
resourceGroupName="yourResourceGroupNameHere"
snapshotName="desiredSnapshotNameHere"

# Get the disk you need to backup
yourDiskID=$(az disk show -n $diskName -g $resourceGroupName --query "id" --
```

```
output tsv)

# Create the snapshot
az snapshot create -g $resourceGroupName -n $snapshotName --source $yourDiskID
--incremental true
```

You can identify incremental snapshots from the same disk with the `SourceResourceId` property of snapshots. `SourceResourceId` is the Azure Resource Manager resource ID of the parent disk.

You can use `SourceResourceId` to create a list of all snapshots associated with a particular disk. Replace `yourResourceGroupNameHere` with your value and then you can use the following example to list your existing incremental snapshots:

Azure CLI

```
# Declare variables and create snapshot list
subscriptionId="yourSubscriptionId"
resourceGroupName="yourResourceGroupNameHere"
diskName="yourDiskNameHere"

az account set --subscription $subscriptionId

diskId=$(az disk show -n $diskName -g $resourceGroupName --query [id] -o tsv)

az snapshot list --query "[?creationData.sourceResourceId=='$diskId' &&
incremental]" -g $resourceGroupName --output table
```

Check snapshot status

Incremental snapshots of Premium SSD v2 or Ultra Disks can't be used to create new disks until the background process copying the data into the snapshot has completed.

You can use either the CLI or PowerShell sections to check the status of the background copy from a disk to a snapshot.

> **ⓘ Important**
>
> You can't use the following sections to get the status of the background copy process for disk types other than Ultra Disk or Premium SSD v2. Snapshots of other disk types always report 100%.

CLI

You have two options for getting the status of snapshots. You can either get a list of all incremental snapshots associated with a specific disk, and their respective status, or you can get the status of an individual snapshot.

CLI - List incremental snapshots

The following script returns a list of all snapshots associated with a particular disk. The value of the `CompletionPercent` property of any snapshot must be 100 before it can be used. Replace `yourResourceGroupNameHere`, `yourSubscriptionId`, and `yourDiskNameHere` with your values then run the script:

Azure CLI

```
# Declare variables and create snapshot list
subscriptionId="yourSubscriptionId"
resourceGroupName="yourResourceGroupNameHere"
diskName="yourDiskNameHere"
az account set --subscription $subscriptionId
diskId=$(az disk show -n $diskName -g $resourceGroupName --query [id] -o tsv)
az snapshot list --query "[?creationData.sourceResourceId=='$diskId' &&
incremental]" -g $resourceGroupName --output table
```

CLI - Individual snapshot

You can also check the status of an individual snapshot by checking the `CompletionPercent` property. Replace `$sourceSnapshotName` with the name of your snapshot then run the following command. The value of the property must be 100 before you can use the snapshot for restoring disk or generate a SAS URI for downloading the underlying data.

Azure CLI

```
az snapshot show -n $sourceSnapshotName -g $resourceGroupName --query
[completionPercent] -o tsv
```

PowerShell

You have two options for getting the status of snapshots. You can either get a list of all incremental snapshots associated with a particular disk and their respective status, or you can get the status of an individual snapshot.

PowerShell - List incremental snapshots

The following script returns a list of all incremental snapshots associated with a particular disk that haven't completed their background copy. Replace `yourResourceGroupNameHere` and `yourDiskNameHere`, then run the script.

Azure PowerShell

```
$resourceGroupName = "yourResourceGroupNameHere"
$snapshots = Get-AzSnapshot -ResourceGroupName $resourceGroupName
$diskName = "yourDiskNameHere"
$yourDisk = Get-AzDisk -DiskName $diskName -ResourceGroupName $resourceGroupName
$incrementalSnapshots = New-Object System.Collections.ArrayList
foreach ($snapshot in $snapshots)
{
    if($snapshot.Incremental -and $snapshot.CreationData.SourceResourceId -eq
$yourDisk.Id -and $snapshot.CreationData.SourceUniqueId -eq $yourDisk.UniqueId)
    {
    $targetSnapshot=Get-AzSnapshot -ResourceGroupName $resourceGroupName -
SnapshotName $snapshotName
        {
        if($targetSnapshot.CompletionPercent -lt 100)
            {
            $incrementalSnapshots.Add($targetSnapshot)
            }
        }
    }
}
$incrementalSnapshots
```

PowerShell - individual snapshots

You can check the `CompletionPercent` property of an individual snapshot to get its status. Replace `yourResourceGroupNameHere` and `yourSnapshotName` then run the script. The value of the property must be 100 before you can use the snapshot for restoring disk or generate a SAS URI for downloading the underlying data.

Azure PowerShell

```
$resourceGroupName = "yourResourceGroupNameHere"
$snapshotName = "yourSnapshotName"
$targetSnapshot=Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName
$snapshotName
$targetSnapshot.CompletionPercent
```

Check sector size

Snapshots with a 4096 logical sector size can only be used to create Premium SSD v2 or Ultra Disks. They can't be used to create other disk types. Snapshots of disks with 4096 logical sector size are stored as VHDX, whereas snapshots of disks with 512 logical sector size are stored as VHD. Snapshots inherit the logical sector size from the parent disk.

To determine whether or your Premium SSD v2 or Ultra Disk snapshot is a VHDX or a VHD, get the `LogicalSectorSize` property of the snapshot.

The following command displays the logical sector size of a snapshot:

Azure CLI

```
az snapshot show -g resourcegroupname -n snapshotname --query
[creationData.logicalSectorSize] -o tsv
```

Next steps

See the following articles to create disks from your snapshots using the Azure CLI or Azure PowerShell module.

See Copy an incremental snapshot to a new region to learn how to copy an incremental snapshot across regions.

If you have more questions on snapshots, see the snapshots section of the FAQ.

If you'd like to see sample code demonstrating the differential capability of incremental snapshots, using .NET, see Copy Azure Managed Disks backups to another region with differential capability of incremental snapshots.

Last updated on 07/02/2025

Instant access snapshots for Azure Managed Disks

Summarize this article for me

Azure Managed Disk snapshots provide point-in-time backups of disks that can be used as backup during software upgrades, disaster recovery, or to create new environments. When creating snapshots from Azure Managed Disks, Azure automatically copies the data from the disk to the snapshot in the background.

Snapshots of Premium SSD, Standard SSD, and Standard HDDs are by default instant access. Immediately upon creation, these snapshots can be used to restore new disks, download underlying data, and copy to other Azure regions.

Snapshots of Ultra Disks and Premium SSD v2 aren't instant access by default and require the background data copy to complete before they can be used. To create instant access snapshots, you can opt in to enable instant access using the existing snapshot API at the time of snapshot creation.

Snapshots of Premium SSD, Standard SSD, and Standard HDDs

Snapshots of Premium SSD, Standard SSD, and Standard HDDs are instant access by default. As soon as snapshots are created, you can immediately use these snapshots to create new disks of any supported disk types, generate SAS URIs to download data, or copy snapshots to other Azure regions for regional disaster recovery. After snapshot creation, Azure automatically initiates a background data copy from the source disk to snapshot.

Disks created from snapshots of Premium SSD, Standard SSD, and Standard HDD can be immediately attached to running virtual machines. During disk creation, Azure automatically initiates a background data copy to hydrate the disk from snapshot data. During this process, the disk may experience temporary performance degradation until the background copy completes. To reduce the performance impact, you can create a full snapshot on Premium Storage and restore the disk from that snapshot.

Snapshots of Ultra Disk and Premium SSD v2

Snapshots of Ultra Disk and Premium SSD v2 aren't instant access by default and can only be used after snapshot's background data copy completes. To bypass this, you can choose to configure instant access by specifying a value in the `InstantAccessDurationMins` parameter at

the time of snapshot creation. This will allow you to create an instant access snapshot and can be immediately used to create new disks. Disks created from instant access snapshots are rapidly hydrated with minimal performance impact and can be immediately attached to a running VM.

After the time specified in `InstantAccessDurationMins` elapses, the snapshot automatically transitions to a Standard Storage snapshot and can be used once the background data copy has completed. You can monitor the snapshot's state via the `SnapshotAccessState` property.

Until the data is fully copied to Standard Storage, snapshots in the InstantAccess state depend on the availability of the source disk and do not provide protection against disk or zonal failures. To ensure durability and protection, the snapshot must complete its background data copy.

Until the data is fully copied to Standard Storage, snapshots in the **InstantAccess** state depend on the availability of the source disk and don't provide protection against disk or zonal failures. To ensure zonal redundancy, the snapshot must complete its background data copy to Standard ZRS. You can monitor a snapshot's background data copy progress by checking SnapshotAccessState property.

Limitations

- Only Ultra Disks and Premium SSD v2 can be created from instant access snapshots of Ultra Disks and Premium SSD v2 disks
- `InstantAccessDurationMins` must be between 60 and 300 minutes
- Instant access snapshots count towards the Ultra Disk and Premium SSD v2 limit of three in-progress snapshots per disk
- You can create up to 15 disks concurrently from all instant access snapshots of an individual disk
- You can't use an instant access snapshot to create an Ultra Disk or a Premium SSD v2 larger than the snapshot's size
- If a Ultra Disk or Premium SSD v2 is being hydrated from a snapshot, you can't create an instant access snapshot of that disk
  - Check the `CompletionPercent` property of the disk, if it's below 100 then it's currently being hydrated
- Instant access snapshots of Ultra Disk or Premium SSD v2 can't be copied across regions and the underlying data can't be downloaded until the background data copy completes
  - Check the `CompletionPercent` property on the snapshot, when it reaches 100 then it can be copied across regions and the underlying data can be downloaded
- The encryption property of a disk created from an instant access snapshot can't be updated during disk hydration and you can't update the encryption settings for Ultra

Disks and Premium SSD v2 that currently have active instant access snapshots
- Attaching Ultra Disks and Premium SSD v2 across fault domains (using either a VM in an availability set or a Virtual Machine Scale Set) triggers the background data copy and prevents you from creating an instant access snapshot during the background data copy.
- Ultra Disk and Premium SSD v2 that have active instant access snapshots can't be attached across fault domains
- To create instant access snapshots from Ultra Disks, the snapshot must be created from a newly provisioned Ultra Disks.
- The greatest read latency improvements for disks created from instant access snapshots are currently available in Germany West Central, East Asia, Southeast Asia, Central India, and Sweden Central.

Regional availability

Instant access snapshots are currently supported in all pulic regions.

Billing of instant access snapshots of Ultra Disks and Premium SSD v2 disks

Instant access snapshots use a usage-based billing model with two components: a storage charge and a one-time restore fee.

- Storage charge: You are billed only for the additional storage consumed by an instant access snapshot during its active lifetime. When a snapshot is first created, it starts at zero additional cost, as it references the source disk as its base. As data on the source disk changes or is deleted over time, the snapshot preserves the original point-in-time state, and its used size grows accordingly. This means you pay only for incremental changes of instant access snapshot, not for a full copy of the disk.
- Restore charge: Each time you restore a disk from an instant access snapshot, a one-time restore operation fee is applied. This fee is calculated based on the provisioned size of the disk at the time of restore, providing predictable billing for restore operations.

Learn more about Instant Access Snapshot billing in here.

Create an instant access snapshot

Instant access snapshots of Ultra Disk and Premium SSD v2 are not a separate snapshot resource class to manage. They are incremental snapshots that temporarily enter Instant Access state for the specified duration. When this duration expires, snapshots automatically transition out of the Instant Access state and continue as Standard ZRS snapshots for better reliability and long-term retention.

To create instant access snapshots of Ultra Disk and Premium SSD v2, you use the same Snapshot API and commands, but add an additional parameter that specifies the how long the snapshot remains in the instant access state (minimum 60 minutes, maximum 300 minutes). After the specified duration expires, the snapshot automatically transitions to a Standard Storage snapshot. You can monitor your snapshot's state by checking the snapshot's access state.

In the Azure portal, you can't specify a duration, so snapshots created through the portal remain instant access snapshots for 300 minutes (5 hours).

Azure CLI

Azure CLI

```
# Declare variables

subscriptionId="yourSubscriptionId"
diskName="yourDiskName"
resourceGroupName="yourResourceGroupName"
snapshotName="desiredInstantAaccessSnapshotName"

# Set Subscription Id

az account set --subscription $subscriptionId

# Get the disk you need to create an instant access snapshot

yourDiskID=$(az disk show -n $diskName -g $resourceGroupName --query "id" --
output tsv)

# Create an instant access snapshot

az snapshot create -g $resourceGroupName -n $snapshotName --source $yourDiskID
--incremental true --location eastus2euap  --sku Standard_ZRS --ia-duration 300
```

Check snapshot access state

You can monitor Managed Disk Snapshot across different states using the `SnapshotAccessState` property of the snapshot resource. This property indicates the current access state of a snapshot, helping you understand its readiness for operations. Here are the states and what they imply:

Expand table

State	Description	Applies To
Pending	This snapshot can't be used to create a new disk, download data, or copy to another region.	Incremental snapshots from Ultra Disks and Premium SSD v2 disks during background data copy; snapshots being copied across regions.
Available	This snapshot can be used to create a new disk (with a performance impact), download data, or copy to another region.	Snapshots of Premium SSD, Standard SSD, and Standard HDDs; incremental snapshots for Ultra Disks and Premium SSD v2 disks after background copy completes; snapshots copied within the same region using shallow copy.
InstantAccess	This snapshot can be used to create rapidly hydrated disk with minimal performance impact, but the underlying data can't be downloaded and it can't be copied to another region.	Instant access snapshots for Ultra Disks and Premium SSD v2 disks when the instant access duration hasn't lapsed and the background data copy is ongoing.
AvailableWithInstantAccess	This snapshot can be used to create rapidly hydrated disk with minimal performance impact, the underlying data can be downloaded, and it can be copied to another region.	Instant access snapshots for Ultra Disks and Premium SSD v2 disks when the instant access duration hasn't lapsed and the background data copy completes.

CLI

Azure CLI

```
snapshotName="DesiredInstantAccessSnapshotTestName"
resourceGroupName="yourResourceGroupName"

snapshotId=$(az snapshot show --name $snapshotName --resource-group
$resourceGroupName --query [id] -o tsv)

az resource show --ids $snapshotId --query "properties.snapshotAccessState" --
output tsv

az resource show --ids $snapshotId --query
"properties.creationData.instantAccessDurationMinutes" --output tsv
```

Last updated on 02/17/2026

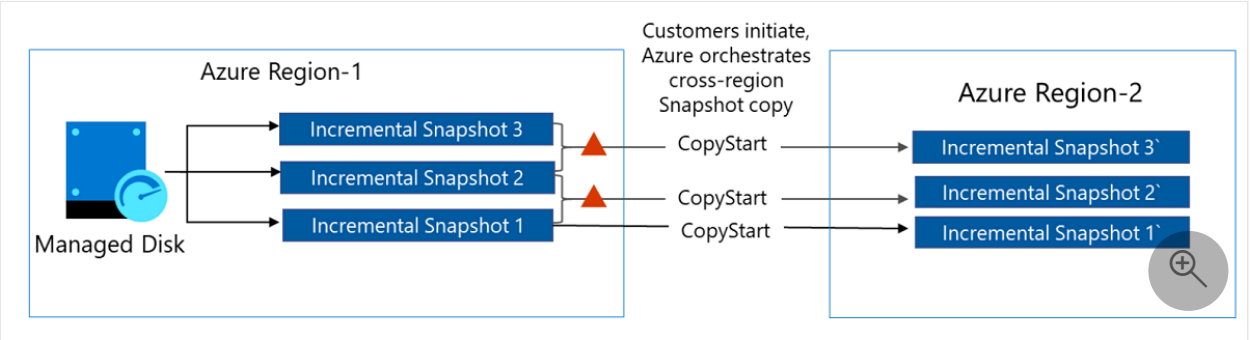
Copy an incremental snapshot to a new region

Article • 08/22/2024

There are two options for copying an incremental snapshot across regions. The first option, a managed process (recommended), that will perform the copy for you. This process is handled by Azure and removes the maintenance overhead of managing the copy process by staging a storage account in the target region. Azure ensures that only changes since the last snapshot in the target region are copied to the target region to reduce the data footprint, reducing the recovery point objective. You can check the process of a copy so you know when a target snapshot is ready to restore disks. For this managed process, you're only billed for the bandwidth cost of the data transfer across the region, and the read transactions on the source snapshot. Don't delete your source snapshot while the target snapshot is being copied.

The second option is a manual copy, where you get the changes between two incremental snapshots, down to the block level, and manually copy it from one region to another. Most users should use the managed process but, if you're interested in improving the copy speed, the second option allows you to use your compute resources to make the copy faster.

This article covers copying an incremental snapshot from one region to another. See Create an incremental snapshot for managed disks for conceptual details on incremental snapshots.



Restrictions

- You can copy 100 incremental snapshots in parallel at the same time per subscription per region.
- If you use the REST API, you must use version 2020-12-01 or newer of the Azure Compute REST API.
- You can only copy one incremental snapshot of a particular disk at a time.

- Snapshots must be copied in the order they were created.
- Only incremental snapshots can be copied across regions. Full snapshots can't be copied across regions.

Managed copy

Azure CLI

You can use the Azure CLI to copy an incremental snapshot. You need the latest version of the Azure CLI. See the following articles to learn how to either install or update the Azure CLI.

The following script copies an incremental snapshot from one region to another:

Azure CLI

```
subscriptionId=<yourSubscriptionID>
resourceGroupName=<yourResourceGroupName>
targetSnapshotName=<name>
sourceSnapshotName=<name>
targetRegion=<validRegion>

sourceSnapshotId=$(az snapshot show -n $sourceSnapshotName -g
$resourceGroupName --query [id] -o tsv)

az snapshot create -g $resourceGroupName -n $targetSnapshotName -l
$targetRegion --source $sourceSnapshotId --incremental --copy-start
```

Check copy status

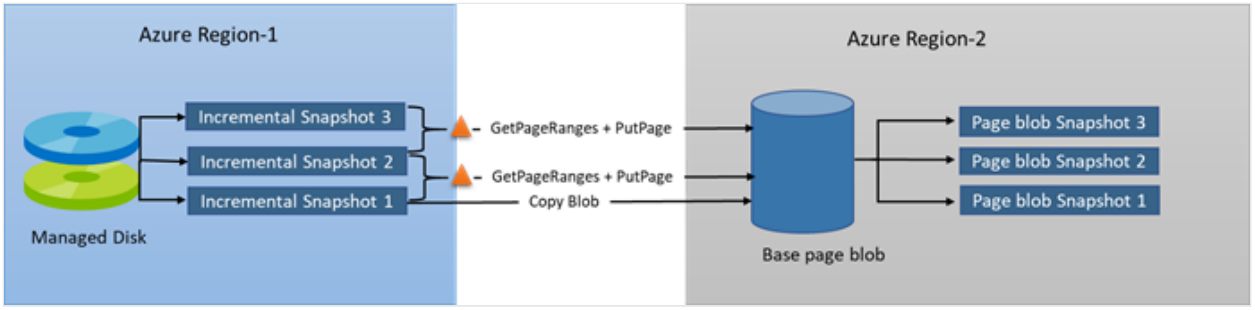
You can check the status of an individual snapshot by checking the `CompletionPercent` property. Replace `$sourceSnapshotName` with the name of your snapshot then run the following command. The value of the property must be 100 before you can use the snapshot for restoring disk or generate a SAS URI for downloading the underlying data.

Azure CLI

```
az snapshot show -n $sourceSnapshotName -g $resourceGroupName --query
[completionPercent] -o tsv
```

Manual copy

Incremental snapshots offer a differential capability. They enable you to get the changes between two incremental snapshots of the same managed disk, down to the block level. You can use this to reduce your data footprint when copying snapshots across regions. For example, you can download the first incremental snapshot as a base blob in another region. For the subsequent incremental snapshots, you can copy only the changes since the last snapshot to the base blob. After copying the changes, you can take snapshots on the base blob that represent your point in time backup of the disk in another region. You can restore your disk either from the base blob or from a snapshot on the base blob in another region.



Next steps

If you'd like to see sample code demonstrating the differential capability of incremental snapshots, using .NET, see Copy Azure Managed Disks backups to another region with differential capability of incremental snapshots.

If you have additional questions on snapshots, see the snapshots section of the FAQ.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Overview of Azure Disk Backup

Summarize this article for me

Azure Disk Backup is a native, cloud-based backup solution that protects your data in managed disks. It's a simple, secure, and cost-effective solution that enables you to configure protection for managed disks in a few steps. It assures that you can recover your data in a disaster scenario.

Azure Disk Backup offers a turnkey solution that provides snapshot lifecycle management for managed disks by automating periodic creation of snapshots and retaining it for configured duration using backup policy. You can manage the disk snapshots with zero infrastructure cost and without the need for custom scripting or any management overhead. This is a crash-consistent backup solution that takes point-in-time backup of a managed disk using incremental snapshots with support for multiple backups per day. It's also an agent-less solution and doesn't impact production application performance. It supports backup and restore of both OS and data disks (including shared disks), whether or not they're currently attached to a running Azure virtual machine.

If you require application-consistent backup of virtual machine including the data disks, or an option to restore an entire virtual machine from backup, restore a file or folder, or restore to a secondary region, then use the Azure VM backup solution. Azure Backup offers side-by-side support for backup of managed disks using Disk Backup in addition to Azure VM backup solutions. This is useful when you need once-a-day application consistent backups of virtual machines and also more frequent backups of OS disks or a specific data disk that are crash consistent, and don't impact the production application performance.

Azure Disk Backup is integrated into Resiliency, which provides a **single unified management experience** in Azure for enterprises to govern, monitor, operate, and analyze backups at scale.

> **Note**
>
> - If the target disk is attached as a Persistent Volume to an AKS cluster, choose Azure Backup for AKS over the standalone Disk Backup solution. It enables backing up the disk as snapshots along with the containerized application in a Kubernetes-aware manner, all as a single unit. Additionally, you get Cross Region Restore and ransomware protection capabilities with AKS Backup.

Key benefits of Disk Backup

Azure Disk backup is an agentless and crash consistent solution that uses incremental snapshots and offers the following advantages:

- More frequent and quick backups without interrupting the virtual machine.
- Doesn't affect the performance of the production application.
- No security concerns since it doesn't require running custom scripts or installing agents.
- A cost-effective solution to back up specific disks when compared to backing up entire virtual machine.

Azure Disk backup solution is useful in the following scenarios:

- Need for frequent backups per day without application being quiescent.
- Apps running in a cluster scenario: both Windows Server Failover Cluster and Linux clusters are writing to shared disks.
- Specific need for agentless backup because of security or performance concerns on the application.
- Application consistent backup of VM isn't feasible since line-of-business apps don't support Volume Shadow Copy Service (VSS).

Consider Azure Disk Backup in scenarios where:

- A mission-critical application is running on an Azure Virtual machine that demands multiple backups per day to meet the recovery point objective, but without impacting the production environment or application performance.
- Your organization or industry regulation restricts installing agents because of security concerns.
- Executing custom pre or post scripts and invoking freeze and thaw on Linux virtual machines to get application-consistent backup puts undue overhead on production workload availability.
- Containerized applications running on Azure Kubernetes Service (AKS cluster) are using managed disks as persistent storage. Today, you must back up the managed disk via automation scripts that are hard to manage.
- A managed disk is holding critical business data, used as a file-share, or contains database backup files, and you want to optimize backup cost by not investing in Azure VM backup.
- You have many Linux and Windows single-disk virtual machines (that is, a virtual machine with just an OS disk and no data disks attached) that host web server, state-less machines, or serves as a staging environment with application configuration settings, and you need a cost efficient backup solution to protect the OS disk. For example, to trigger a quick on-demand backup before upgrading or patching the virtual machine.
- A virtual machine is running an OS configuration that is unsupported by Azure VM backup solution (for example, Windows 2008 32-bit Server).

How the backup and restore process works

- The first step in configuring backup for Azure Managed Disks is creating a Backup vault. The vault gives you a consolidated view of the backups configured across different workloads. Azure Disk backup supports only Operational Tier backup. Copying of backups to the vault storage tier is not supported. So, the Backup vault storage redundancy setting (LRS/GRS) doesn't apply to the backups stored in Operational Tier.

- Then create a Backup policy that allows you to configure backup frequency and retention duration.

- To configure backup, go to the Backup vault, assign a backup policy, select the managed disk that needs to be backed up and provide a resource group where the snapshots are to be stored and managed. Azure Backup automatically triggers scheduled backup jobs that create an incremental snapshot of the disk according to the backup frequency. Older snapshots are deleted according to the retention duration specified by the backup policy.

- Azure Backup uses incremental snapshots of the managed disk. Incremental snapshots are a cost-effective, point-in-time backup of managed disks that are billed for the delta changes to the disk since the last snapshot. These are always stored on the most cost-effective storage, standard HDD storage regardless of the storage type of the parent disks. The first snapshot of the disk will occupy the used size of the disk, and consecutive incremental snapshots store delta changes to the disk since the last snapshot. Azure Backup automatically assigns tag to the snapshots it creates to uniquely identify them.

- Once you configure the backup of a managed disk, a backup instance will be created within the backup vault. Using the backup instance, you can find health of backup operations, trigger on-demand backups, and perform restore operations. You can also view health of backups across multiple vaults and backup instances using Resiliency, which provides a single pane of glass view.

- To restore, just select the recovery point from which you want to restore the disk. Provide the resource group where the restored disk is to be created from the snapshot. Azure Backup provides an instant restore experience since the snapshots are stored locally in your subscription.

- Backup Vault uses Managed Identity to access other Azure resources. To configure backup of a managed disk and to restore from past backup, Backup Vault's managed identity requires a set of permissions on the source disk, the snapshot resource group where snapshots are created and managed, and the target resource group where you want to restore the backup. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). Managed identity is a service principal of a

special type that may only be used with Azure resources. Learn more about Managed Identities.

- Currently Azure Disk Backup supports operational backup of managed disks and doesn't copy the backups to Backup Vault storage. Refer to the support matrix for a detailed list of supported and unsupported scenarios, and region availability.

How does the disk backup scheduling and retention period work?

Azure Disk Backup currently supports only the Operational Tier, which helps store backups as Disk Snapshots in your tenant that aren't moved to the vault. The backup policy defines the schedule and retention period of your backups in the Operational Tier (when the snapshots will be taken and how long they will be retained).

By using the Azure Disk backup policy, you can define the backup schedule with Hourly frequency of 1, 2, 4, 6, 8, or 12 hours and Daily frequency. Although backups have scheduled timing as per the policy, there can be a difference in the actual start time of the backups from the scheduled one.

The retention period of snapshots is governed by the snapshot limit for a disk. Currently, a maximum of 500 snapshots can be retained for a disk. If the limit is reached, no new snapshots can be taken, and you need to delete the older snapshots.

The retention period for a backup also follows the maximum limit of 450 snapshots with 50 snapshots kept aside for on-demand backups.

For example, if the scheduling frequency for backups is set as Daily, then you can set the retention period for backups at a maximum value of 450 days. Similarly, if the scheduling frequency for backups is set as Hourly with a 1-hour frequency, then you can set the retention for backups at a maximum value of 18 days.

> **Note**
>
> - For Azure Disks belonging to Standard HDD, Standard SSD, and Premium SSD SKUs, you can define the backup schedule with *Hourly* frequency (of 1, 2, 4, 6, 8, or 12 hours) and *Daily* frequency.
> - For Azure Disks belonging to Premium V2 and Ultra Disk SKUs, you can define the backup schedule with *Hourly* frequency of only 12 hours and *Daily* frequency.

Why do I see more snapshots than my retention policy?

If a retention policy is set as *1*, you can find two snapshots. This configuration ensures that at least one latest recovery point is always present in the vault, if all subsequent backups fail due to any issue. This causes the presence of two snapshots.

So, if the policy is for *n* snapshots, you can find *n+1* snapshots at times. Further, you can even find *n+1+2* snapshots if there is a delay in deletion of recovery points whose retention period is over (garbage collection). This can happen at rare times when:

- You clean up snapshots, which are past retentions.
- The garbage collector (GC) in the backend is under heavy load.

Pricing

Azure Backup uses incremental snapshots of the managed disk. Incremental snapshots are charged per GiB of the storage occupied by the delta changes since the last snapshot. For example, if you're using a managed disk with a provisioned size of 128 GiB, with 100 GiB used, the first incremental snapshot is billed only for the used size of 100 GiB. 20 GiB of data is added on the disk before you create the second snapshot. Now, the second incremental snapshot is billed for only 20 GiB.

Incremental snapshots are always stored on standard storage, irrespective of the storage type of parent-managed disks, and are charged based on the pricing of standard storage. For example, incremental snapshots of a Premium SSD-Managed Disk are stored on standard storage. By default, they are stored on ZRS in regions that support ZRS. Otherwise, they are stored on locally redundant storage (LRS). The per GiB pricing of both the options, LRS and ZRS, is the same.

The snapshots created by Azure Backup are stored in the resource group within your Azure subscription and incur Snapshot Storage charges. For more details about the snapshot pricing, see Managed Disk Pricing. Because the snapshots aren't copied to the Backup Vault, Azure Backup doesn't charge a Protected Instance fee and Backup Storage cost doesn't apply.

The number of recovery points is determined by the Backup policy used to configure backups of the disk backup instances. Older block blobs are deleted according to the garbage collection process as the corresponding older recovery points are pruned.

Next steps

Azure Disk Backup support matrix.

Related content

- Create a Backup policy for Azure Managed Disk using REST API.
- Configure backup for Azure Managed Disk using REST API.
- Restore Azure Managed Disk using REST API.

Last updated on 01/30/2026

Back up Azure Disks using Azure portal

Summarize this article for me

This article describes how to back up Azure Disk using the Azure portal. You can also use REST API to create a Backup policy and configure backup for Azure Managed Disk.

To view the supported Azure Disk backup and restore scenarios, region availability, and limitations, see the support matrix. For common questions, see the frequently asked questions.

> **Note**
>
> If the target disk is attached as a Persistent Volume to an AKS cluster, choose Azure Backup for AKS over the standalone Disk Backup solution. It enables backing up the disk as snapshots along with the containerized application in a Kubernetes-aware manner, all as a single unit. Additionally, you get Cross Region Restore and ransomware protection capabilities with AKS Backup.

Create a Backup vault

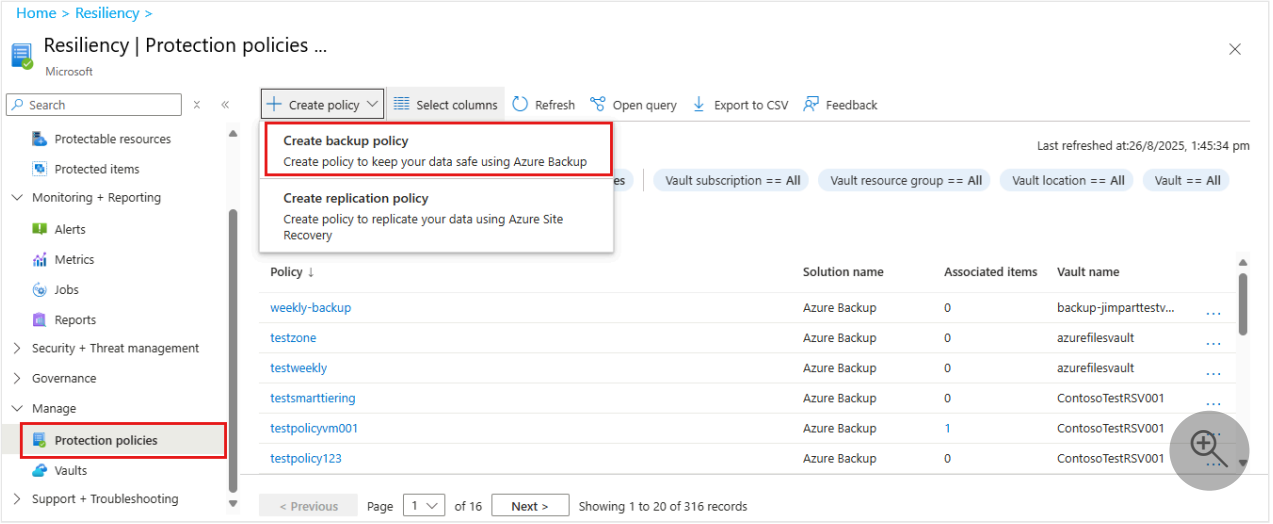
A Backup vault is a storage entity in Azure that holds backup data for various newer workloads that Azure Backup supports, such as Azure Database for PostgreSQL servers and Azure Disks. Backup vaults make it easy to organize your backup data, while minimizing management overhead. Backup vaults are based on the Azure Resource Manager model of Azure, which provides enhanced capabilities to help secure backup data.

Learn how to create a Backup vault.

Create backup policy for Azure Disks

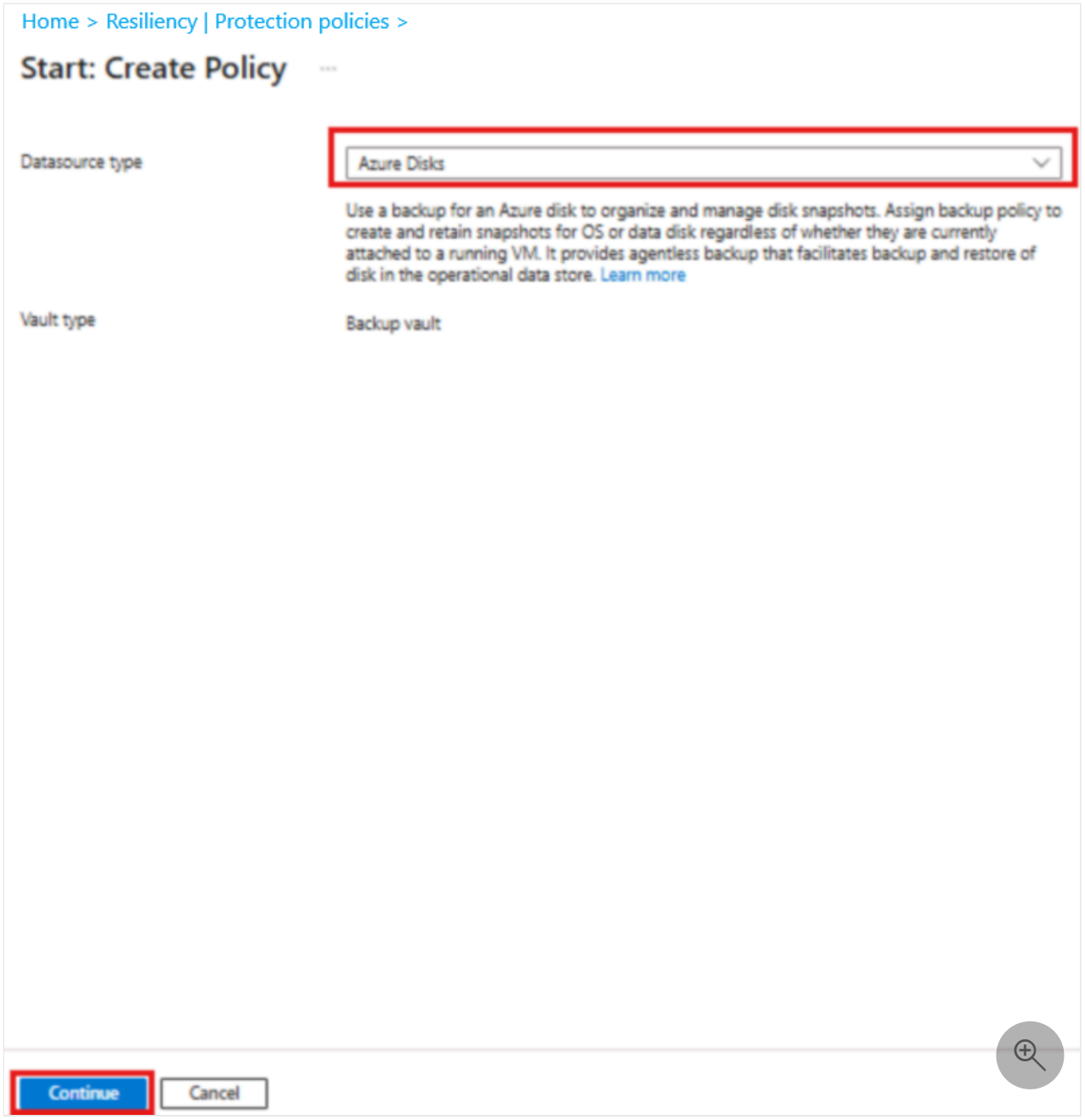
To create a backup policy for Azure Disks, follow these steps:

1. Go to **Resiliency**, and then select **Manage** > **Protection policies**.



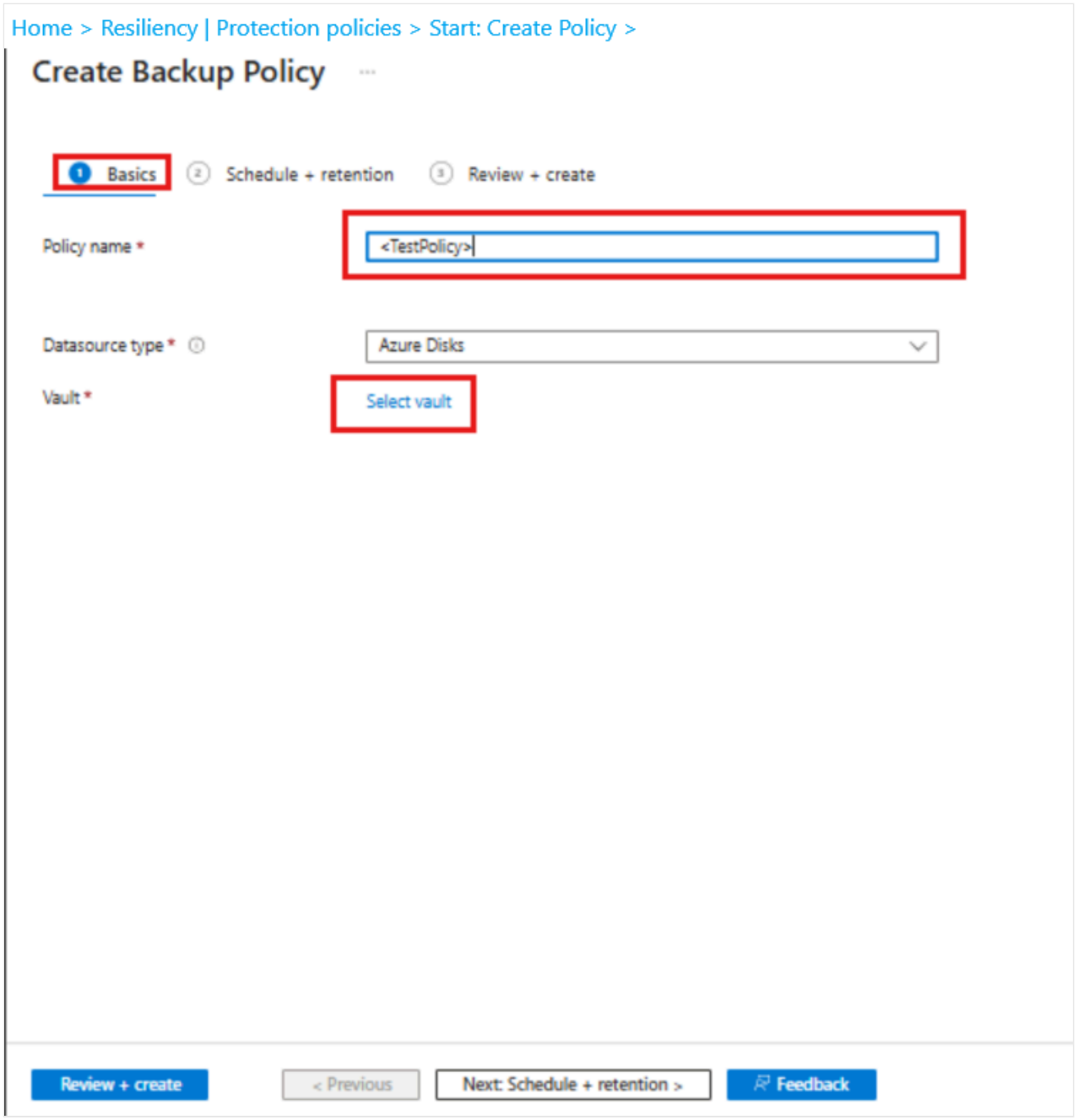
2. On the **Protection policies** pane, select **+ Create policy** > **Create backup policy**.

3. On the **Start: Create Policy** pane, select the **Datasource type** as **Azure Disks**, and then select **Continue**.



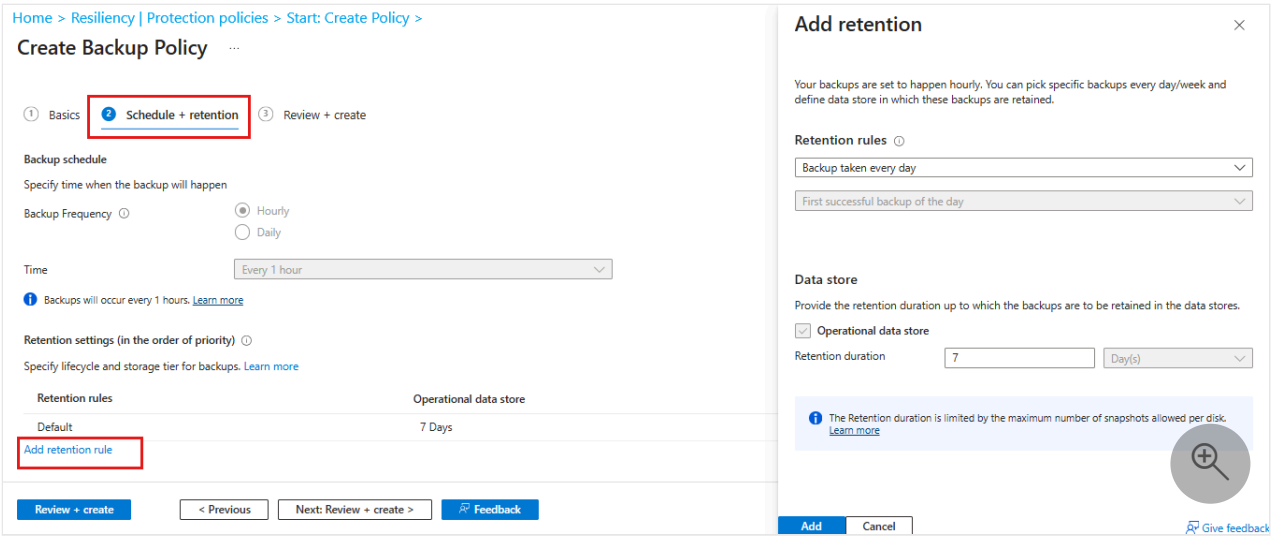
4. On the **Create Backup Policy** pane, on the **Basics** tab, enter a **Policy name**, and then under **Vault**, click **Select**.

> ⓘ **Note**
>
> Although the selected vault may have the global-redundancy setting, currently Azure Disk Backup supports snapshot datastore only. All backups are stored in a resource group in your subscription and aren't copied to backup vault storage.



5. On the **Select a Vault** pane, select the vault from the list that you created, and then click **Select**.

6. On the **Create Backup Policy** pane, on the **Schedule + retention** tab, under **Backup schedule**, select the backup frequency.



Azure Disk Backup offers multiple backups per day. If you require more frequent backups, choose the **Hourly** backup frequency with the ability to take backups with intervals of every *1*, *2*, *4*, *6*, *8*, or *12* hours. The backups are scheduled based on the **Time** interval selected. For example, if you select **Every 4 hours**, then the backups are taken at approximately in the interval of every 4 hours so the backups are distributed equally across the day. If a once a day backup is sufficient, then choose the **Daily** backup frequency. In the daily backup frequency, you can specify the time of the day when your backups are taken. It's important to note that the time of the day indicates the backup start time and not the time when the backup completes. The time required for completing the backup operation is dependent on various factors including size of the disk, and churn rate between consecutive backups. However, Azure Disk backup is an agentless backup that uses incremental snapshots, which doesn't impact the production application performance.

7. Under **Retention rules**, select retention settings that meet the recovery point objective (RPO) requirement.

   The default retention rule applies if no other retention rule is specified. The default retention rule can be modified to change the retention duration, but it cannot be deleted. You can add a new retention rule by selecting **Add retention rule**.

   You can pick **first successful backup** taken daily or weekly, and provide the retention duration that the specific backups are to be retained before they're deleted. This option is useful to retain specific backups of the day or week for a longer duration of time. All other frequent backups can be retained for a shorter duration.

   > **Note**
   >
   > Azure Backup for Managed Disks uses incremental snapshots, which are limited to 500 snapshots per disk. At a point in time, you can have 500 snapshots for a disk.

Thus, to prevent backup failure the retention duration is limited by the snapshot limit. To allow you to take on-demand backups aside from scheduled backups, backup policy limits the total backups to 450. Learn more about incremental snapshots for managed disks.

You can either set a maximum retention limit of 1 year or 450 disk snapshots, whichever reaches first. For example, if you opt for a backup frequency of 12 hours, then you can retain each recovery point for maximum 225 days as the snapshot limit is breached beyond that.

8. On the **Review + create** tab, select **Create** to complete the backup policy creation.

> **Note**
>
> - For Azure Disks belonging to Standard HDD, Standard SSD, and Premium SSD SKUs, you can define the backup schedule with *Hourly* frequency (of 1, 2, 4, 6, 8, or 12 hours) and *Daily* frequency.
> - For Azure Disks belonging to Premium V2 and Ultra Disk SKUs, you can define the backup schedule with *Hourly* frequency of only 12 hours and *Daily* frequency.

Configure Azure Disk backup

- Azure Disk backup supports only the operational tier backup. Copying of backups to the vault storage tier is currently not supported. The Backup vault storage redundancy setting (LRS/GRS) doesn't apply to the backups stored in the operational tier.
  Incremental snapshots are stored in a Standard HDD storage, irrespective of the selected storage type of the parent disk. For more reliability, incremental snapshots are stored on Zone Redundant Storage (ZRS) by default in ZRS supported regions.

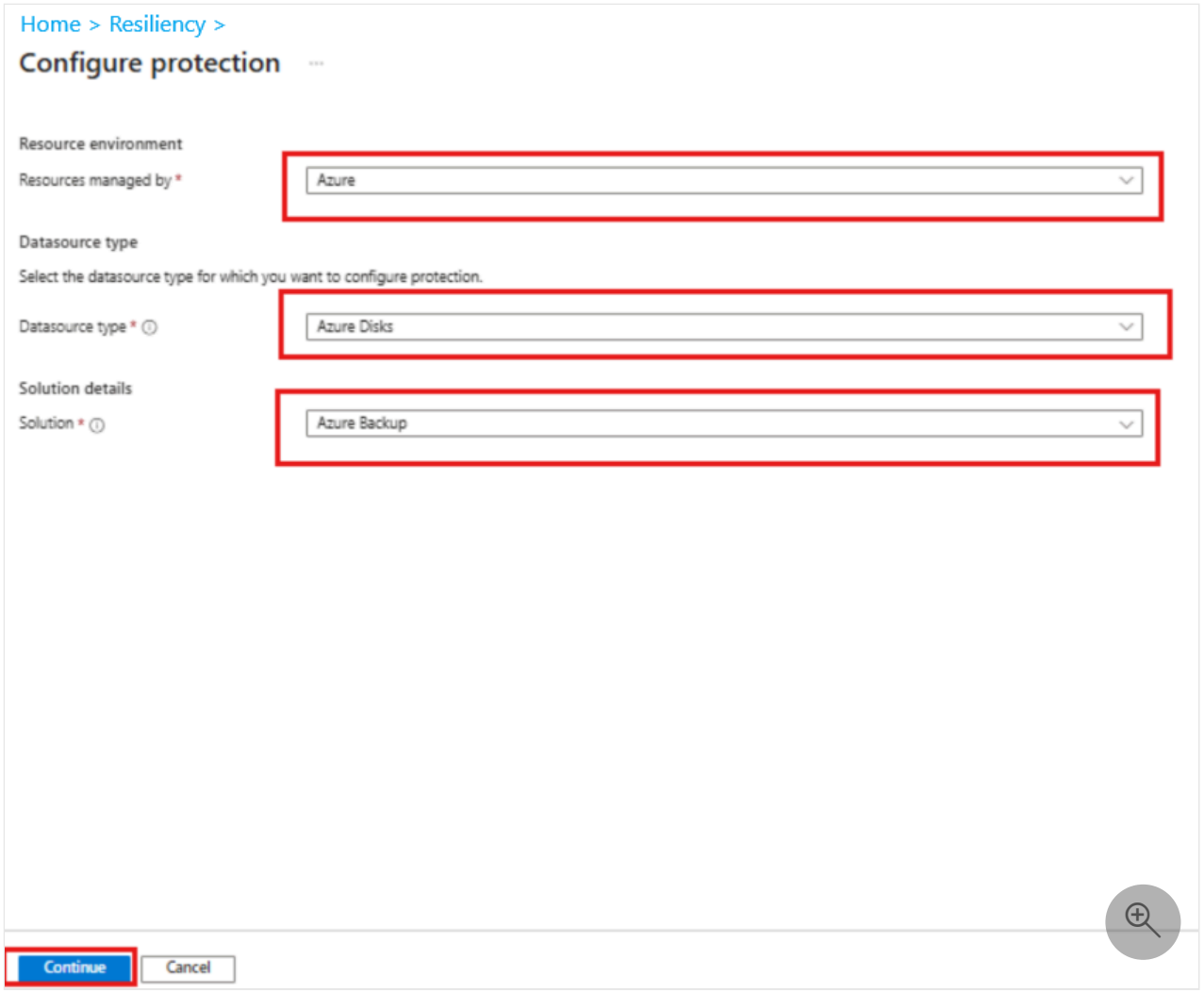
- Azure Disk backup supports cross-subscription (backup vault in one subscription and the source disk in another) backup and restore. Currently, cross-region backup and restore aren't supported by Azure Disk backup, that is, the backup vault and disk to back up are in different regions.
  So, to use Azure Disk backup, ensure that the backup vault and disk to back up are in the same region.

- Once you configure the disk backup, you can't change the Snapshot Resource Group that's assigned to a backup instance.

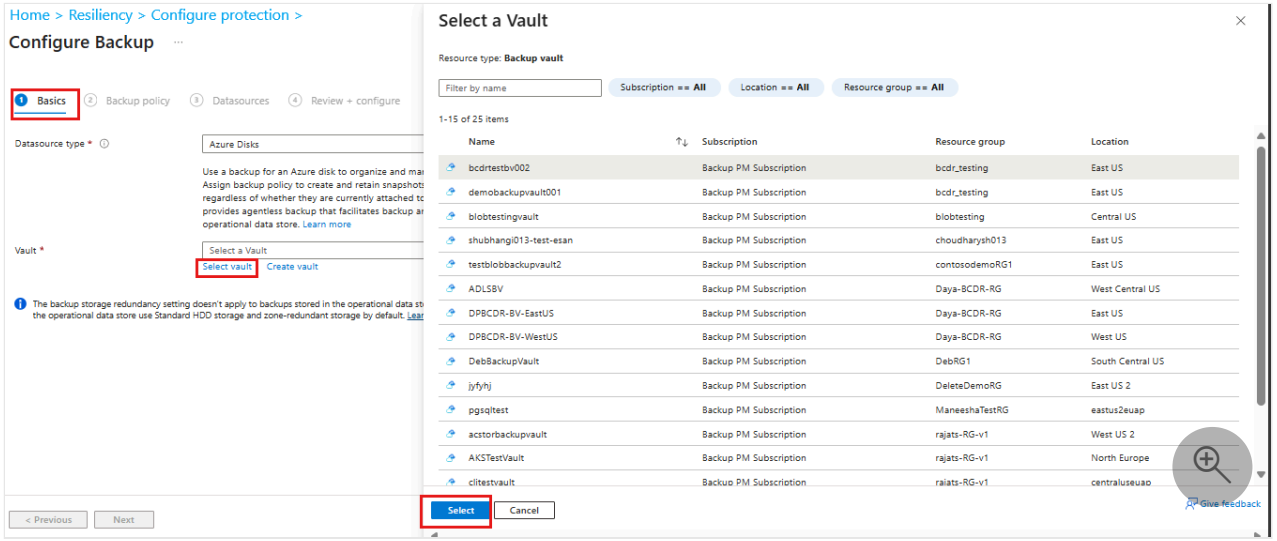
To configure Azure Disk backup, follow these steps:

1. Go to **Resiliency**, and then select **+ Configure protection**.

2. On the **Configure protection** pane, select **Resource managed by** as **Azure**, **Datasource type** as **Azure Disks**, **Solution** as **Azure Backup**, and then select **Continue**.

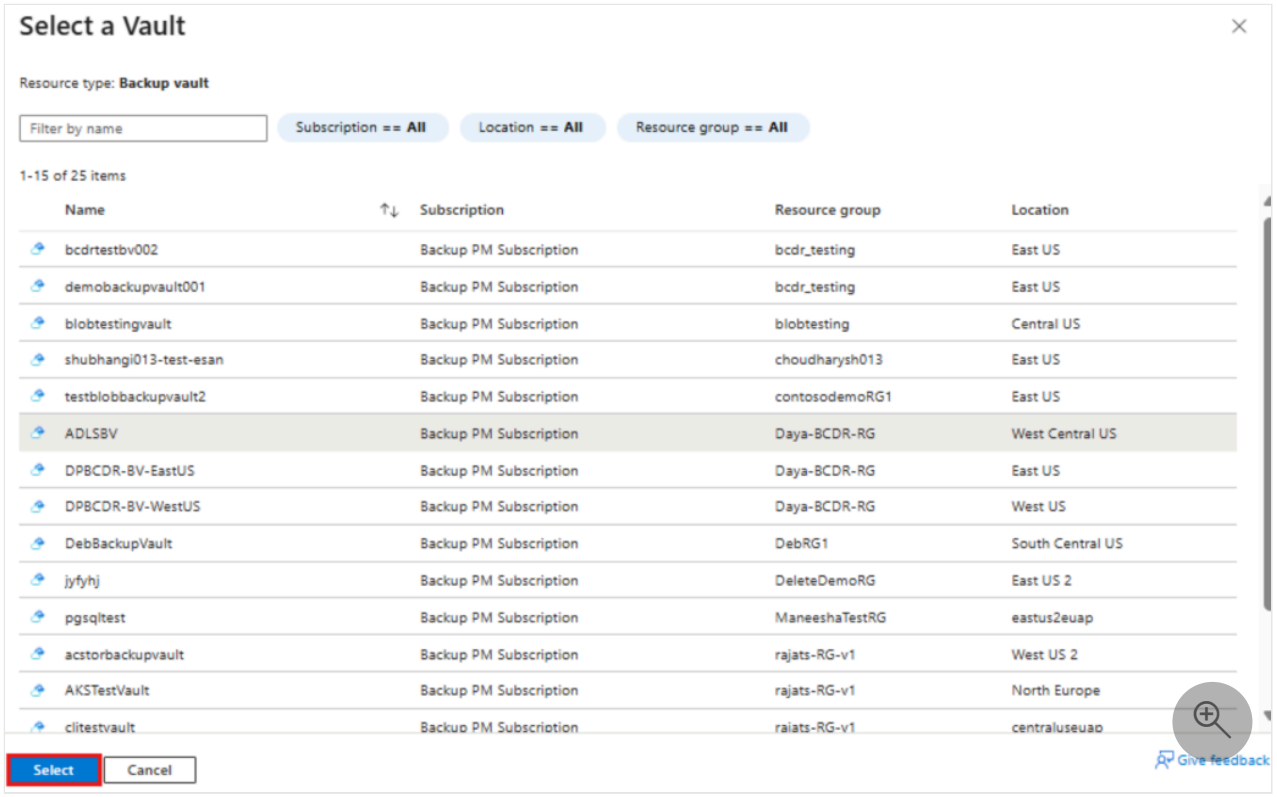


3. On the **Configure Backup** pane, on the **Basics** tab, under **Vault**, click **Select vault**.



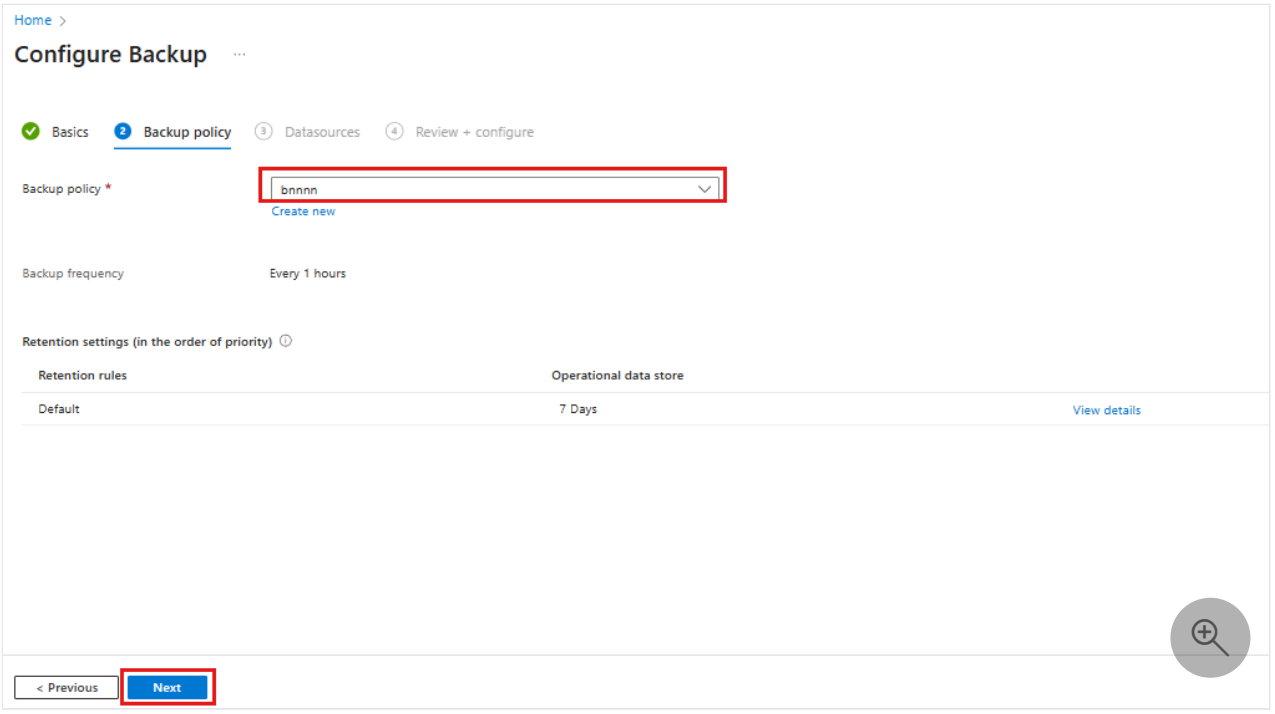
4. On the **Select a Vault** pane, select the Backup vault from the list you created, and then click **Select**.

> **Note**
>
> - Ensure that both the backup vault and the disk to be backed up are in same location.
> - Azure Backup uses ***incremental snapshots*** of managed disks, which store only the delta changes to the disk as the last snapshot on Standard HDD storage, regardless of the storage type of the parent disk. For additional reliability, incremental snapshots are stored on Zone Redundant Storage (ZRS) by default in the ZRS supported regions. Currently, Azure Disk Backup supports operational backup of managed disks that doesn't copy backups to the Backup vault storage. So, the backup storage redundancy setting of the Backup vault doesn't apply to the recovery points.

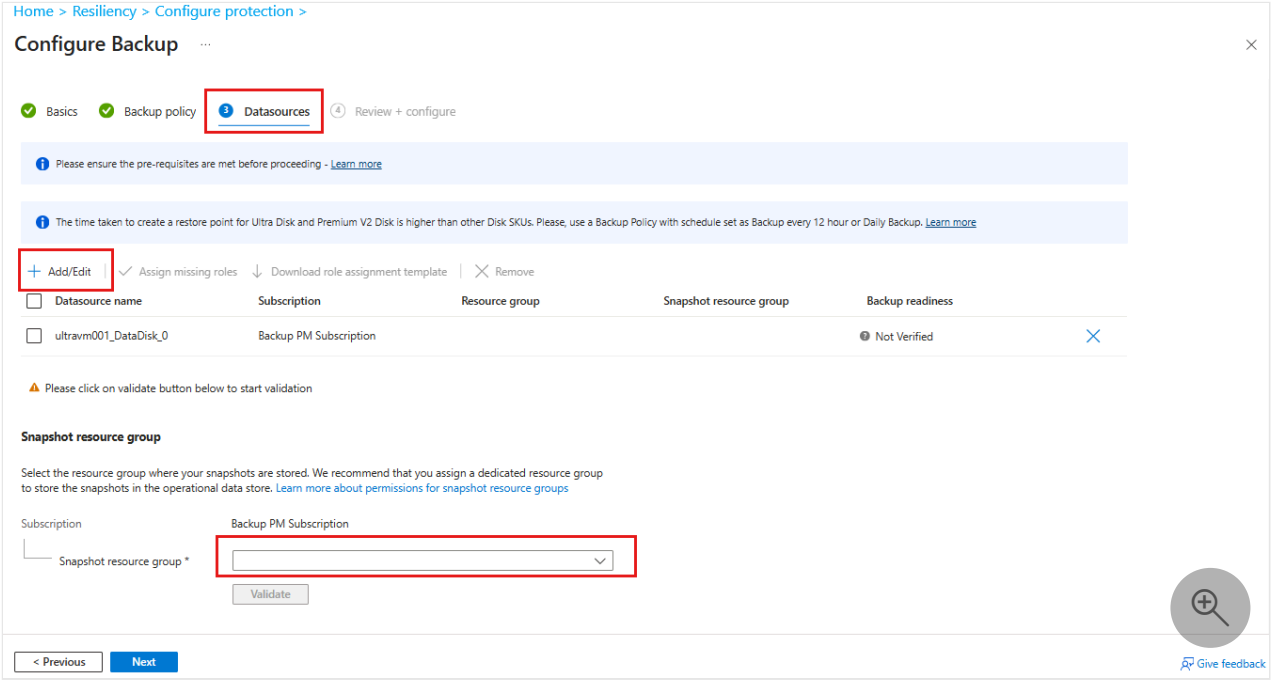


5. On the **Configure Backup** pane, on the **Basics** tab, select **Next**.

6. On the **Backup policy** tab, choose a Backup policy, and then select **Next**.



7. On the **Datasources** tab, select **+ Add/Edit** to choose one or more Azure Managed Disks for which you want to configure backup.



> **Note**
>
> While the portal allows you to select multiple disks and configure backup, each disk is an individual backup instance. Currently, Azure Disk Backup only supports backup of individual disks. Point-in-time backup of multiple disks attached to a virtual machine isn't supported.
>
> In the Azure portal, you can only select disks within the same subscription. If you have several disks to be backed up or if the disks reside in different subscriptions,

you can use scripts (PowerShell/CLI) to automate.

See the support matrix for more information on the Azure Disk backup region availability, supported scenarios, and limitations.

8. Select **Snapshot resource group** and then select **Validate** to initiate prerequisites checks.

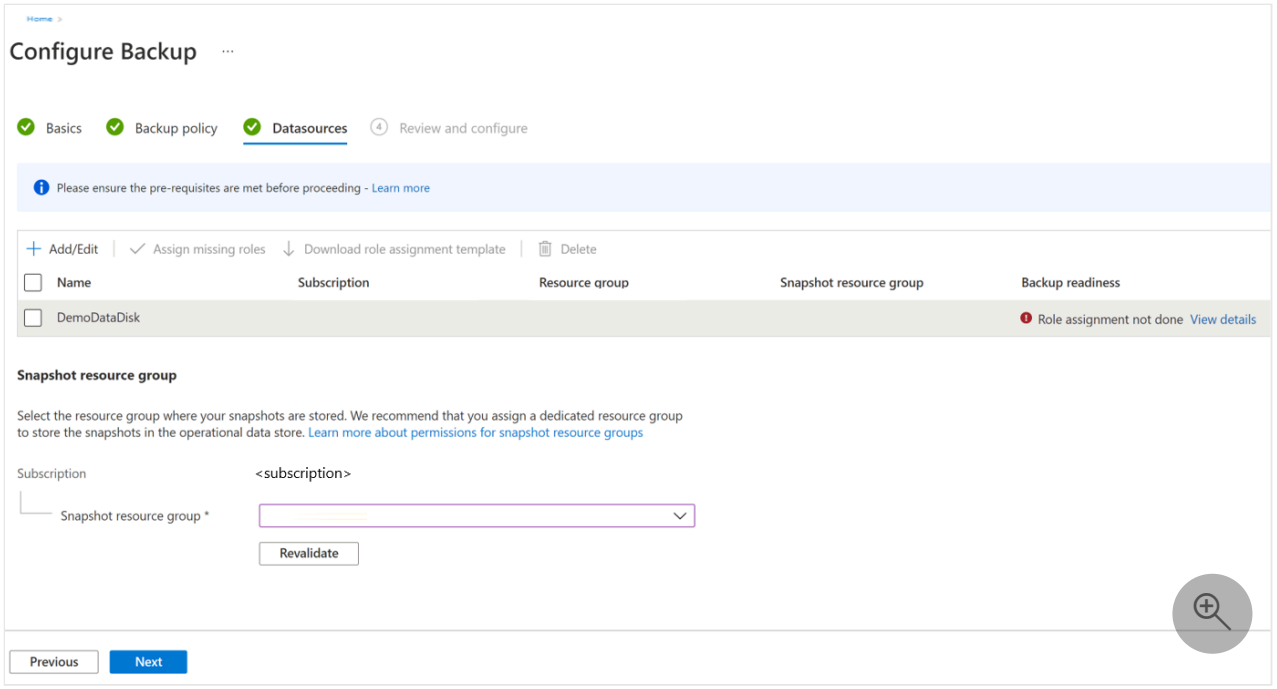
   Choosing resource group for storing and managing snapshots:

   - Don't select the same resource group as that of the source disk.
   - As a guideline, it's recommended to create a dedicated resource group as a snapshot datastore to be used by the Azure Backup service. Having a dedicated resource group allows restricting access permissions on the resource group, providing safety and ease of management of the backup data.
   - You can use this resource group for storing snapshots across multiple disks that are being (or planned to be) backed up.
   - You can't create an incremental snapshot for a particular disk outside of that disk's subscription. So, choose the resource group within the same subscription where the disk needs to be backed up. Learn more about incremental snapshot for managed disks.
   - Once you configure the backup of a disk, you can't change the Snapshot Resource Group that's assigned to a backup instance.

9. When the validation is complete, check if there are any errors reported in the Backup readiness column.

   > ⓘ **Note**
   >
   > Validation might take few minutes to complete. Validation may fail if:
   >
   > - A disk is unsupported. See the support matrix for unsupported scenarios for more information.
   > - The Backup vault managed identity does not have valid role assignments on the *disk* to be backed up or on the *snapshot resource group* where incremental snapshots are stored.

   If the *Role assignment not done* error message displays in the **Backup readiness** column, the Backup vault managed identity needs role permissions on one or more selected disk(s) and/or on the Snapshot resource group.



To configure backup of managed disks, the following prerequisites are required:

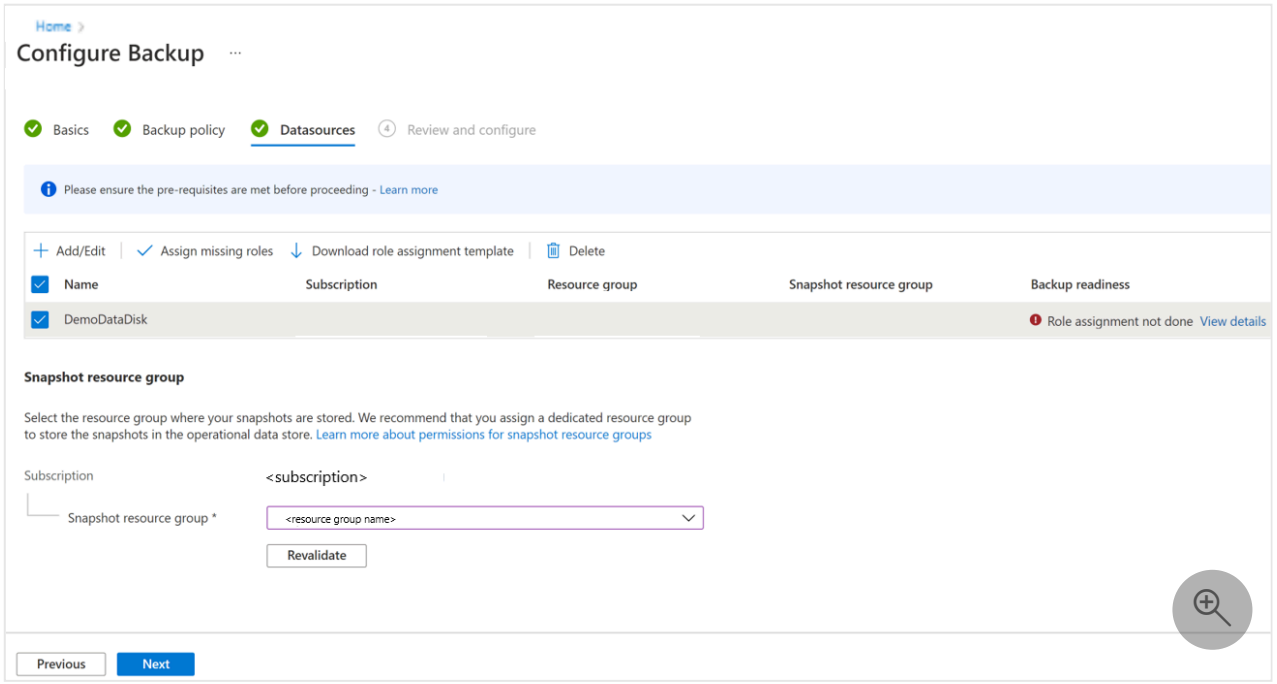
> **Note**
>
> Backup vault uses managed identity to access other Azure resources. To configure a backup of managed disks, Backup Vault's managed identity requires a set of permissions on the source disks and resource groups where snapshots are created and managed.

A system-assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. To grant permissions to the managed identity, use Azure role-based access control (Azure RBAC). Managed identity is a service principal of a special type that may only be used with Azure resources. Learn more about managed identities.

- Assign the **Disk Backup Reader** role to Backup Vault's managed identity on the Source disk that needs to be backed up.
- Assign the Disk Snapshot Contributor role to the Backup vault's managed identity on the Resource group where backups are created and managed by the Azure Backup service. The disk snapshots are stored in a resource group within your subscription. To allow Azure Backup service to create, store, and manage snapshots, you need to provide permissions to the backup vault.

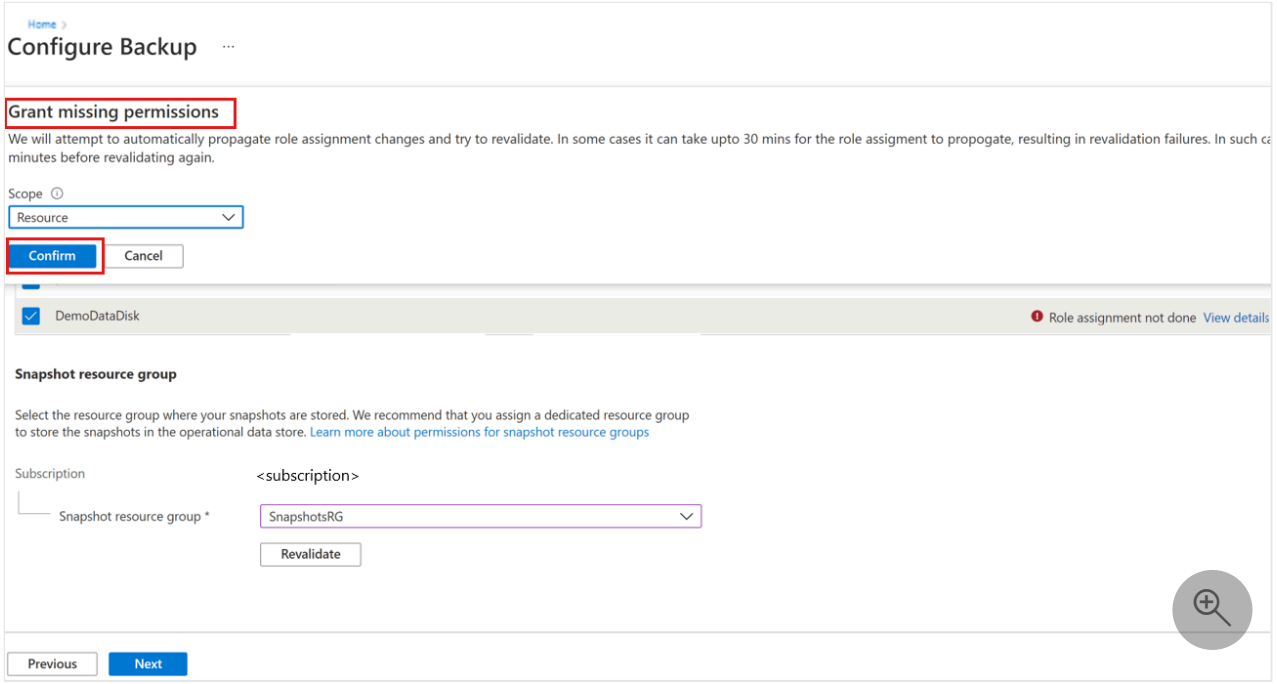
> **Note**
>
> The Configure Backup flow using Azure portal helps you in granting required role permissions to the above resources.

10. Select the checkbox next to each row with the *Role assignment not done* error message status in the Backup readiness column and select **Add missing roles** to automatically grant required role permissions for the Backup vault managed identity on selected resources.



11. Select **Confirm** to provide consent. Azure Backup automatically propagates role assignment changes on your behalf and tries to revalidate.

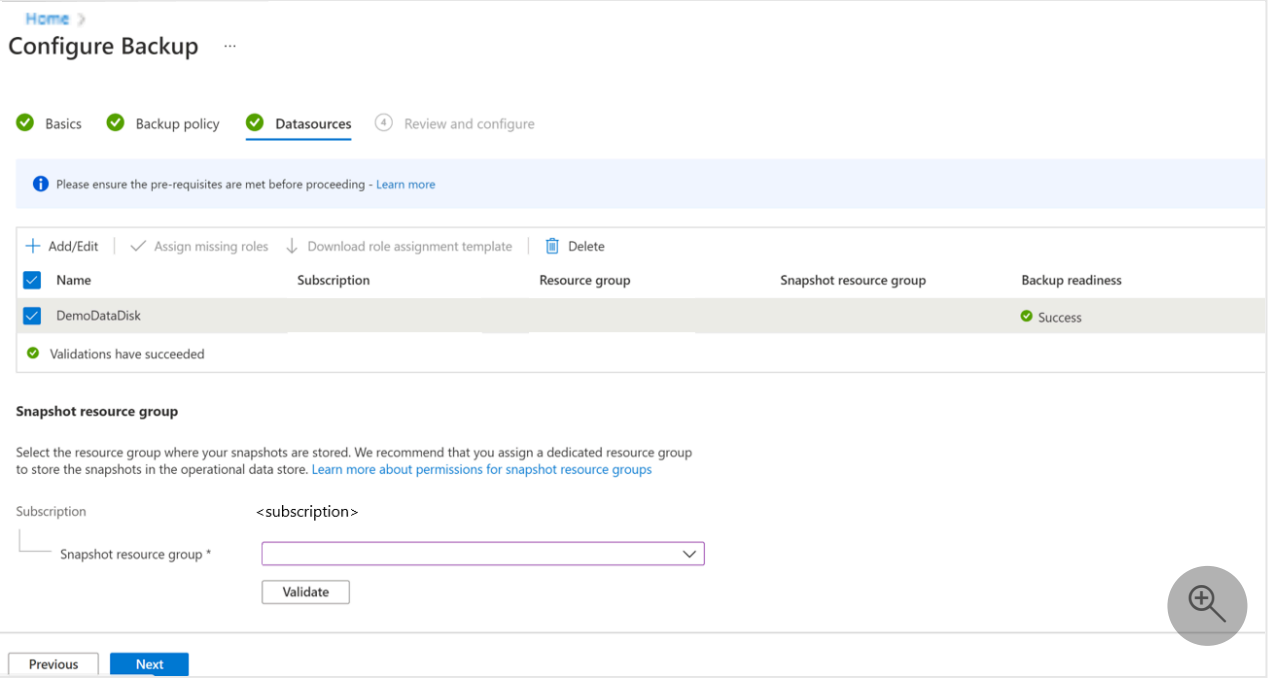
    If you want to grand permission for the Backup vault managed identity to the selected disk(s) and snapshot resource group, select **Resource** in the **Scope** drop-down list.



> **Tip**

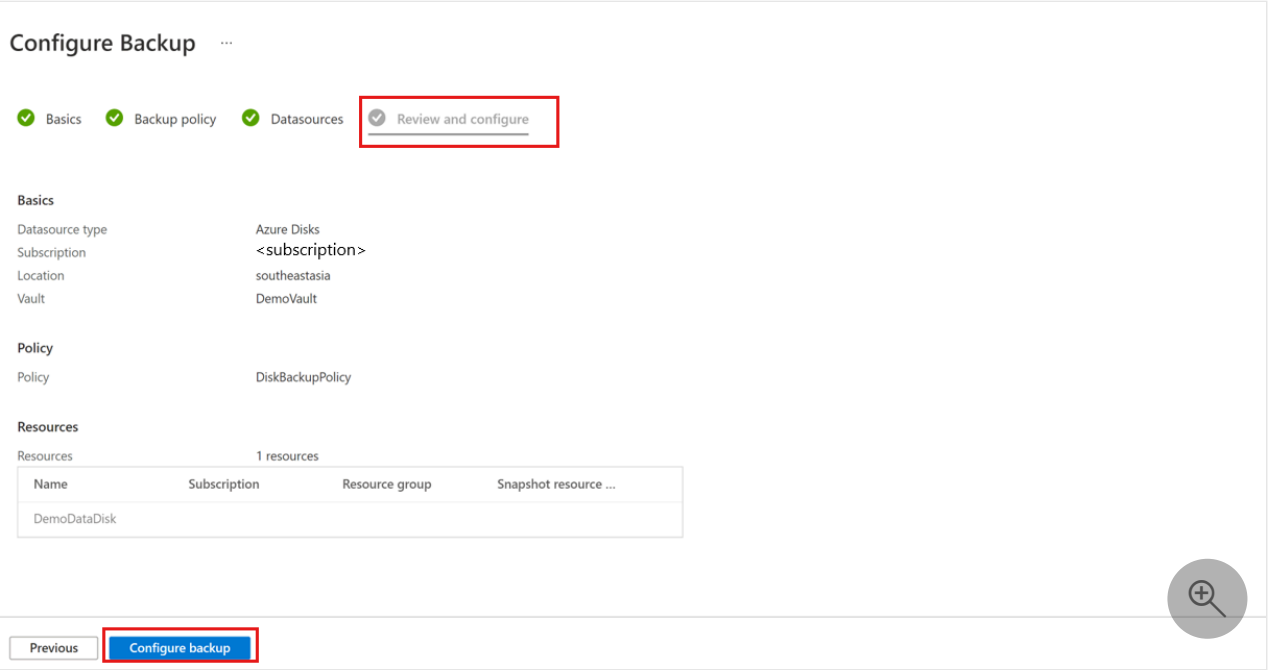
If you plan to configure backup for other disks in the same resource group/subscription in future, you can choose to provide permission at the scope of resource group or subscription.

> **Note**
>
> - In some cases, it can take up to 30 minutes for the role assignments to propagate, causing revalidation failure. In this scenario, retry after some time.
> - If the **Add missing roles** action fails to assign permissions with the error 'Insufficient permission for role assignment' in Backup readiness column, it indicates that you don't have the privilege to assign role permissions. Choose Download role assignment template to download role assignments as scripts and seek support from your IT Administrator to run the scripts to complete the prerequisites.



12. After a successful validation, select **Next**.

13. On the **Review and configure** tab, select **Configure backup** to complete the backup configuration of selected disks.

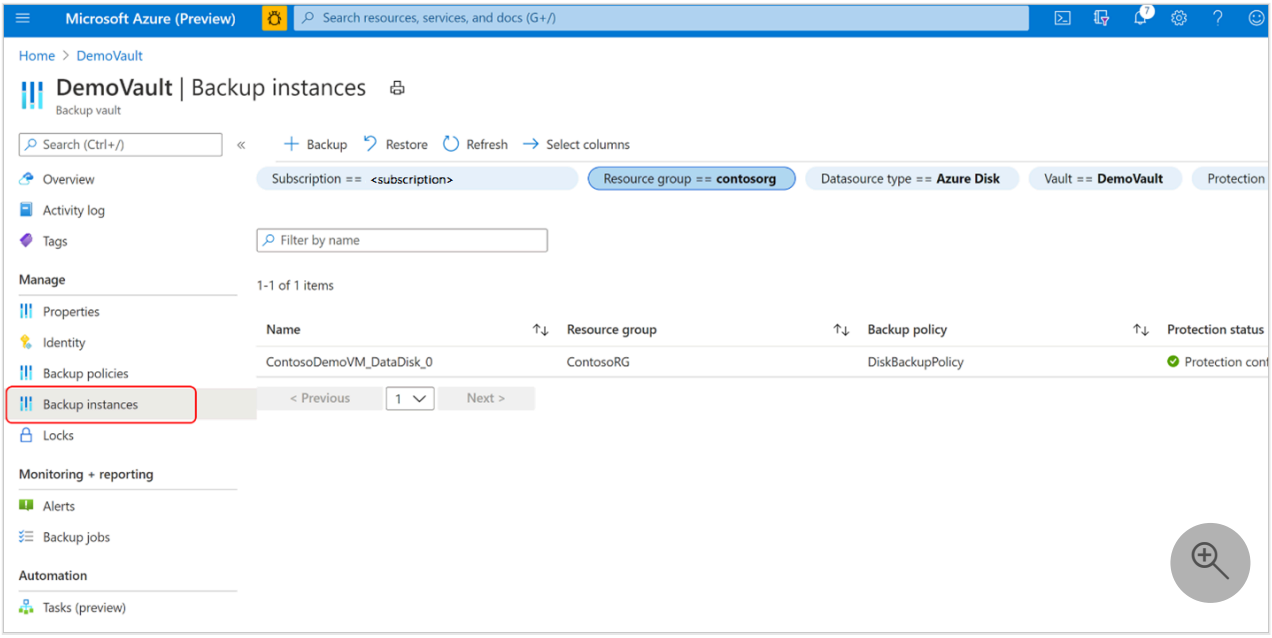


Run an on-demand backup for Azure Disks

You can run an on-demand backup for Azure Disks at any time. This operation is useful if you want to create a backup outside of the scheduled backup times defined in the backup policy.

To run an on-demand backup for Azure Disks, follow these steps:

1. In the *DemoVault* **Backup vault** created in the previous step, go to **Backup instances** and select a backup instance.

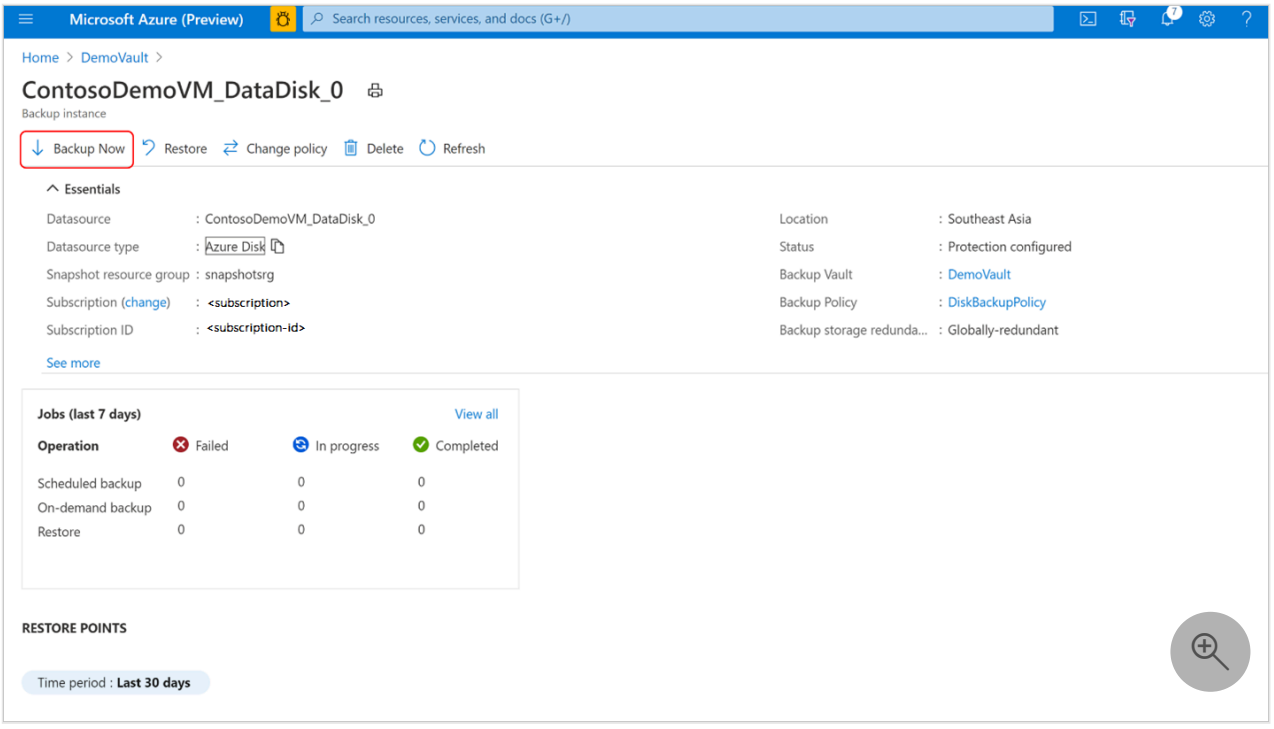


2. On the **Backup instances** pane, you can find:

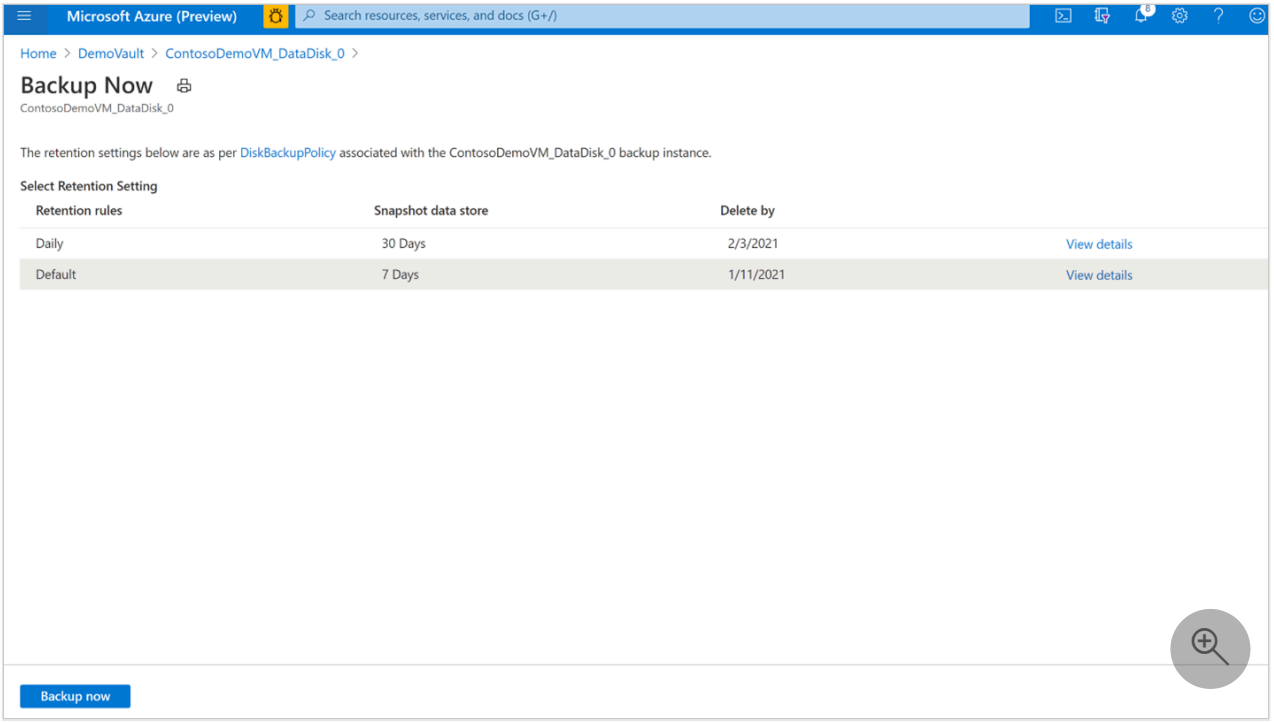
   - **essential** information including source disk name, the snapshot resource group where incremental snapshots are stored, backup vault, and backup policy.

- **Job status** showing summary of backup and restore operations and their status in the last seven days.
- A list of **restore points** for the selected time period.

3. Select **Backup** to initiate an on-demand backup.



4. Select one of the retention rules associated with the backup policy. This retention rule determines the retention duration of this on-demand backup. Select **Backup now** to start the backup.



Track Azure Disk backup operations

The Azure Backup service creates a job for scheduled backups or if you trigger on-demand backup operation for tracking. Learn how to view the backup job status and details.

Next steps

- Restore Managed Disk using Azure portal, Azure PowerShell, Azure CLI.
- Troubleshoot Azure Disk backup.

Related content

- Create a backup policy to protect Managed Disk using REST API.
- Back up Managed Disk using REST API.
- Restore Managed Disk using REST API.

Last updated on 02/13/2026

Restore Azure Managed Disks

Summarize this article for me

This article describes how to restore Azure Managed Disks from a restore point created by Azure Backup. You can also restore Managed Disk using Azure PowerShell, Azure CLI, REST API.

Backup Vault uses Managed Identity to access other Azure resources. To restore from backup, Backup vault's managed identity requires a set of permissions on the resource group where the disk is to be restored.

Backup vault uses a system assigned managed identity, which is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (RBAC). Managed identity is a service principal of a special type that may only be used with Azure resources. Learn more about Managed Identities.

> **Note**
>
> Currently, the Original-Location Recovery (OLR) option of restoring by replacing existing the source disk from where the backups were taken isn't supported. You can restore from a recovery point to create a new disk either in the same resource group as that of the source disk from where the backups were taken or in any other resource group. This is known as Alternate-Location Recovery (ALR) and this helps to keep both the source disk and the restored (new) disk.

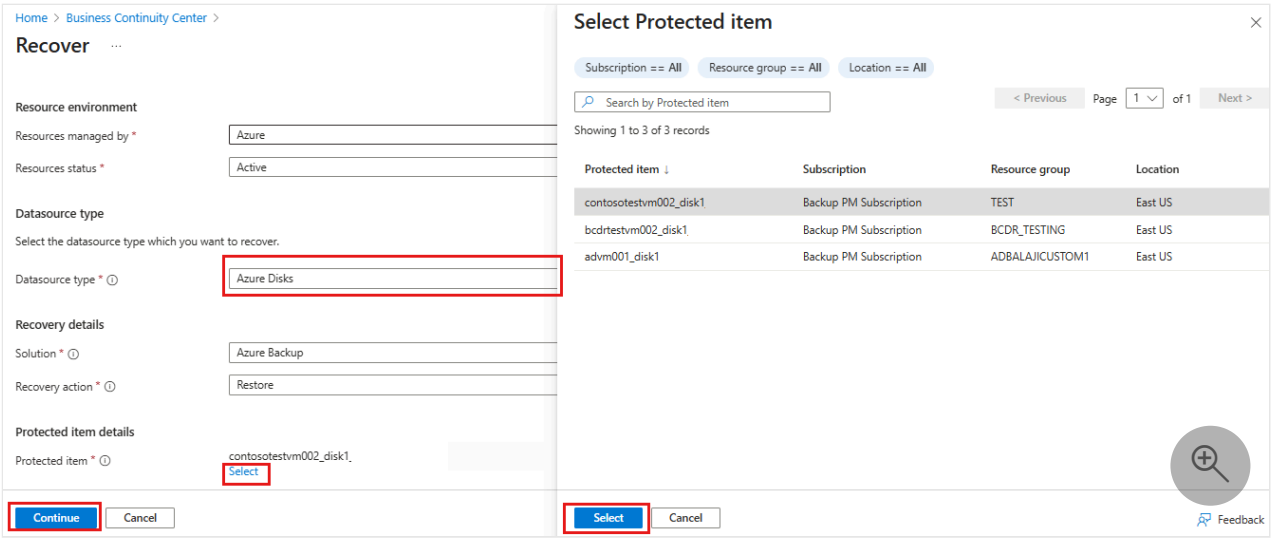
Restore a new disk from a recovery point

To restore a new disk from a recovery point, follow these steps:

1. Go to **Resiliency**, and then select **Recover**.

   Alternately, you can perform this operation from the Backup vault you used to configure backup for the disk.

2. On the **Recover** pane, select **Datasource type** as **Azure Disks**, and then under **Protected item**, click **Select** to choose a disk instance.

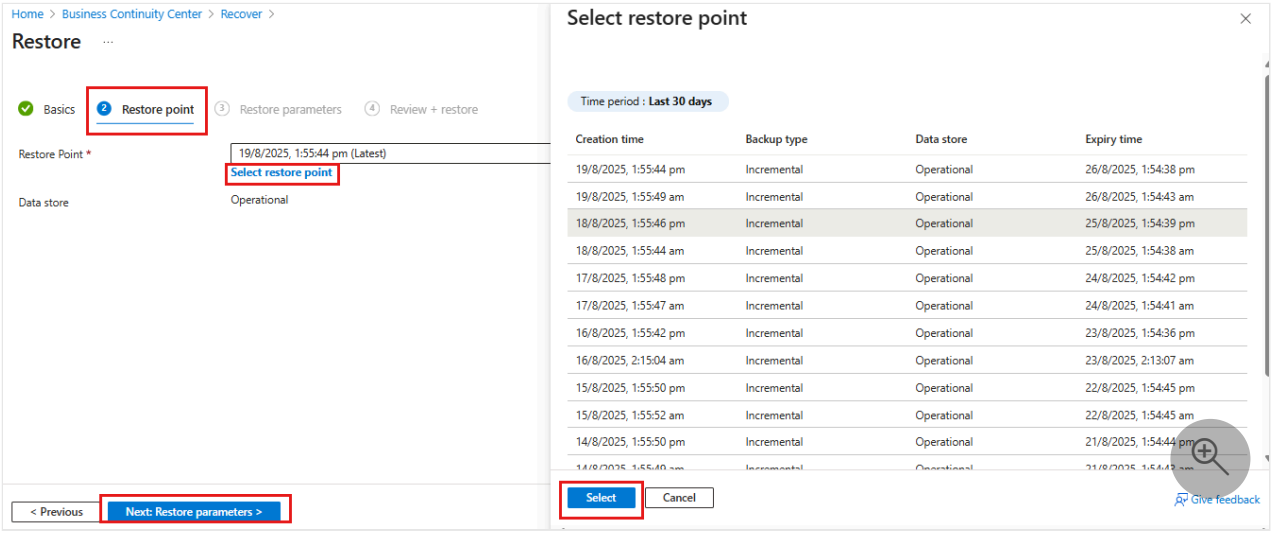


3. On the **Select Protected item** pane, select a disk instance from the list, and then click **Select**.

4. On the **Recover** pane, select **Continue**.

5. On the **Restore** pane, on the **Basics** tab, select **Next: Restore point**.

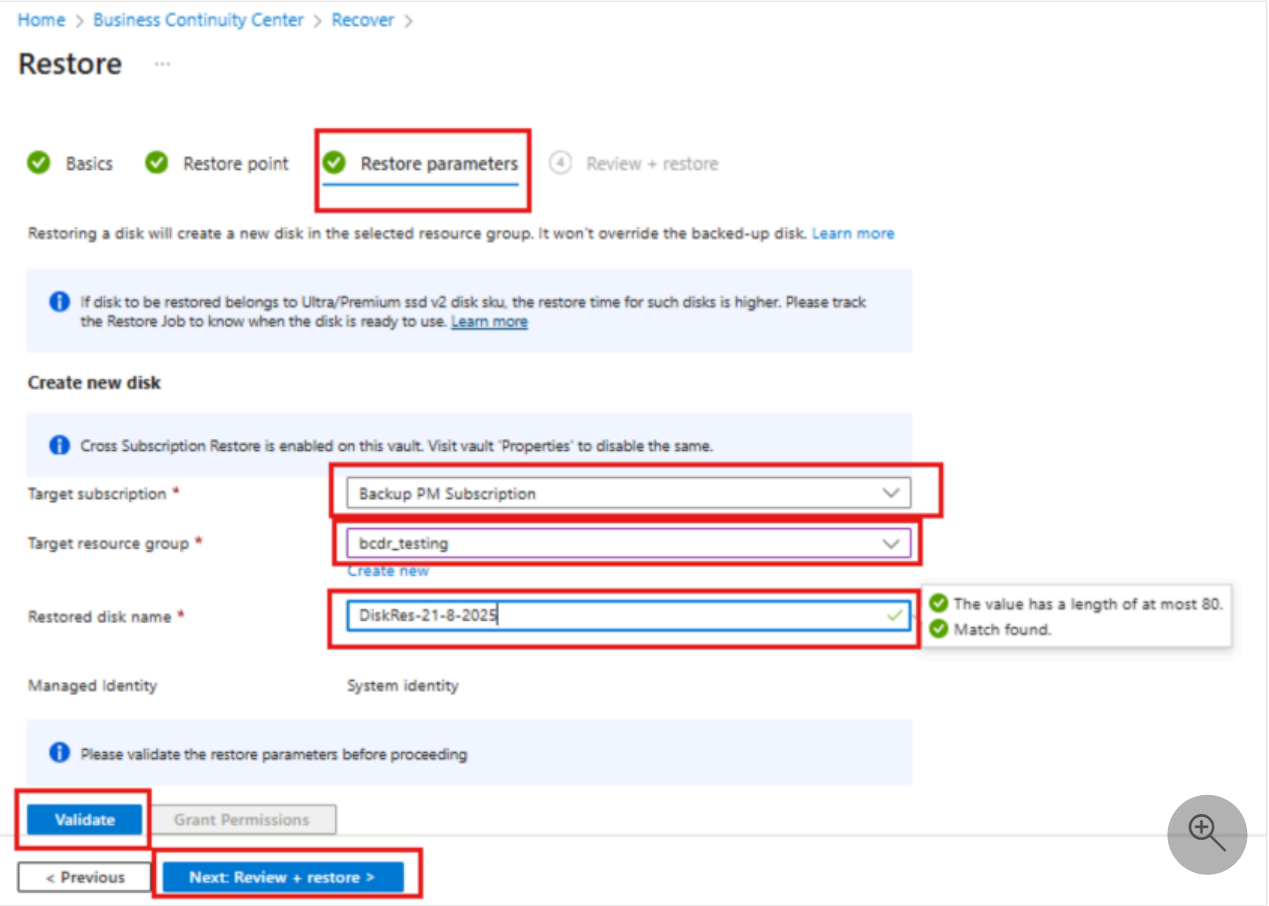
6. On the **Restore point** tab, under **Restore Point**, click **Select restore point**.



7. On the **Select restore point** pane, choose a restore point from the list, and then click **Select**.

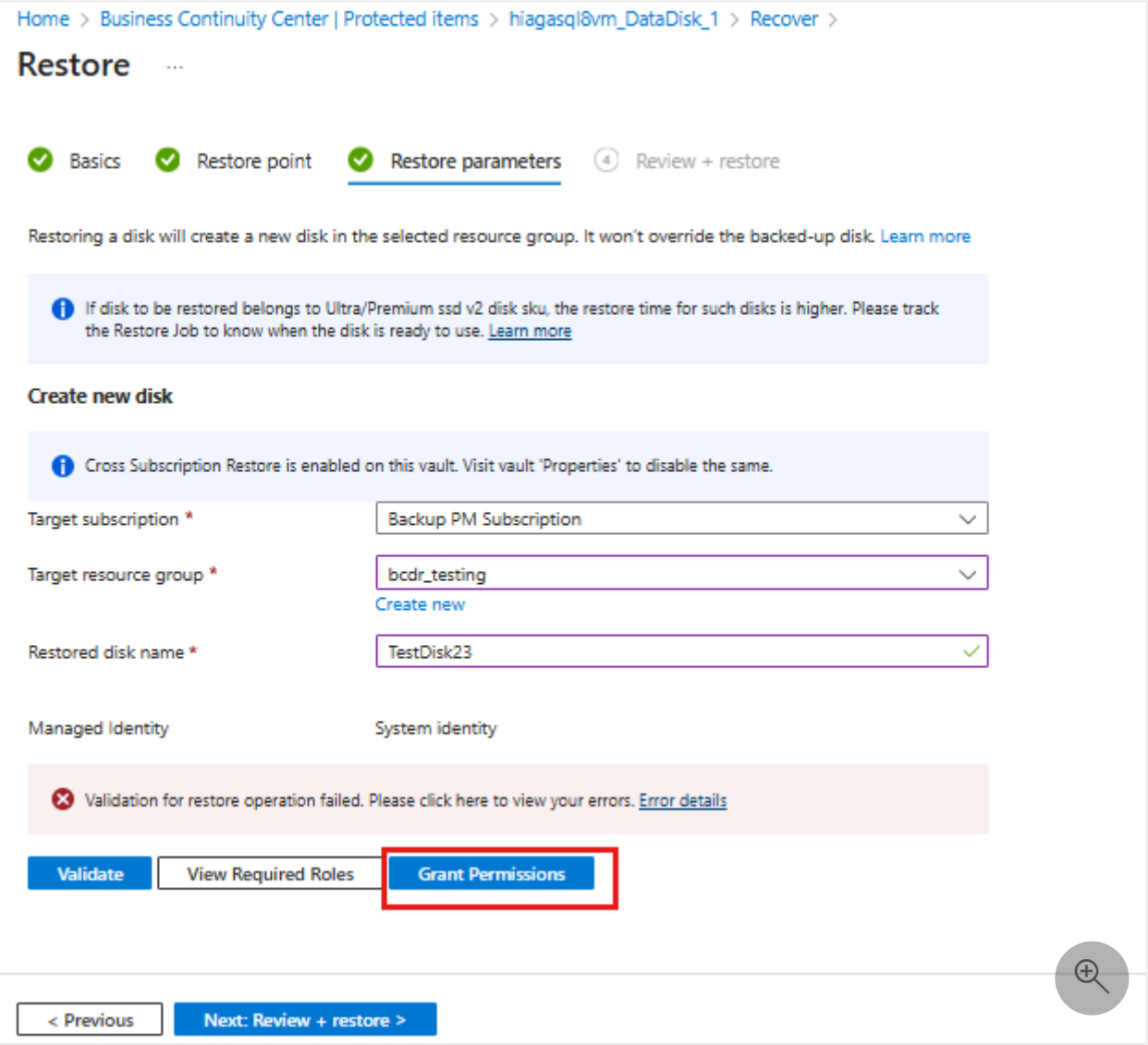
8. On the **Restore** pane, select **Next: Restore parameters**.

9. On the **Restore parameters** pane, select the **Target subscription** and **Target resource group** where you want to restore the backup to, enter a value under **Restored disk name**, and then select **Validate**.



> **Tip**
>
> You can protect disks with Azure Backup using either the Disk Backup solution or the Azure Virtual Machine (VM) backup solution with a Recovery Services vault. If the Azure VM is already protected that uses this disk, you can restore the VM, individual disks, or files and folders from the VM backup recovery point. For more information, see Azure VM backup.

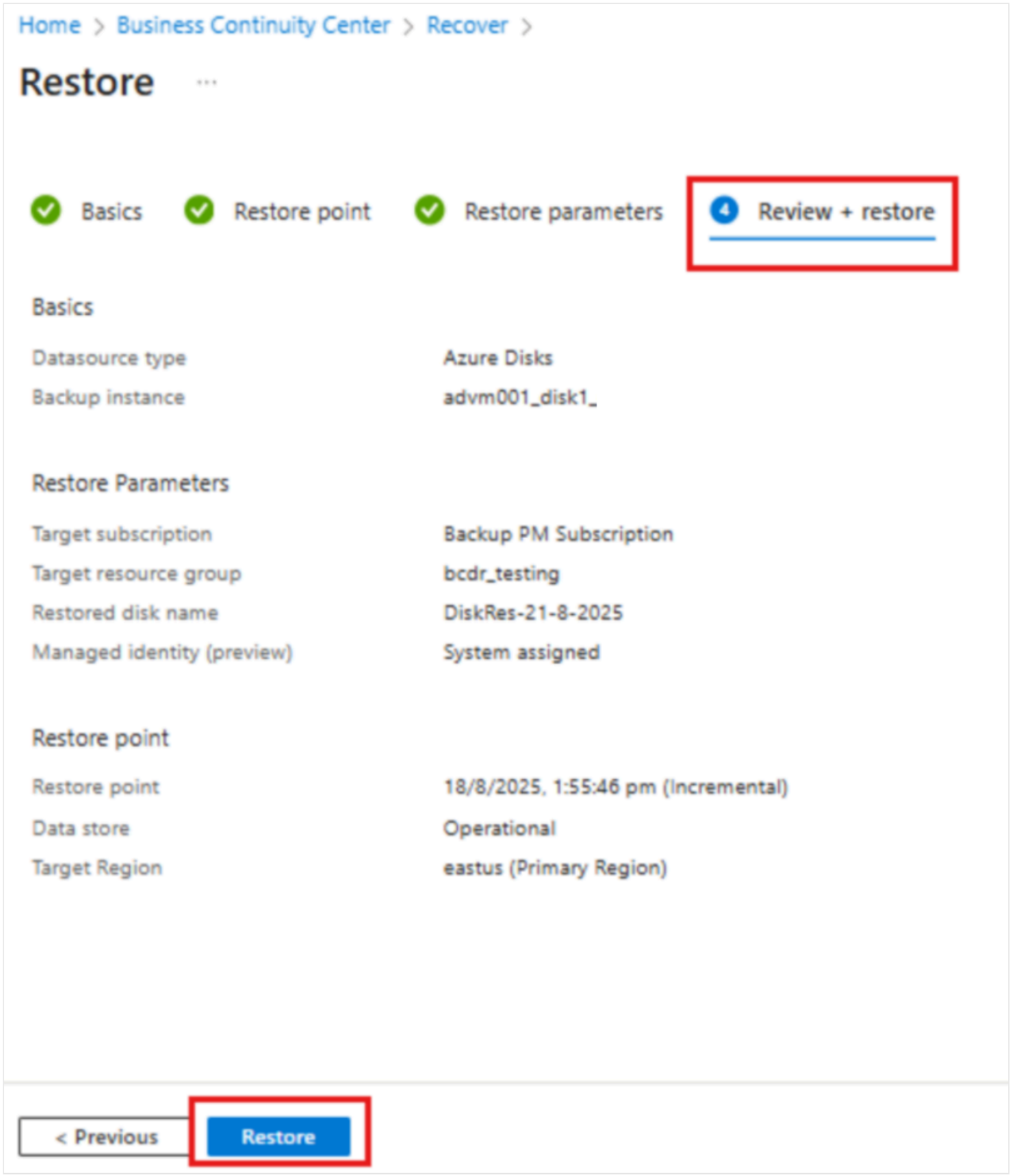
If the validation fails due to role assignment not done, select **Grant Permissions**. After the role assignment is complete, the revalidation runs automatically.



> ⓘ **Note**
>
> While the role assignments are reflected correctly on the portal, it may take approximately 15 minutes for the permission to be applied on the backup vault's managed identity.
>
> During scheduled backups or an on-demand backup operation, Azure Backup stores the disk incremental snapshots in the Snapshot Resource Group provided during configuring backup of the disk. Azure Backup uses these incremental snapshots during the restore operation. If the snapshots are deleted or moved from the Snapshot Resource Group or if the Backup vault role assignments are revoked on the Snapshot Resource Group, the restore operation fails.

10. When the validation is successful, select **Next: Review + restore**.

11. On the **Review + restore** pane, review the restore settings, and then select **Restore** to start the restore operation.

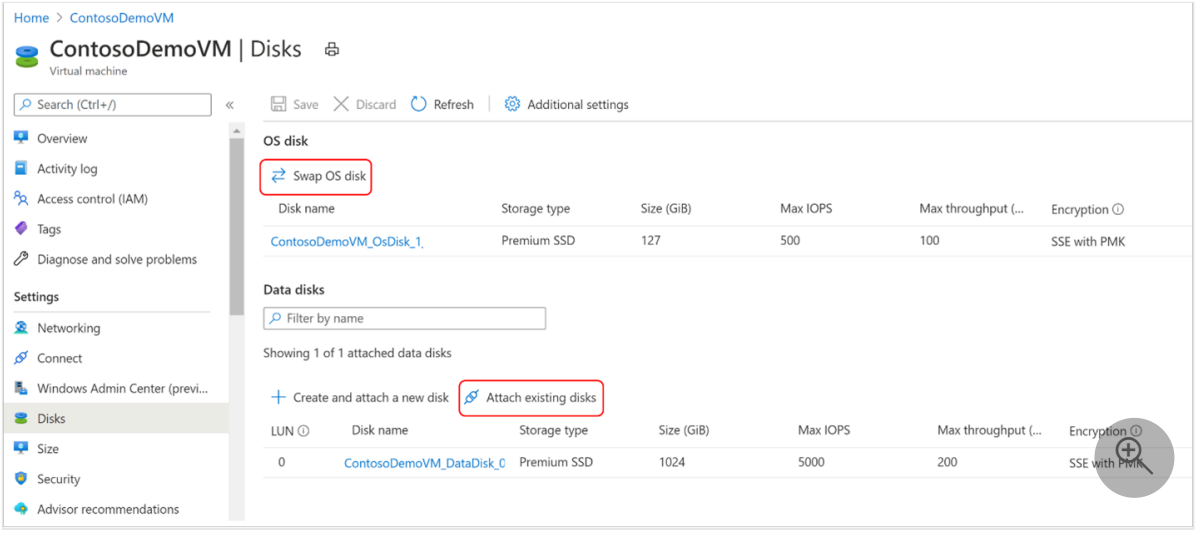


> ⓘ Note
>
> Validation might take few minutes to complete before you can trigger restore operation. Validation may fail if:
>
> - a disk with the same name provided in **Restored disk name** already exists in the **Target resource group**
> - the Backup vault's managed identity doesn't have valid role assignments on the **Target resource group**

- the Backup vault's managed identity role assignments are revoked on the **Snapshot resource group** where incremental snapshots are stored
- If incremental snapshots are deleted or moved from the snapshot resource group

The restore operation creates a new disk from the selected recovery point in the target resource group provided during the restore operation. To use the restored disk on an existing virtual machine, do the following actions:

- If the restored disk is a data disk, you can attach an existing disk to a virtual machine. If the restored disk is OS disk, you can swap the OS disk of a virtual machine from the Azure portal under the **Virtual machine** pane - > **Disks** menu in the **Settings** section.



- For Windows virtual machines, if the restored disk is a data disk, follow the instructions to detach the original data disk from the virtual machine. Then attach the restored disk to the virtual machine. Follow the instructions to swap the OS disk of the virtual machine with the restored disk.

- For Linux virtual machines, if the restored disk is a data disk, follow the instructions to detach the original data disk from the virtual machine. Then attach the restored disk to the virtual machine. Follow the instructions to swap the OS disk of the virtual machine with the restored disk.

> **Note**
>
> We recommend revoking the **Disk Restore Operator** role assignment from the Backup vault's managed identity on the **Target resource group** after the restore operation completes successfully.

Track a restore operation

After you trigger the restore operation, the backup service creates a job for tracking. Azure Backup displays notifications about the job in the portal. Learn how to view the restore job progress.

Next steps

- Azure Disk Backup FAQ.
- Manage Azure Managed Disks.

Last updated on 01/30/2026

Backup and disaster recovery for Azure managed disks

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This article explains how to plan for backup and disaster recovery for Azure managed disks. It introduces Azure built-in redundancy and some common failure types. After that, it covers typical backup and disaster recovery scenarios. Finally, it compares each backup and disaster recovery option for Azure managed disks.

Azure built-in redundancy options

Azure managed disks have two built-in redundancy options to protect your data against failures:

- Locally redundant storage (LRS) replicates your data three times within a single data center in a particular region. LRS protects your data against server rack and drive failures.
- Zone-redundant storage (ZRS) provides synchronous replication of data across zones in a region, enabling disks to tolerate zonal failures that may occur due to natural disasters or hardware issues.

However, major disasters (such as earthquakes, fires, or hurricanes) can result in outages or inaccessibility of large-scale storage servers, sometimes impacting a whole data center or zone (impacting LRS disks), or an entire region (impacting ZRS disks). In addition to platform failures, problems with an application or data can also occur (such as accidental deletes and ransomware attack). When these happen, you might want to revert the application and the data to a prior version that contains the last known good state. Reverting to a good state requires regular backups.

To protect your IaaS workloads from outages, plan for redundancy, and create regular backups. To protect IaaS workloads from regional disasters, create backups in a different geographic location than your primary site. This ensures your backups aren't affected by the same events that affected your other resources. For more information, see Disaster Recovery for Azure Applications.

Backup and Disaster Recovery Scenarios

Let's look at a few examples of application workload scenarios and things to consider when planning for backup and disaster recovery.

Scenario 1: Major database solutions

In this scenario, you have a production database server (such as SQL Server and Oracle) that supports high availability. Critical production applications and users depend on this database. The disaster recovery plan for this system should include the following requirements:

- The data must be protected and recoverable.
- The server must be available for use.
- (Optional) A replica of the database in a different region as backup

Depending on your requirements for server availability and data recovery, solutions could range from an active-active or active-passive replica site to periodic offline backups of the data. Relational databases, such as SQL Server and Oracle, provide various options for replication. For SQL Server, use SQL Server Always On Availability Groups for high availability.

NoSQL databases, like MongoDB, also support replicas for redundancy. The replicas for high availability are used.

Scenario 2: A cluster of redundant VMs

In this scenario, you have a workload handled by a cluster of VMs that provide redundancy and load balancing, like a Cassandra cluster deployed in a region. This type of architecture already provides a high level of redundancy within that region. However, to protect the workload from a regional-level failure, consider spreading the cluster across two regions or making periodic backups to another region.

Scenario 3: IaaS application workload

In this scenario, you have an application with a typical production workload running on an Azure VM (IaaS application workload). It could be a web server or file server holding content and other resources of a site, or a custom-built business application running on a VM that stored its data, resources, and application state on the VM disks. In this case, it's important to make sure you take backups regularly. Backup frequency should be based on the nature of the VM workload. For example, if the application runs daily and modifies data, then the backup should be taken hourly.

Another common scenario involves a reporting server that pulls data from other sources and generates aggregated reports. The loss of this VM or disks could lead to the loss of the reports. However, it's possible to regenerate the output by rerunning the reporting process. In this case, you don't really experience data loss, even if the reporting server is hit with a disaster. So you have a higher level of tolerance for losing part of the data on the reporting server. In this case, taking backups less often would reduce your cost.

Scenario 4: IaaS application data issues

IaaS application data issues are another possibility. Consider an application that computes, maintains, and serves critical commercial data, such as pricing information. A new version of your application had a software bug that incorrectly computed the pricing and corrupted the existing commerce data served by the platform. Here, the best course of action is to revert to the earlier version of the application and the data. To enable this, take periodic backups of your system.

Backup and Disaster Recovery Solutions

Comparison Overview

This section covers some of Azure's options for backup and disaster recovery. You can refer to the following comparison table for a high level overview.

Expand table

Solution	Snapshot	Restore Points	Azure Backup	Azure Site Recovery
Description	Snapshot is a ready-only poin-in-time copy of the disk that you can use for backup.	Restore Points can be used to implement granular backup of all disks attached to your Virtual Machine	Azure Backup is a fully managed Azure service to provide simple, secure, and cost-effective solution to back up your data and recover it	Azure Site Recovery helps ensure your organization's business continuity by keeping apps and workloads running during outages
Incremental Backup	Yes	Yes	Yes	Yes
Cross-Region Copy	Yes	Available in public preview	Yes, with Azure VM backup	Yes

Solution	Snapshot	Restore Points	Azure Backup	Azure Site Recovery
Pricing	See Azure Disk Pricing	N/A	See Estimate costs for backing up Azure VMs or on-premises servers	See Azure Site Recovery Pricing
Operational Maintenance	High	Medium	Low	Low
Key benefits	Most cost effective, suitable for disk backup	Back up disks at VM level, Application consistent for VM running Windows OS, File system consistency for VMs running Linux OS	Frequent and quick backup without interrupting the VM, custom backup policy, agentless solution	Simple BCDR solution, keep application consistency over failover, orchestrate replication without intercepting application data

Snapshots

A snapshot is a read-only point in time copy of the disk. You can take a snapshot at any time. Snapshots exist independently of the source disk and can only be used to create new managed disks. You can't use them to change the state of an existing disk. You can also use incremental snapshots for periodic incremental backup of your managed disks.

Generally, you should use incremental snapshots for backup and disaster recovery purposes since they lower your costs and offer a faster recovery time. Incremental snapshots are point-in-time backups for managed disks that, when taken, consist only of the changes since the last snapshot. The first incremental snapshot is a full copy of the disk. Any incremental snapshots after the first one consists only of the changes to a disk since the last snapshot. When you create a disk from a snapshot, the system reconstructs the full disk that represents the point in time backup of the disk when the incremental snapshot was taken. You can also Copy an incremental snapshot to a new region for disaster recovery.

You can implement backup mechanisms through snapshots. To do this, you need to create consistent snapshots for all disks used by a VM and then replicate them to another region. An option to create consistent backups with snapshot is to shut down the VM and take snapshots of each disk. If applications running on the VM can pause the IOs, you should take advantage of pausing it and then take snapshots of all the disks attached to VMs. Taking offline snapshots is easier than coordinating snapshots of a running VM, but it requires a few minutes of downtime.

You can take a snapshot anytime, but if you're taking snapshots while the VM is running, keep these things in mind:

- When the VM is running, data is still being streamed to the disks. As a result, snapshots of a running VM might contain partial operations that were in flight.
- If there are several disks involved in a VM, snapshots of different disks might have occurred at different times.

In the described scenario, snapshots weren't coordinated. This lack of coordination is a problem for striped volumes whose files might be corrupted if changes were being made during a backup. So the backup process must implement the following steps:

1. Freeze all the disks.
2. Flush all the pending writes.
3. Create an incremental snapshot for managed disks for all the disks.

Some Windows applications, like SQL Server, provide a coordinated backup mechanism through a volume shadow service to create application-consistent backups. On Linux, you can use a tool like fsfreeze to coordinate disks (this tool provides file-consistent backups, not application-consistent snapshots). This backup procedure is complex, so you should consider Overview of Azure Disk Backup or a third-party backup solution that already implements this procedure. This would result a collection of coordinated snapshots for all the VM disks, representing a specific point-in-time view of the VM - in other words, a backup restore point for the VM. You can repeat the process at scheduled intervals to create periodic backups.

Restore Points

Azure VM Restore Points can be used to implement granular backup and retention policies of all disks attached to your virtual machine. Individual VM restore point is a resource that stores VM configuration and point-in-time application consistent snapshots of all the managed disks attached to the VM. You can use VM Restore Points to easily capture multi-disk consistent backups of all disks attached to your VM.

Restore points have three levels of hierarchy – VM Restore Point Collection, VM Restore Points, and Disk Restore Points:

- Level 1: VM Restore Points are organized into Restore Point Collections. A Restore Point collection is an Azure Resource Management resource that contains the restore points for a specific VM.
- Level 2: VM Restore Points contain a disk restore point for each of the attached disks.
- Level 3: A Disk Restore Point consists of a snapshot of an individual managed disk.

Restore points are incremental. First restore point stores a full copy of all disks attached to the VM, while successive restore point only contains incremental changes to VM disks. With Restore Points, you can:

- Copy VM restore points between regions, restore VMs in a different region than the source VM and track the progress of copy operation.
- Create disks using disk restore points and get a shared access signature for the disk. These disks can then be used to create a new VM.

See the following articles to learn how to Create VM restore points and Manage VM restore points.

Azure Backup

Azure Backup provides simple, secure, and cost-effective solutions to backup your data and recover it from Azure. Azure Disk Backup is a native, cloud-based backup solution that protects your data in managed disks. It's a simple, secure, and cost-effective solution that enables you to configure protection for managed disks in a few steps. It ensures your data is protected in the event of a disaster.

Azure Disk Backup offers a turnkey solution that provides snapshot lifecycle management for managed disks by automating periodic creation of snapshots and retaining it for however long you specify, using backup policy. You can manage disk snapshots, with no infrastructure costs, without the need for custom scripting, or any management overhead.

Azure Disk Backup is a crash-consistent backup solution that takes point-in-time backup of a managed disk using incremental snapshots and supports multiple backups per day. It's also an agent-less solution, and doesn't impact production application performance. It supports backup and restore of both OS and data disks (including shared disks), whether they're currently attached to a running Azure VM or not.

Azure Disk Backup is integrated into Backup Center, which provides a single unified management experience in Azure for enterprises to govern, monitor, operate, and analyze backups at scale. If you need an application-consistent backup of virtual machine including the data disks, or an option to restore an entire virtual machine from backup, restore a file or folder, or restore to a secondary region, use the Azure VM backup solution. If you're unable to use Azure Backup, you can implement your own backup mechanism by using snapshots. But creating consistent snapshots for all the disks used by a VM, replicating those snapshots to another region, and continually managing this process is complicated and time consuming.

Azure Site Recovery

Azure Site Recovery helps ensure your organization's business continuity by keeping apps and workloads running during outages. It's a paid, fully managed service to help you achieve your business continuity and disaster recovery (BCDR) strategy.

Azure Site Recovery replicates workloads running on physical and virtual machines from a primary site to a secondary location. When an outage occurs at your primary site, your workload failover to a secondary location and can be accessed from there. After the primary location is running again, your workloads can fail back to it.

You can easily set up disaster recovery to a secondary Azure region with a few steps. Azure Site Recovery enables many disaster recovery scenarios – Azure to Azure, VMware to Azure, Physical to Azure, Azure Stack VM, Hyper-V to Azure, DR for apps, DR to a secondary site. For a full list of benefits that Azure Site Recovery provides, refer to About Site Recovery.

Other options

SQL Server running in a VM has its own built-in capabilities to back up your SQL Server database to Azure Blob storage or a file share. For more information, see Back up and restore for SQL Server in Azure virtual machines. In addition to back up and restore, SQL Server Always On availability groups can maintain secondary replicas of databases. This ability greatly reduces the disaster recovery time.

Next steps

Explore your options:

- Create an incremental snapshot for managed disks
- Copy an incremental snapshot to a new region
- Overview of VM restore points
- Overview of Azure Disk Backup
- About Site Recovery

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a snapshot of a virtual hard disk

Article • 04/01/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

A snapshot is a full, read-only copy of a virtual hard disk (VHD). You can use a snapshot as a point-in-time backup, or to help troubleshoot virtual machine (VM) issues. You can take a snapshot of both operating system (OS) or data disk VHDs. With snapshots, you can back up your managed disks at any point in time. These snapshots exist independently of the source disk, and you can use them to create new managed disks.

Snapshots are billed based on the used size. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GiB and an actual used data size of 10 GiB, that snapshot is billed only for the used data size of 10 GiB. You can see the used size of your snapshots by checking the Azure usage report. For example, if the used data size of a snapshot is 10 GiB, the *daily* usage report shows 10 GiB/(31 days) = 0.3226 as the consumed quantity. Snapshots are billed separately from their original disk. For details, see the pricing page.

Create a snapshot of a VHD

If you want to use a snapshot to create a new VM, ensure that you first cleanly shut down the VM. This action clears any processes that are in progress.

Portal

To create a snapshot using the Azure portal, complete these steps.

1. In the Azure portal, select **Create a resource**.
2. Search for and select **Snapshot**.
3. In the **Snapshot** window, select **Create**. The **Create snapshot** window appears.
4. For **Resource group**, select an existing resource group or enter the name of a new one.
5. Enter a **Name**, then select a **Region** and **Snapshot type** for the new snapshot. If you would like to store your snapshot in zone-resilient storage, you need to select a region that supports availability zones. For a list of supporting regions, see Azure regions with availability zones.
6. For **Source subscription**, select the subscription that contains the managed disk to be backed up.
7. For **Source disk**, select the managed disk to snapshot.

8. For **Storage type**, select **Standard HDD**, unless you require zone-redundant storage or high-performance storage for your snapshot.
9. If needed, configure settings on the **Encryption**, **Networking**, and **Tags** tabs. Otherwise, default settings are used for your snapshot.
10. Select **Review + create**.

Next steps

To recover using a snapshot, you must create a new disk from the snapshot, then either deploy a new VM, and use the managed disk as the OS disk, or attach the disk as a data disk to an existing VM.

Portal

For more information, see the example in Create a VM from a VHD by using the Azure portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Back up Azure unmanaged Virtual Machine disks with incremental snapshots

Article • 09/25/2024

Overview

Azure Storage provides the capability to take snapshots of blobs. Snapshots capture the blob state at that point in time. In this article, we describe a scenario in which you can maintain backups of virtual machine disks using snapshots. You can use this methodology when you choose not to use Azure Backup and Recovery Service, and wish to create a custom backup strategy for your virtual machine disks. For virtual machines running business or mission critical workloads, it's recommended to use Azure Backup as part of the backup strategy.

Azure virtual machine disks are stored as page blobs in Azure Storage. Since we are describing a backup strategy for virtual machine disks in this article, we refer to snapshots in the context of page blobs. To learn more about snapshots, refer to Creating a Snapshot of a Blob.

What is a snapshot?

A blob snapshot is a read-only version of a blob that is captured at a point in time. Once a snapshot has been created, it can be read, copied, or deleted, but not modified. Snapshots provide a way to back up a blob as it appears at a moment in time. Until REST version 2015-04-05, you had the ability to copy full snapshots. With the REST version 2015-07-08 and above, you can also copy incremental snapshots.

Full snapshot copy

Snapshots can be copied to another storage account as a blob to keep backups of the base blob. You can also copy a snapshot over its base blob, which is like restoring the blob to an earlier version. When a snapshot is copied from one storage account to another, it occupies the same space as the base page blob. Therefore, copying whole snapshots from one storage account to another is slow and consumes much space in the target storage account.

> **Note**
>
> If you copy the base blob to another destination, the snapshots of the blob are not copied along with it. Similarly, if you overwrite a base blob with a copy, snapshots associated with the base blob are not affected and stay intact under the base blob name.

Back up disks using snapshots

As a backup strategy for your virtual machine disks, you can take periodic snapshots of the disk or page blob, and copy them to another storage account using tools like Copy Blob operation or AzCopy. You can copy a snapshot to a destination page blob with a different name. The resulting destination page blob is a writeable page blob and not a snapshot. Later in this article, we describe steps to take backups of virtual machine disks using snapshots.

Restore disks using snapshots

When it is time to restore your disk to a stable version that was previously captured in one of the backup snapshots, you can copy a snapshot over the base page blob. After the snapshot is promoted to the base page blob, the snapshot remains, but its source is overwritten with a copy that can be both read and written. Later in this article we describe steps to restore a previous version of your disk from its snapshot.

Implementing full snapshot copy

You can implement a full snapshot copy by doing the following,

- First, take a snapshot of the base blob using the Snapshot Blob operation.
- Then, copy the snapshot to a target storage account using Copy Blob.
- Repeat this process to maintain backup copies of your base blob.

Incremental snapshot copy

The new feature in the GetPageRanges API provides a much better way to back up the snapshots of your page blobs or disks. The API returns the list of changes between the base blob and the snapshots, which reduces the amount of storage space used on the backup account. The API supports page blobs on Premium Storage as well as Standard

Storage. Using this API, you can build faster and more efficient backup solutions for Azure VMs. This API will be available with the REST version 2015-07-08 and higher.

Incremental Snapshot Copy allows you to copy from one storage account to another the difference between,

- Base blob and its Snapshot OR
- Any two snapshots of the base blob

Provided the following conditions are met,

- The blob was created on Jan-1-2016 or later.
- The blob was not overwritten with PutPage or Copy Blob between two snapshots.

> ⓘ **Note**
>
> This feature is available for Premium and Standard Azure Page Blobs.

When you have a custom backup strategy using snapshots, copying the snapshots from one storage account to another can be slow and can consume much storage space. Instead of copying the entire snapshot to a backup storage account, you can write the difference between consecutive snapshots to a backup page blob. This way, the time to copy and the space to store backups is substantially reduced.

Implementing Incremental Snapshot Copy

You can implement incremental snapshot copy by doing the following,

- Take a snapshot of the base blob using Snapshot Blob.
- Copy the snapshot to the target backup storage account in same or any other Azure region using Copy Blob. This is the backup page blob. Take a snapshot of the backup page blob and store it in the backup account.
- Take another snapshot of the base blob using Snapshot Blob.
- Get the difference between the first and second snapshots of the base blob using GetPageRanges. Use the new parameter **prevsnapshot**, to specify the DateTime value of the snapshot you want to get the difference with. When this parameter is present, the REST response includes only the pages that were changed between target snapshot and previous snapshot including clear pages.
- Use PutPage to apply these changes to the backup page blob.
- Finally, take a snapshot of the backup page blob and store it in the backup storage account.

In the next section, we will describe in more detail how you can maintain backups of disks using Incremental Snapshot Copy

Scenario

In this section, we describe a scenario that involves a custom backup strategy for virtual machine disks using snapshots.

Consider a DS-series Azure VM with a premium storage P30 disk attached. The P30 disk called *mypremiumdisk* is stored in a premium storage account called *mypremiumaccount*. A standard storage account called *mybackupstdaccount* is used for storing the backup of *mypremiumdisk*. We would like to keep a snapshot of *mypremiumdisk* every 12 hours.

To learn about creating a storage account, see Create a storage account.

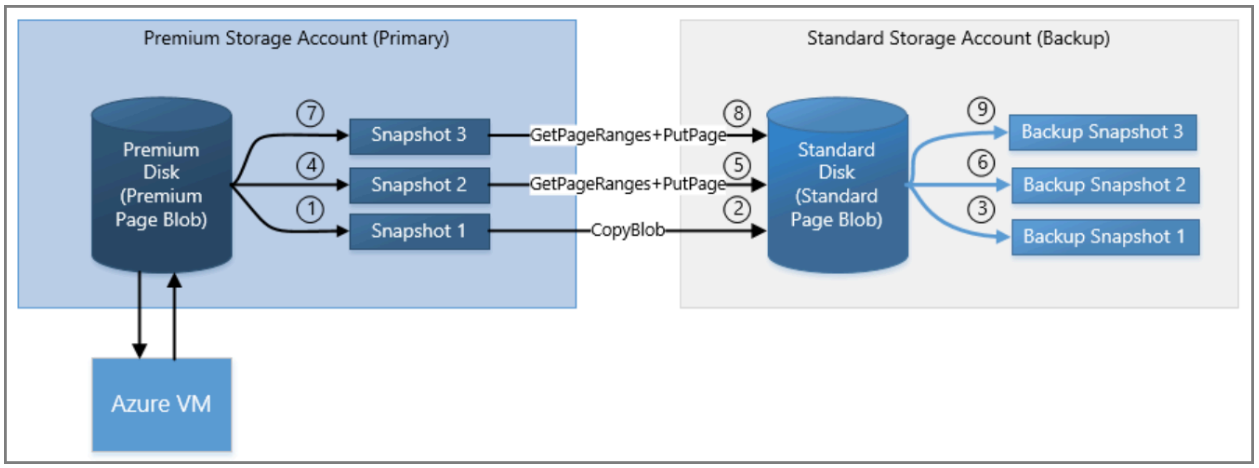
To learn about backing up Azure VMs, refer to Plan Azure VM backups.

Steps to maintain backups of a disk using incremental snapshots

The following steps describe how to take snapshots of *mypremiumdisk* and maintain the backups in *mybackupstdaccount*. The backup is a standard page blob called *mybackupstdpageblob*. The backup page blob always reflects the same state as the last snapshot of *mypremiumdisk*.

1. Create the backup page blob for your premium storage disk, by taking a snapshot of *mypremiumdisk* called *mypremiumdisk_ss1*.
2. Copy this snapshot to mybackupstdaccount as a page blob called *mybackupstdpageblob*.
3. Take a snapshot of *mybackupstdpageblob* called *mybackupstdpageblob_ss1*, using Snapshot Blob and store it in *mybackupstdaccount*.
4. During the backup window, create another snapshot of *mypremiumdisk*, say *mypremiumdisk_ss2*, and store it in *mypremiumaccount*.
5. Get the incremental changes between the two snapshots, *mypremiumdisk_ss2* and *mypremiumdisk_ss1*, using GetPageRanges on *mypremiumdisk_ss2* with the **prevsnapshot** parameter set to the timestamp of *mypremiumdisk_ss1*. Write these incremental changes to the backup page blob *mybackupstdpageblob* in *mybackupstdaccount*. If there are deleted ranges in the incremental changes, they must be cleared from the backup page blob. Use PutPage to write incremental changes to the backup page blob.

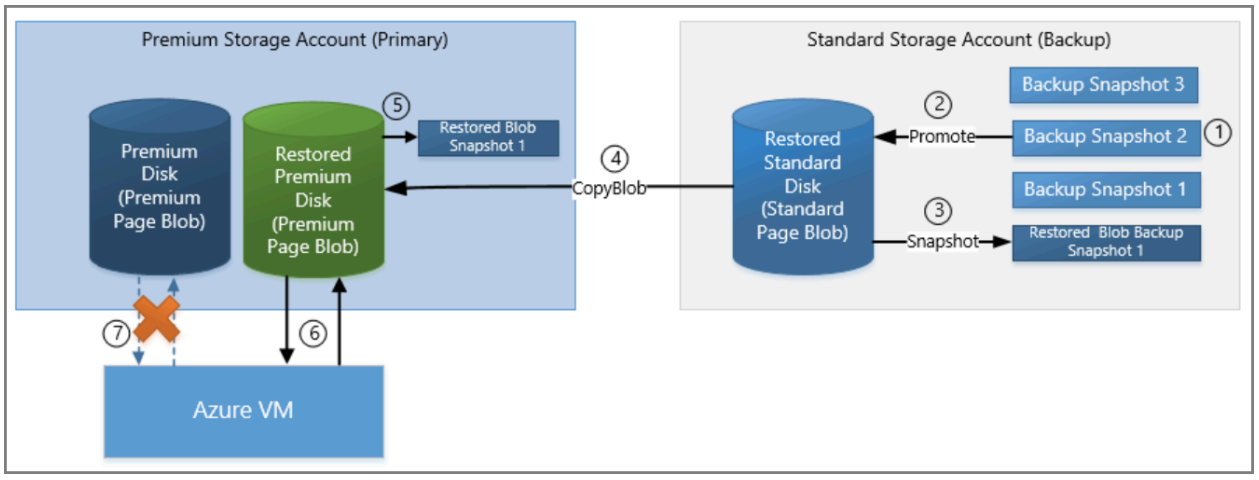
6. Take a snapshot of the backup page blob *mybackupstdpageblob*, called *mybackupstdpageblob_ss2*. Delete the previous snapshot *mypremiumdisk_ss1* from premium storage account.
7. Repeat steps 4-6 every backup window. In this way, you can maintain backups of *mypremiumdisk* in a standard storage account.



Steps to restore a disk from snapshots

The following steps, describe how to restore the premium disk, *mypremiumdisk* to an earlier snapshot from the backup storage account *mybackupstdaccount*.

1. Identify the point in time that you wish to restore the premium disk to. Let's say that it is snapshot *mybackupstdpageblob_ss2*, which is stored in the backup storage account *mybackupstdaccount*.
2. In mybackupstdaccount, promote the snapshot *mybackupstdpageblob_ss2* as the new backup base page blob *mybackupstdpageblobrestored*.
3. Take a snapshot of this restored backup page blob, called *mybackupstdpageblobrestored_ss1*.
4. Copy the restored page blob *mybackupstdpageblobrestored* from *mybackupstdaccount* to *mypremiumaccount* as the new premium disk *mypremiumdiskrestored*.
5. Take a snapshot of *mypremiumdiskrestored*, called *mypremiumdiskrestored_ss1* for making future incremental backups.
6. Point the DS series VM to the restored disk *mypremiumdiskrestored* and detach the old *mypremiumdisk* from the VM.
7. Begin the Backup process described in previous section for the restored disk *mypremiumdiskrestored*, using the *mybackupstdpageblobrestored* as the backup page blob.



Next steps

Use the following links to learn more about creating snapshots of a blob and planning your VM backup infrastructure.

- Creating a Snapshot of a Blob
- Plan your VM Backup Infrastructure

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Backup and disaster recovery for Azure unmanaged disks

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This article explains how to plan for backup and disaster recovery (DR) of IaaS virtual machines (VMs) and disks in Azure. This document covers unmanaged disks, or page blobs.

First, we cover the built-in fault tolerance capabilities in the Azure platform that helps guard against local failures. We then discuss the disaster scenarios not fully covered by the built-in capabilities. We also show several examples of workload scenarios where different backup and DR considerations can apply. We then review possible solutions for the DR of IaaS disks.

Introduction

The Azure platform uses various methods for redundancy and fault tolerance to help protect customers from localized hardware failures. Local failures can include problems with an Azure Storage server machine that stores part of the data for a virtual disk or failures of an SSD or HDD on that server. Such isolated hardware component failures can happen during normal operations.

The Azure platform is designed to be resilient to these failures. Major disasters can result in failures or the inaccessibility of many storage servers or even a whole datacenter. Although your VMs and disks are normally protected from localized failures, additional steps are necessary to protect your workload from region-wide catastrophic failures, such as a major disaster, that can affect your VM and disks.

In addition to the possibility of platform failures, problems with a customer application or data can occur. For example, a new version of your application might inadvertently make a change to the data that causes it to break. In that case, you might want to revert the application and the data to a prior version that contains the last known good state. This requires maintaining regular backups.

For regional disaster recovery, you must back up your IaaS VM disks to a different region.

Before we look at backup and DR options, let’s recap a few methods available for handling localized failures.

Azure IaaS resiliency

Resiliency refers to the tolerance for normal failures that occur in hardware components. Resiliency is the ability to recover from failures and continue to function. It's not about avoiding failures, but responding to failures in a way that avoids downtime or data loss. The goal of resiliency is to return the application to a fully functioning state following a failure. Azure virtual machines and disks are designed to be resilient to common hardware faults. Let's look at how the Azure IaaS platform provides this resiliency.

A virtual machine consists mainly of two parts: a compute server and the persistent disks. Both affect the fault tolerance of a virtual machine.

If the Azure compute host server that houses your VM experiences a hardware failure, which is rare, Azure is designed to automatically restore the VM on another server. If this scenario, your computer reboots, and the VM comes back up after some time. Azure automatically detects such hardware failures and executes recoveries to help ensure the customer VM is available as soon as possible.

Regarding IaaS disks, the durability of data is critical for a persistent storage platform. Azure customers have important business applications running on IaaS, and they depend on the persistence of the data. Azure designs protection for these IaaS disks, with three redundant copies of the data that is stored locally. These copies provide for high durability against local failures. If one of the hardware components that holds your disk fails, your VM is not affected, because there are two additional copies to support disk requests. It works fine, even if two different hardware components that support a disk fail at the same time (which is rare).

To ensure that you always maintain three replicas, Azure Storage automatically spawns a new copy of the data in the background if one of the three copies becomes unavailable. Therefore, it should not be necessary to use RAID with Azure disks for fault tolerance. A simple RAID 0 configuration should be sufficient for striping the disks, if necessary, to create larger volumes.

Because of this architecture, Azure has consistently delivered enterprise-grade durability for IaaS disks, with an industry-leading zero percent annualized failure rate.

Localized hardware faults on the compute host or in the Storage platform can sometimes result in of the temporary unavailability of the VM that is covered by the Azure SLA for VM availability. Azure also provides an industry-leading SLA for single VM instances that use Azure premium SSDs.

To safeguard application workloads from downtime due to the temporary unavailability of a disk or VM, customers can use availability sets. Two or more virtual machines in an

availability set provide redundancy for the application. Azure then creates these VMs and disks in separate fault domains with different power, network, and server components.

Because of these separate fault domains, localized hardware failures typically do not affect multiple VMs in the set at the same time. Having separate fault domains provides high availability for your application. It's considered a good practice to use availability sets when high availability is required. The next section covers the disaster recovery aspect.

Backup and disaster recovery

Disaster recovery is the ability to recover from rare, but major, incidents. These incidents include non-transient, wide-scale failures, such as service disruption that affects an entire region. Disaster recovery includes data backup and archiving, and might include manual intervention, such as restoring a database from a backup.

The Azure platform's built-in protection against localized failures might not fully protect the VMs/disks if a major disaster causes large-scale outages. These large-scale outages include catastrophic events, such as if a datacenter is hit by a hurricane, earthquake, fire, or if there is a large-scale hardware unit failure. In addition, you might encounter failures due to application or data issues.

To help protect your IaaS workloads from outages, you should plan for redundancy and have backups to enable recovery. For disaster recovery, you should back up in a different geographic location away from the primary site. This approach helps ensure your backup is not affected by the same event that originally affected the VM or disks. For more information, see Disaster recovery for Azure applications.

Your DR considerations might include the following aspects:

- High availability: The ability of the application to continue running in a healthy state, without significant downtime. By *healthy state*, this state means that the application is responsive, and users can connect to the application and interact with it. Certain mission-critical applications and databases might be required to always be available, even when there are failures in the platform. For these workloads, you might need to plan redundancy for the application, as well as the data.

- Data durability: In some cases, the main consideration is ensuring that the data is preserved if a disaster happens. Therefore, you might need a backup of your data in a different site. For such workloads, you might not need full redundancy for the application, but only a regular backup of the disks.

Backup and DR scenarios

Let's look at a few typical examples of application workload scenarios and the considerations for planning for disaster recovery.

Scenario 1: Major database solutions

Consider a production database server, like SQL Server or Oracle, that can support high availability. Critical production applications and users depend on this database. The disaster recovery plan for this system might need to support the following requirements:

- The data must be protected and recoverable.
- The server must be available for use.

The disaster recovery plan might require maintaining a replica of the database in a different region as a backup. Depending on the requirements for server availability and data recovery, the solution might range from an active-active or active-passive replica site to periodic offline backups of the data. Relational databases, such as SQL Server and Oracle, provide various options for replication. For SQL Server, use SQL Server Always On Availability Groups for high availability.

NoSQL databases, like MongoDB, also support replicas for redundancy. The replicas for high availability are used.

Scenario 2: A cluster of redundant VMs

Consider a workload handled by a cluster of VMs that provide redundancy and load balancing. One example is a Cassandra cluster deployed in a region. This type of architecture already provides a high level of redundancy within that region. However, to protect the workload from a regional-level failure, you should consider spreading the cluster across two regions or making periodic backups to another region.

Scenario 3: IaaS application workload

Let's look at the IaaS application workload. For example, this application might be a typical production workload running on an Azure VM. It might be a web server or file server holding the content and other resources of a site. It might also be a custom-built business application running on a VM that stored its data, resources, and application state on the VM disks. In this case, it's important to make sure you take backups on a regular basis. Backup frequency should be based on the nature of the VM workload. For

example, if the application runs every day and modifies data, then the backup should be taken every hour.

Another example is a reporting server that pulls data from other sources and generates aggregated reports. The loss of this VM or disks might lead to the loss of the reports. However, it might be possible to rerun the reporting process and regenerate the output. In that case, you don't really have a loss of data, even if the reporting server is hit with a disaster. As a result, you might have a higher level of tolerance for losing part of the data on the reporting server. In that case, less frequent backups are an option to reduce costs.

Scenario 4: IaaS application data issues

IaaS application data issues are another possibility. Consider an application that computes, maintains, and serves critical commercial data, such as pricing information. A new version of your application had a software bug that incorrectly computed the pricing and corrupted the existing commerce data served by the platform. Here, the best course of action is to revert to the earlier version of the application and the data. To enable this, take periodic backups of your system.

Disaster recovery solution: Azure Backup

Azure Backup is used for backups and DR, and it works with managed disks as well as unmanaged disks. You can create a backup job with time-based backups, easy VM restoration, and backup retention policies.

If you use premium SSDs, managed disks, or other disk types with the locally redundant storage option, it's especially important to make periodic DR backups. Azure Backup stores the data in your recovery services vault for long-term retention. Choose the geo-redundant storage option for the backup recovery services vault. That option ensures that backups are replicated to a different Azure region for safeguarding from regional disasters.

For unmanaged disks, you can use the locally redundant storage type for IaaS disks, but ensure that Azure Backup is enabled with the geo-redundant storage option for the recovery services vault.

> **Note**
>
> If you use the geo-redundant storage or read-access geo-redundant storage option for your unmanaged disks, you still need consistent snapshots for backup

and DR. Use either Azure Backup or consistent snapshots.

The following table is a summary of the solutions available for DR.

Expand table

Scenario	Automatic replication	DR solution
Premium SSD disks	Local (locally redundant storage)	Azure Backup
Managed disks	Local (locally redundant storage)	Azure Backup
Unmanaged locally redundant storage disks	Local (locally redundant storage)	Azure Backup
Unmanaged geo-redundant storage disks	Cross region (geo-redundant storage)	Azure Backup Consistent snapshots
Unmanaged read-access geo-redundant storage disks	Cross region (read-access geo-redundant storage)	Azure Backup Consistent snapshots

High availability is best met by using managed disks in an availability set along with Azure Backup. If you use unmanaged disks, you can still use Azure Backup for DR. If you are unable to use Azure Backup, then taking consistent snapshots, as described in a later section, is an alternative solution for backup and DR.

Your choices for high availability, backup, and DR at application or infrastructure levels can be represented as follows:

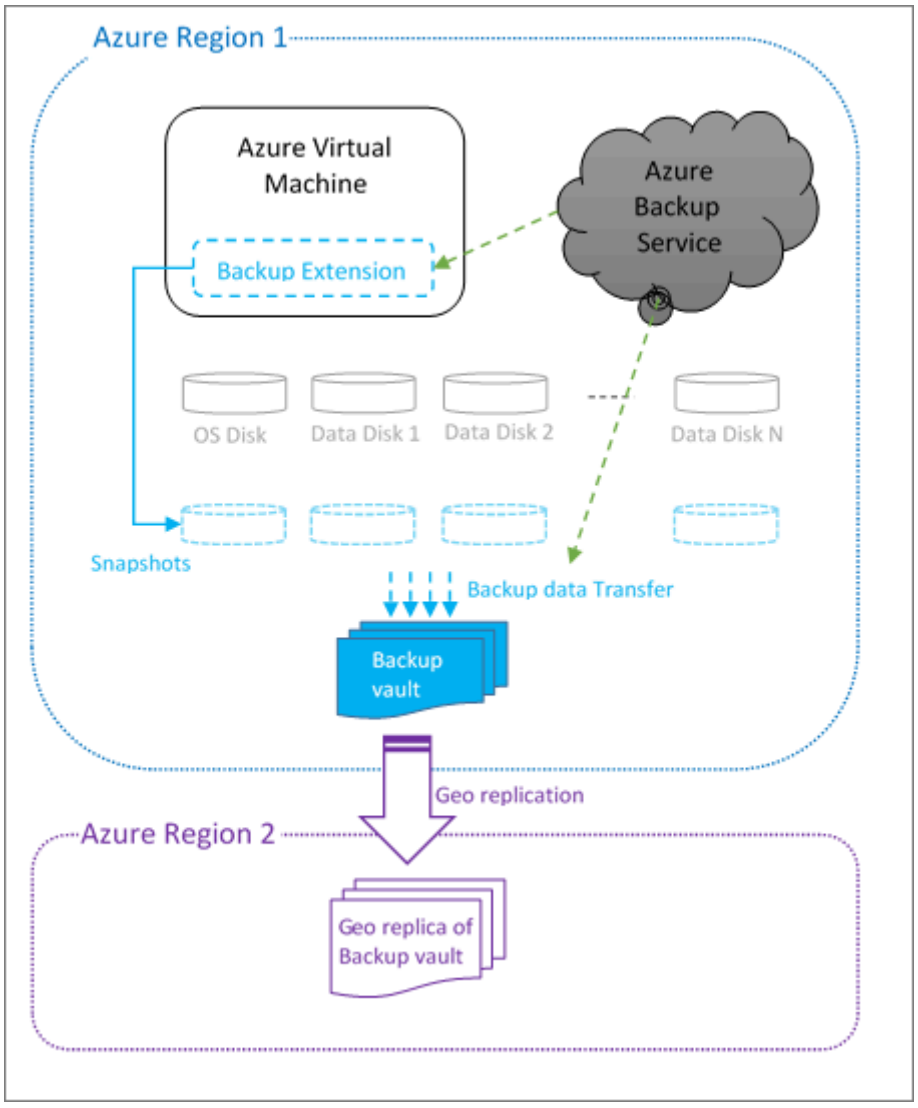
Expand table

Level	High availability	Backup or DR
Application	SQL Server Always On	Azure Backup
Infrastructure	Availability set	Geo-redundant storage with consistent snapshots

Using Azure Backup

Azure Backup can back up your VMs running Windows or Linux to the Azure recovery services vault. Backing up and restoring business-critical data is complicated by the fact that business-critical data must be backed up while the applications that produce the data are running.

To address this issue, Azure Backup provides application-consistent backups for Microsoft workloads. It uses the volume shadow service to ensure that data is written correctly to storage. For Linux VMs, the default backup consistency mode is file-consistent backups, because Linux does not have functionality equivalent to the volume shadow service as in the case of Windows. For Linux machines, see Application-consistent backup of Azure Linux VMs.



When Azure Backup initiates a backup job at the scheduled time, it triggers the backup extension installed in the VM to take a point-in-time snapshot. A snapshot is taken in coordination with the volume shadow service to get a consistent snapshot of the disks in the virtual machine without having to shut it down. The backup extension in the VM flushes all writes before taking a consistent snapshot of all of the disks. After taking the snapshot, the data is transferred by Azure Backup to the backup vault. To make the backup process more efficient, the service identifies and transfers only the blocks of data that have changed after the last backup.

To restore, you can view the available backups through Azure Backup and then initiate a restore. You can create and restore Azure backups through the Azure portal, by using PowerShell, or by using the Azure CLI.

Steps to enable a backup

Use the following steps to enable backups of your VMs by using the Azure portal. There is some variation depending on your exact scenario. Refer to the Azure Backup documentation for full details. Azure Backup also supports VMs with managed disks.

1. Create a recovery services vault for a VM:

   a. In the Azure portal, browse **All resources** and find **Recovery Services vaults**.

   b. On the **Recovery Services vaults** menu, click **Add** and follow the steps to create a new vault in the same region as the VM. For example, if your VM is in the West US region, pick West US for the vault.

2. Verify the storage replication for the newly created vault. Access the vault under **Recovery Services vaults** and go to **Properties** > **Backup Configuration** > **Update**. Ensure the **geo-redundant storage** option is selected by default. This option ensures that your vault is automatically replicated to a secondary datacenter. For example, your vault in West US is automatically replicated to East US.

3. Configure the backup policy and select the VM from the same UI.

4. Make sure the Backup Agent is installed on the VM. If your VM is created by using an Azure gallery image, then the Backup Agent is already installed. Otherwise (that is, if you use a custom image), use the instructions to install the VM agent on a virtual machine.

5. After the previous steps are completed, the backup runs at regular intervals as specified in the backup policy. If necessary, you can trigger the first backup manually from the vault dashboard on the Azure portal.

For automating Azure Backup by using scripts, refer to PowerShell cmdlets for VM backup.

Steps for recovery

If you need to repair or rebuild a VM, you can restore the VM from any of the backup recovery points in the vault. There are a couple of different options for performing the recovery:

- You can create a new VM as a point-in-time representation of your backed-up VM.
- You can restore the disks, and then use the template for the VM to customize and rebuild the restored VM.

For more information, see the instructions to use the Azure portal to restore virtual machines. This document also explains the specific steps for restoring backed-up VMs to a paired datacenter by using your geo-redundant backup vault if there is a disaster at the primary datacenter. In that case, Azure Backup uses the Compute service from the secondary region to create the restored virtual machine.

You can also use PowerShell for creating a new VM from restored disks.

Alternative solution: Consistent snapshots

If you are unable to use Azure Backup, you can implement your own backup mechanism by using snapshots. Creating consistent snapshots for all the disks used by a VM and then replicating those snapshots to another region is complicated. For this reason, Azure considers using the Backup service as a better option than building a custom solution.

If you use read-access geo-redundant storage/geo-redundant storage for disks, snapshots are automatically replicated to a secondary datacenter. If you use locally redundant storage for disks, you need to replicate the data yourself. For more information, see Back up Azure-unmanaged VM disks with incremental snapshots.

A snapshot is a representation of an object at a specific point in time. A snapshot incurs billing for the incremental size of the data it holds. For more information, see Create a blob snapshot.

Create snapshots while the VM is running

Although you can take a snapshot at any time, if the VM is running, there is still data being streamed to the disks. The snapshots might contain partial operations that were in flight. Also, if there are several disks involved, the snapshots of different disks might have occurred at different times. These scenarios may cause to the snapshots to be uncoordinated. This lack of coordination is especially problematic for striped volumes whose files might be corrupted if changes were being made during backup.

To avoid this situation, the backup process must implement the following steps:

1. Freeze all the disks.
2. Flush all the pending writes.
3. Create a blob snapshot for all the disks.

Some Windows applications, like SQL Server, provide a coordinated backup mechanism via a volume shadow service to create application-consistent backups. On Linux, you can

use a tool like *fsfreeze* for coordinating the disks. This tool provides file-consistent backups, but not application-consistent snapshots. This process is complex, so you should consider using Azure Backup or a third-party backup solution that already implements this procedure.

The previous process results in a collection of coordinated snapshots for all of the VM disks, representing a specific point-in-time view of the VM. This is a backup restore point for the VM. You can repeat the process at scheduled intervals to create periodic backups. See Copy the backups to another region for steps to copy the snapshots to another region for DR.

Create snapshots while the VM is offline

Another option to create consistent backups is to shut down the VM and take blob snapshots of each disk. Taking blob snapshots is easier than coordinating snapshots of a running VM, but it requires a few minutes of downtime.

1. Shut down the VM.
2. Create a snapshot of each virtual hard drive blob, which only takes a few seconds.

   To create a snapshot, you can use PowerShell, the Azure Storage REST API, Azure CLI, or one of the Azure Storage client libraries, such as the Storage client library for .NET.
3. Start the VM, which ends the downtime. Typically, the entire process finishes within a few minutes.

This process yields a collection of consistent snapshots for all the disks, providing a backup restore point for the VM.

Copy the snapshots to another region

Creation of the snapshots alone might not be sufficient for DR. You must also replicate the snapshot backups to another region.

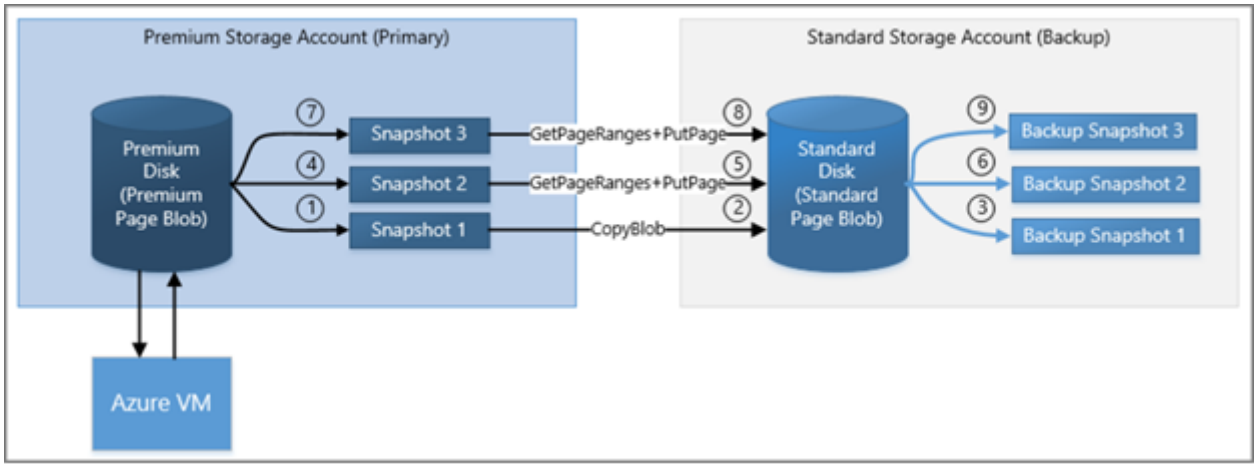
If you use geo-redundant storage or read-access geo-redundant storage for your disks, then the snapshots are replicated to the secondary region automatically. There can be a few minutes of lag before the replication. If the primary datacenter goes down before the snapshots finish replicating, you cannot access the snapshots from the secondary datacenter. The likelihood of this is small.

> **Note**

> Only having the disks in a geo-redundant storage or read-access geo-redundant storage account does not protect the VM from disasters. You must also create coordinated snapshots or use Azure Backup. This is required to recover a VM to a consistent state.

If you use locally redundant storage, you must copy the snapshots to a different storage account immediately after creating the snapshot. The copy target might be a locally redundant storage account in a different region, resulting in the copy being in a remote region. You can also copy the snapshot to a read-access geo-redundant storage account in the same region. In this case, the snapshot is lazily replicated to the remote secondary region. Your backup is protected from disasters at the primary site after the copying and replication is complete.

To copy your incremental snapshots for DR efficiently, review the instructions in Back up Azure unmanaged VM disks with incremental snapshots.



Recovery from snapshots

To retrieve a snapshot, copy it to make a new blob. If you are copying the snapshot from the primary account, you can copy the snapshot over to the base blob of the snapshot. This process reverts the disk to the snapshot. This process is known as promoting the snapshot. If you are copying the snapshot backup from a secondary account, in the case of a read-access geo-redundant storage account, you must copy it to a primary account. You can copy a snapshot by using PowerShell or by using the AzCopy utility. For more information, see Transfer data with the AzCopy command-line utility.

For VMs with multiple disks, you must copy all the snapshots that are part of the same coordinated restore point. After you copy the snapshots to writable VHD blobs, you can use the blobs to recreate your VM by using the template for the VM.

Other options

SQL Server

SQL Server running in a VM has its own built-in capabilities to back up your SQL Server database to Azure Blob storage or a file share. If the storage account is geo-redundant storage or read-access geo-redundant storage, you can access those backups in the storage account's secondary datacenter in the event of a disaster, with the same restrictions as previously discussed. For more information, see Back up and restore for SQL Server in Azure virtual machines. In addition to back up and restore, SQL Server Always On availability groups can maintain secondary replicas of databases. This ability greatly reduces the disaster recovery time.

Other considerations

This article has discussed how to back up or take snapshots of your VMs and their disks to support disaster recovery and how to use those backups or snapshots to recover your data. With the Azure Resource Manager model, many people use templates to create their VMs and other infrastructures in Azure. You can use a template to create a VM that has the same configuration every time. If you use custom images for creating your VMs, you must also make sure that your images are protected by using a read-access geo-redundant storage account to store them.

Consequently, your backup process can be a combination of two things:

- Back up the data (disks).
- Back up the configuration (templates and custom images).

Depending on the backup option you choose, you might have to handle the backup of both the data and the configuration, or the backup service might handle all of that for you.

Appendix: Understanding the impact of data redundancy

For storage accounts in Azure, there are three types of data redundancy that you should consider regarding disaster recovery: locally redundant, geo-redundant, or geo-redundant with read access.

Locally redundant storage retains three copies of the data in the same datacenter. When the VM writes the data, all three copies are updated before success is returned to the caller, so you know they are identical. Your disk is protected from local failures, because it's unlikely that all three copies are affected at the same time. In the case of locally redundant storage, there is no geo-redundancy, so the disk is not protected from catastrophic failures that can affect an entire datacenter or storage unit.

With geo-redundant storage and read-access geo-redundant storage, three copies of your data are retained in the primary region that is selected by you. Three more copies of your data are retained in a corresponding secondary region that is set by Azure. For example, if you store data in West US, the data is replicated to East US. Copy retention is done asynchronously, and there is a small delay between updates to the primary and secondary sites. Replicas of the disks on the secondary site are consistent on a per-disk basis (with the delay), but replicas of multiple active disks might not be in sync with each other. To have consistent replicas across multiple disks, consistent snapshots are needed.

The main difference between geo-redundant storage and read-access geo-redundant storage is that with read-access geo-redundant storage, you can read the secondary copy at any time. If there is a problem that renders the data in the primary region inaccessible, the Azure team makes every effort to restore access. While the primary is down, if you have read-access geo-redundant storage enabled, you can access the data in the secondary datacenter. Therefore, if you plan to read from the replica while the primary is inaccessible, then read-access geo-redundant storage should be considered.

If it turns out to be a significant outage, the Azure team might trigger a geo-failover and change the primary DNS entries to point to secondary storage. At this point, if you have either geo-redundant storage or read-access geo-redundant storage enabled, you can access the data in the region that used to be the secondary. In other words, if your storage account is geo-redundant storage and there is a problem, you can access the secondary storage only if there is a geo-failover.

For more information, see What to do if an Azure Storage outage occurs.

Next steps

See Back up Azure unmanaged Virtual Machine disks with incremental snapshots.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Ephemeral OS disks for Azure VMs

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

> **Note**
>
> NVMe disk placement for Ephemeral OS disks is now Generally Available (GA). Customers can use Ephemeral OS disks for production workloads on supported v6 VM series.

Ephemeral OS disks are created on the local virtual machine (VM) storage and not saved to the remote Azure Storage. Ephemeral OS disks are ideal for stateless workloads, where applications can tolerate individual VM failures but are sensitive to VM deployment times or the reimaging of individual VM instances. With Ephemeral OS disk, you get lower read/write latency to the OS disk and faster VM reimage.

The key features of ephemeral disks are:

- Ephemeral OS disks are supported on VM sizes with local SSD storage, including Premium SSD, NVMe, and temp disks. The performance and reliability of the ephemeral OS disk are directly tied to the underlying local SSD storage of the VM. For best results, select VM sizes that offer Premium SSD or NVMe-based local storage.
- Designed for stateless applications.
- Supported on all images - Marketplace, custom images, and Azure Compute Gallery (formerly known as Shared Image Gallery).
- Provides fast reimage to reset virtual machines (VMs) and scale set instances to their original boot state.
- Offers lower latency, similar to a temporary disk.
- Supports Premium SSD & Standard SSD for higher SLA
- Supported in all Azure regions.

Key differences between persistent and ephemeral OS disks:

Expand table

	Persistent OS Disk	Ephemeral OS Disk
Size limit for OS disk	4* TiB	Cache, temp, or NVMe disk size for the VM size or 2,040 GiB, whichever is smaller. For the **cache, temp or NVMe size in GiB**, see DSv3, Esv3, M, FS, and GS
VM sizes supported	All	VM sizes with local storage such as D(a)dsv4, D(a)dsv5, D(a)dsv6, FXmdsv2, E(a)dsv5,

	Persistent OS Disk	Ephemeral OS Disk
		L(a)sv3, etc.
Disk type support	Managed and unmanaged OS disk	Managed OS disk only
Region support	All regions	All regions
Data persistence	OS disk data written to OS disk are stored in Azure Storage	Data written to OS disk is stored on local VM storage and isn't persisted to Azure Storage.
Stop-deallocated state	VMs and scale set instances can be stop-deallocated and restarted from the stop-deallocated state	Not Supported
Specialized OS disk support	Yes	No
OS disk resize	Supported during VM creation and after VM is stop-deallocated	Supported during VM creation only
Resizing to a new VM size	OS disk data is preserved	Data on the OS disk is deleted, OS is reprovisioned
Redeploy	OS disk data is preserved	Data on the OS disk is deleted, OS is reprovisioned
Stop/ Start of VM	OS disk data is preserved	Not Supported
Page file placement	For Windows, page file is stored on the temp disk	For Windows, page file is stored on the OS disk (for cache placement, Temp disk placement & NVMe disk placement).
Maintenance of VM/VMSS using healing	OS disk data is preserved	OS disk data isn't preserved
Maintenance of VM/VMSS using Live Migration	OS disk data is preserved	OS disk data is preserved

* 4 TiB is the maximum supported OS disk size for managed (persistent) disks. However, many OS disks are partitioned with master boot record (MBR) by default and are limited to 2 TiB. For details, see OS disk.

Placement options for Ephemeral OS disks

Ephemeral OS Disk utilizes local storage within the VM. Since different VMs have different types of local storage (cache disk, temp disk, and NVMe disk), the placement option defines where the Ephemeral OS Disk is stored. Placement option however doesn't impact the performance or cost of Ephemeral OS disk. Its performance is dependent upon the VM's local storage. Depending upon the VM type, we offer three different types of placement:

- **NVMe Disk Placement (Generally Available)** - NVMe disk placement type is now generally available (GA) on the latest generation v6 VM series onwards like Dadsv6, Ddsv6, Dpdsv6, etc.
- **Temp Disk Placement (also known as Resource Disk Placement)** - Temp disk placement type is available on VMs with Temp disk like Dadsv5, Ddsv5, etc.
- **Cache Disk Placement** - Cache disk placement type is available on older VMs that had cache disk like Dsv2, Dsv3, etc.

DiffDiskPlacement is the property that can be used to specify where you want to place the Ephemeral OS disk. By default, Azure will pick up the right placement type, depending on the VM SKU. The customers are recommended to use the latest VM series (v5/v6) with either Temp Disk or NVMe Disk placement.

Size requirements

You can choose to deploy Ephemeral OS Disk on NVMe disk, temp disk, or cache on the VM. The image OS disk's size should be less than or equal to the NVMe/temp/cache size of the VM size chosen.

For **OS cache placement**: Standard Windows Server images from the marketplace are about 127 GiB, which means that you need a VM size that has a cache equal to or larger than 127 GiB. The Standard_DS3_v2 has a cache size of 127 GiB, which is large enough. In this case, the Standard_DS3_v2 is the smallest size in the DSv2 series that you can use with this image.

For **Temp disk placement**: Standard Ubuntu server image from marketplace is about 30 GiB. To enable Ephemeral OS disk on temp, the temp disk size must be equal to or larger than 30 GiB. Standard_B4ms has a temp size of 32 GiB, which can fit the 30-GiB OS disk. Upon creation of the VM, the temp disk space would be 2 GiB.

For **NVMe disk placement (GA)**: Standard Ubuntu server image from marketplace is about 30 GiB. To enable Ephemeral OS disk on NVMe, the NVMe disk size must be equal to or larger than 30 GiB. Standard_D2ads_v6 has a Nvme disk size of 110 GiB, which can easily fit the 30-GiB OS disk. However, Ephemeral OS disk occupies the entire NVMe disk and there's no NVMe disk space given back. One way to maximize the use of NVMe disk is by maximizing the OS disk Size property to 110 GiB.

> **ⓘ Important**
>
> If opting for temp disk placement the Final Temp disk size = (Initial temp disk size - OS image size).
>
> If opting for NVMe disk placement (GA), Final NVMe Disk size = (Total no. of NVMe disks - NVMe Disks used for OS) * Size of each NVMe disk. Where NVMe Disks used for OS is the minimum number of disks required for OS disk depending on the size of OS disk and the size of each NVMe disk.

If Ephemeral OS disk is using **Temp Disk Placement**, it shares the IOPS(input/output operations per second) with temp disk. If Ephemeral OS disk is using **NVMe Disk Placement**, it provides the IOPS(input/output operations per second) of NVMe disks being used.

Basic Linux and Windows Server images in the Marketplace that are denoted with `[smallsize]` tend to be around 30 GiB and can use most of the available VM sizes.

> **ⓘ Note**
>
> Ephemeral disks aren't accessible through the portal. You receive a "Resource not Found" or "404" error when accessing the ephemeral disk which is expected.

Unsupported features

- VM Image Capture
- Disk snapshots
- Azure Disk Encryption
- Azure Backup
- Azure Site Recovery
- OS Disk Swaps

Trusted Launch for Ephemeral OS disks

Ephemeral OS disks can be created with Trusted launch. All regions are supported for Trusted Launch; not all virtual machines sizes are supported. Check Virtual machines sizes supported for supported sizes. VM guest state (VMGS) is specific to trusted launch VMs. It's an Azure-managed blob and contains the unified extensible firmware interface (UEFI) secure boot signature databases and other security information. VMs using trusted launch by default

reserve **1 GiB** from the **OS cache** or **temp disk** or **Nvme Disk** based on the chosen placement option for VMGS. The lifecycle of the VMGS blob is tied to that of the OS Disk.

For example, If you try to create a Trusted launch Ephemeral OS disk VM using OS image of size 75 GiB with VM size Standard_D2ads_v5 using temp disk placement you would get an error as **"OS disk of Ephemeral VM with size greater than 74 GB is not allowed for VM size Standard_Dads_v5 when the DiffDiskPlacement is ResourceDisk."** This error occurs because the temp disk for Standard_D2ads_v5 is 75 GiB, and 1 GiB is reserved for VMGS when using trusted launch. For the same example, if you create a standard Ephemeral OS disk VM you wouldn't get any errors and it would be a successful operation.

> **Important**
>
> If you use ephemeral disks with Trusted Launch VMs, any keys or secrets that the vTPM generates or seals after the VM is created might not be saved. As a result, these keys and secrets could be lost during actions such as reimaging or service healing events.

For more information on how to deploy a trusted launch VM

Confidential VMs using Ephemeral OS disks

AMD-based Confidential VMs cater to high security and confidentiality requirements of customers. These VMs provide a strong, hardware-enforced boundary to help meet your security needs. There are limitations to use Confidential VMs. Check the region, size, and OS supported limitations for confidential VMs. Virtual machine guest state (VMGS) blob contains the security information of the confidential VM. Confidential VMs using Ephemeral OS disks by default **1 GiB** from the **OS cache** or **temp storage** based on the chosen placement option is reserved for VMGS. The lifecycle of the VMGS blob is tied to that of the OS Disk.

> **Important**
>
> When choosing a confidential VM with full OS disk encryption before VM deployment that uses a customer-managed key (CMK). **Updating a CMK key version** or **key rotation** isn't supported with Ephemeral OS disk. Confidential VMs using Ephemeral OS disks need to be deleted before updating or rotating the keys and can be re-created later.

For more information on confidential VM

Customer Managed key

You can choose to use customer managed keys or platform managed keys when you enable end-to-end encryption for VMs using Ephemeral OS disk. Currently this option is available only via PowerShell, CLI, and SDK in all regions.

> **Important**
>
> Updating a CMK key version or key rotation of customer managed key isn't supported with Ephemeral OS disk. VMs using Ephemeral OS disks need to be deleted before updating or rotating the keys and can be re-created later.

For more information on Encryption at host

SSD storage account support for Ephemeral OS disks

SSD support is a new option that allows customers to choose the type of base disk that is used for the ephemeral OS disk. Previously, the base disk could only be Standard HDD. Now, customers can choose between the three types of disks: Standard HDD(Standard_LRS), Standard SSD (StandardSSD_LRS) or Premium SSD (Premium_LRS). By utilizing SSD with Ephemeral OS disk, customers can benefit from the following enhancements:

- **Enhanced SLA**: VMs created with Premium SSD provide higher SLA than VMs created with Standard HDD. Customers can enhance SLA for their Ephemeral VMs by choosing Premium SSD as base disk.
- **Improved performance**: By choosing Premium SSD as the base disk, customers can enhance the disk read performance of their VMs. While most writes occur on the local temp disk, some reads are performed from managed disks. Premium SSD disks provide 8-10 times higher IOPS than Standard HDD.

Local temporary storage

Some Azure VM sizes include local temporary storage, with some of the newer sizes using Temporary local NVMe disks. Local temporary disks are different from Ephemeral OS disks.

Local temporary storage, also known as local ephemeral storage, are additional disks provisioned directly as local storage to an Azure virtual machine host, rather than on remote Azure Storage. This type of storage is best suited for data that does not need to be retained permanently, such as caches, buffers, and temporary files. Local ephemeral storage is not backed up and is lost when the VM is deallocated or deleted. The ephemeral storage is recreated on startup.

Azure VM sizes featuring a 'd' in their naming convention - such as the Da***d***sv6, Ea***d***sv6, and FXm***d***sv2 series - include dedicated local temporary disks.

Next steps

Create a VM with ephemeral OS disk using Azure Portal/CLI/PowerShell/ARM template. Check out the frequently asked questions on ephemeral os disk.

Last updated on 12/01/2025

How to deploy Ephemeral OS disks for Azure VMs

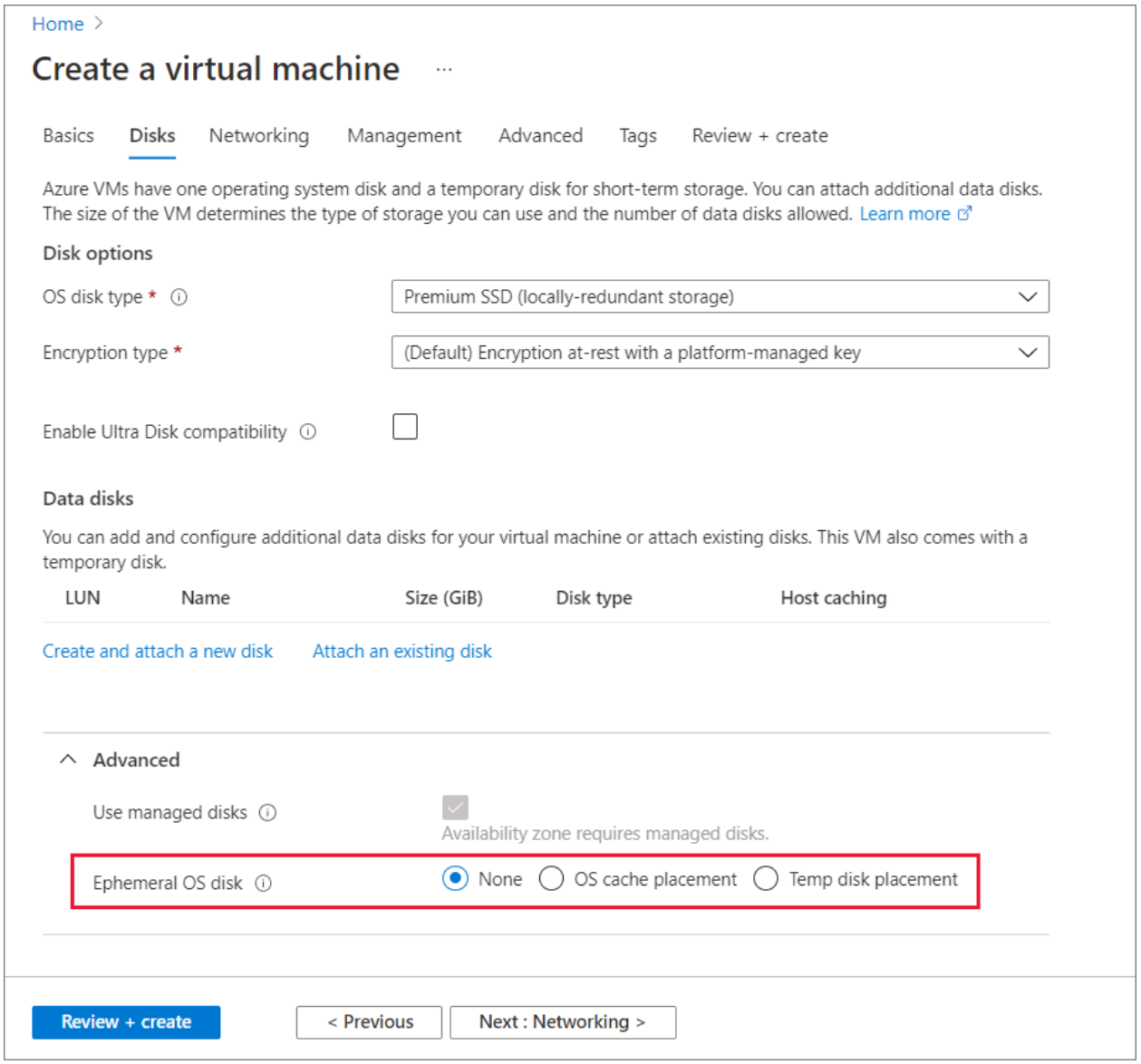
Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article shows you how to create a virtual machine or virtual machine scale sets with Ephemeral OS disks through Portal, ARM template deployment, CLI and PowerShell.

Portal

In the Azure portal, you can choose to use ephemeral disks when deploying a virtual machine or virtual machine scale sets by opening the **Advanced** section of the **Disks** tab. For choosing placement of Ephemeral OS disk, select **OS cache placement** or **Temp disk placement**.



If the option for using an ephemeral disk or OS cache placement or Temp disk placement is greyed out, you might have selected a VM size that doesn't have a cache/temp size larger than the OS image or that doesn't support Premium storage. Go back to the **Basics** page and try choosing another VM size.

Scale set template deployment

The process to create a scale set that uses an ephemeral OS disk is to add the `diffDiskSettings` property to the `Microsoft.Compute/virtualMachineScaleSets/virtualMachineProfile` resource type in the template. Also, the caching policy must be set to `ReadOnly` for the ephemeral OS disk. placement can be changed to `CacheDisk` for OS cache disk placement.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachineScaleSets",
```

```
  "name": "myScaleSet",
  "location": "East US 2",
  "apiVersion": "2019-12-01",
  "sku": {
    "name": "Standard_DS2_v2",
    "capacity": "2"
  },
  "properties": {
    "upgradePolicy": {
      "mode": "Automatic"
    },
    "virtualMachineProfile": {
       "storageProfile": {
        "osDisk": {
          "diffDiskSettings": {
            "option": "Local" ,
            "placement": "ResourceDisk"
          },
          "caching": "ReadOnly",
          "createOption": "FromImage"
        },
        "imageReference":  {
          "publisher": "publisherName",
          "offer": "offerName",
          "sku": "skuName",
          "version": "imageVersion"
        }
      },
      "osProfile": {
        "computerNamePrefix": "myvmss",
        "adminUsername": "azureuser",
        "adminPassword": "P@ssw0rd!"
      }
    }
  }
}
```

> ⓘ **Note**
>
> Replace all the other values accordingly.

VM template deployment

You can deploy a VM with an ephemeral OS disk using a template. The process to create a VM that uses ephemeral OS disks is to add the `diffDiskSettings` property to Microsoft.Compute/virtualMachines resource type in the template. Also, the caching policy must be set to `ReadOnly` for the ephemeral OS disk. placement option can be changed to `CacheDisk` for OS cache disk placement.

JSON

```
{
  "type": "Microsoft.Compute/virtualMachines",
  "name": "myVirtualMachine",
  "location": "East US 2",
  "apiVersion": "2019-12-01",
  "properties": {
       "storageProfile": {
            "osDisk": {
              "diffDiskSettings": {
                "option": "Local" ,
                "placement": "ResourceDisk"
              },
              "caching": "ReadOnly",
              "createOption": "FromImage"
            },
            "imageReference": {
                "publisher": "MicrosoftWindowsServer",
                "offer": "WindowsServer",
                "sku": "2016-Datacenter-smalldisk",
                "version": "latest"
            },
            "hardwareProfile": {
                 "vmSize": "Standard_DS2_v2"
             }
      },
      "osProfile": {
        "computerNamePrefix": "myvirtualmachine",
        "adminUsername": "azureuser",
        "adminPassword": "P@ssw0rd!"
      }
    }
 }
```

CLI

To use an ephemeral disk for a CLI VM deployment, set the `--ephemeral-os-disk` parameter in az vm create to `true` and the `--ephemeral-os-disk-placement` parameter to `ResourceDisk` for temp disk placement or `CacheDisk` for cache disk placement and the `--os-disk-caching` parameter to `ReadOnly`.

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --image imageName \
  --ephemeral-os-disk true \
  --ephemeral-os-disk-placement ResourceDisk \
```

```
  --os-disk-caching ReadOnly \
  --admin-username azureuser \
  --generate-ssh-keys
```

> **Note**
>
> Replace `myVM`, `myResourceGroup`, `imageName` and `azureuser` accordingly.

For scale sets, you use the same `--ephemeral-os-disk true` parameter for az-vmss-create and set the `--os-disk-caching` parameter to `ReadOnly` and the `--ephemeral-os-disk-placement` parameter to `ResourceDisk` for temp disk placement or `CacheDisk` for cache disk placement.

Reimage a VM using REST

You can reimage a Virtual Machine instance with ephemeral OS disk using REST API as described below and via Azure portal by going to Overview pane of the VM. For scale sets, reimaging is already available through PowerShell, CLI, and the portal.

```
POST https://management.azure.com/subscriptions/{sub-id}/resourceGroups/{rgName}/providers/Microsoft.Compute/VirtualMachines/{vmName}/reimage?api-version=2019-12-01"
```

PowerShell

To use an ephemeral disk for a PowerShell VM deployment, use Set-AzVMOSDisk in your VM configuration. Set the `-DiffDiskSetting` to `Local` and `-Caching` to `ReadOnly` and `-DiffDiskPlacement` to `ResourceDisk`.

PowerShell

```
Set-AzVMOSDisk -DiffDiskSetting Local -DiffDiskPlacement ResourceDisk -Caching ReadOnly
```

To use an ephemeral disk on cache disk for a PowerShell VM deployment, use Set-AzVMOSDisk in your VM configuration. Set the `-DiffDiskSetting` to `Local` , `-Caching` to `ReadOnly` and `-DiffDiskPlacement` to `CacheDisk`.

PowerShell

```
Set-AzVMOSDisk -DiffDiskSetting Local -DiffDiskPlacement CacheDisk -Caching
ReadOnly
```

For scale set deployments, use the Set-AzVmssStorageProfile cmdlet in your configuration. Set the `-DiffDiskSetting` to `Local` , `-Caching` to `ReadOnly` and `-DiffDiskPlacement` to `ResourceDisk` or `CacheDisk`.

PowerShell

```
Set-AzVmssStorageProfile -DiffDiskSetting Local -DiffDiskPlacement
ResourceDisk -OsDiskCaching ReadOnly
```

Next steps

For more information on Ephemeral OS disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Frequently asked questions about Ephemeral OS disks

Article • 10/18/2024

Q: What is the size of the Ephemeral OS Disks?

A: We support platform, Shared Image Gallery, and custom images. Size of Ephemeral OS disk depends on the local storage (cache disk, temp disk, or NVMe disk) available on the Virtual Machine (VM). Max size of Ephemeral OS disk can be 2 TiB. All writes to the OS disk are local on the same node as the Virtual Machine.

Q: Can the ephemeral OS disk be resized?

A: No, once the ephemeral OS disk is provisioned, the OS disk can't be resized.

Q: Can the ephemeral OS disk placement be modified after creation of VM?

A: No, once the ephemeral OS disk is provisioned, the OS disk placement can't be changed. But the VM can be recreated via ARM template deployment/PowerShell/CLI by updating the OS disk placement of choosing. This results in the recreation of the VM with Data on the OS disk deleted and OS is reprovisioned.

Q: Is there any Temp disk created if image size equals to Temp disk size of VM size selected?

A: No, in that case, Temp disk is not created.

Q: How is NVMe disk placement different from temp disk placement?

A: Nvme disk placement utilizes the entire Nvme disk, while temp disk placement utilizes space equal to the OS disk size. For example, if you're using Standard Ubuntu image on Standard_D2ads_v5 with temp disk placement, you get an OS disk size of 30 GiB & temp disk size of 45 GiB. If you're using the same Standard Ubuntu image on Standard_D2ads_v6 with NVMe disk placement, then you get and OS disk size of 30 GiB with no NVMe disk.

Q: How does NVMe disk placement work for VMs with multiple NVMe disks?

A: NVMe Disk Placement utilizes the minimum number of disks required for the OS disk. For example, Standard_D16ads_v6 has two disks of 440 GiB each. If you create and Ephemeral OS disk of size less than 440 GiB, then only one disk is used for creating the ephemeral OS disk. The second NVMe disk is returned to you as a raw local disk.

Q: Are Ephemeral OS disks supported on low-priority VMs and Spot VMs?

A: Yes. There's no option of Stop-Deallocate for Ephemeral VMs, rather users need to Delete instead of deallocating them.

Q: Can I attach a Managed Disks to an Ephemeral VM?

A: Yes, you can attach a managed data disk to a VM that uses an ephemeral OS disk.

Q: Will all VM sizes be supported for ephemeral OS disks?

A: No, most Premium Storage VM sizes are supported (DS, ES, FS, GS, M, etc.). To know whether a particular VM size supports ephemeral OS disks for an OS image size, you can use the following script. It takes the OS image size and location as inputs and provides a list of VMs and corresponding placement supported. If cache, temp, and NVMe disk placement are marked as not supported then Ephemeral OS disk can't be used for the given OS image size.

Azure PowerShell

```
[CmdletBinding()]
param([Parameter(Mandatory=$true)]
      [ValidateNotNullOrEmpty()]
      [string]$Location,
      [Parameter(Mandatory=$true)]
      [long]$OSImageSizeInGB
      )

Function HasSupportEphemeralOSDisk([object[]] $capability)
{
    return $capability | where { $_.Name -eq "EphemeralOSDiskSupported" -and
$_.Value -eq "True"}
}

Function Get-MaxTempDiskAndCacheSize([object[]] $capabilities)
{
    $MaxResourceVolumeGB = 0;
    $CachedDiskGB = 0;
    $NvmeDiskGB = 0;

    foreach($capability in $capabilities)
    {
        if ($capability.Name -eq "MaxResourceVolumeMB")
        { $MaxResourceVolumeGB = [int]($capability.Value / 1024) }

        if ($capability.Name -eq "CachedDiskBytes")
        { $CachedDiskGB = [int]($capability.Value / (1024 * 1024 * 1024)) }

         if ($capability.Name -eq "NvmeDiskSizeInMiB")
        { $NvmeDiskGB = [int]($capability.Value / (1024)) }
```

```
        if ($capability.Name -eq "SupportedEphemeralOSDiskPlacements")
        { $NvmeSupported = [bool]($capability.Value -contains "NvmeDisk") }

    }

    if (!$NvmeSupported)
    { $NvmeDiskGB = 0; }
    return ($MaxResourceVolumeGB, $CachedDiskGB, $NvmeDiskGB)
}

Function Get-EphemeralSupportedVMSku
{
    [CmdletBinding()]
    Param
    (
        [Parameter(Mandatory=$true)]
        [long]$OSImageSizeInGB,
        [Parameter(Mandatory=$true)]
        [string]$Location
    )

    $VmSkus = Get-AzComputeResourceSku $Location | Where-Object {
$_.ResourceType -eq "virtualMachines" -and (HasSupportEphemeralOSDisk
$_.Capabilities) -ne $null }

    $Response = @()
    foreach ($sku in $VmSkus)
    {
        ($MaxResourceVolumeGB, $CachedDiskGB, $NvmeDiskGB) = Get-
MaxTempDiskAndCacheSize $sku.Capabilities

        $Response += New-Object PSObject -Property @{
            ResourceSKU = $sku.Size
            NvmeDiskPlacement = @{ $true = "NOT SUPPORTED"; $false =
"SUPPORTED"}[$NvmeDiskGB -lt $OSImageSizeInGB]
            TempDiskPlacement = @{ $true = "NOT SUPPORTED"; $false =
"SUPPORTED"}[$MaxResourceVolumeGB -lt $OSImageSizeInGB]
            CacheDiskPlacement = @{ $true = "NOT SUPPORTED"; $false =
"SUPPORTED"}[$CachedDiskGB -lt $OSImageSizeInGB]
         };
    }

    return $Response
}

Get-EphemeralSupportedVMSku -OSImageSizeInGB $OSImageSizeInGB -Location
$Location | Format-Table
```

Q: Can the ephemeral OS disk be applied to existing VMs and scale sets?

A: No, ephemeral OS disk can only be used during VM and scale set creation.

Q: Can you mix ephemeral and normal OS disks in a scale set?

A: No, you can't have a mix of ephemeral and persistent OS disk instances within the same scale set.

Q: Can the ephemeral OS disk be created using PowerShell or CLI?

A: Yes, you can create VMs with Ephemeral OS Disk using REST, Templates, PowerShell, and CLI.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Restrict managed disks from being imported or exported

Summarize this article for me

This article provides an overview of your options for preventing your Azure Managed Disks from being imported or exported.

Custom role

To limit the number of people who can import or export managed disks or snapshots using Azure RBAC, create a custom RBAC role that doesn't have the following permissions:

- Microsoft.Compute/disks/beginGetAccess/action
- Microsoft.Compute/disks/endGetAccess/action
- Microsoft.Compute/snapshots/beginGetAccess/action
- Microsoft.Compute/snapshots/endGetAccess/action

Any custom role without those permissions can't upload or download managed disks.

Microsoft Entra authentication

If you're using Microsoft Entra ID to control resource access, you can also use it to restrict uploading of Azure Managed Disks. When a user attempts to upload a disk, Azure validates the identity of the requesting user in Microsoft Entra ID, and confirms that user has the required permissions. To learn more, see Secure downloads and uploads of Azure Managed Disks.

Private links

You can use private endpoints to restrict the upload and download of managed disks and more securely access data over a private link from clients on your Azure virtual network. The private endpoint uses an IP address from the virtual network address space for your managed disks. Network traffic between clients on their virtual network and managed disks only traverses over the virtual network and a private link on the Microsoft backbone network, eliminating exposure from the public internet. Use a disk access resource to facilitate private link connections to your disks. For details, see either the portal or CLI articles.

Azure policy

Configure an Azure Policy to disable public network access to your managed disks.

Configure the Network access policy

Each managed disk and snapshot has its own NetworkAccessPolicy parameter that can prevent the resource from being exported. You can use the Azure CLI or Azure PowerShell module to set the parameter to **DenyAll**, which prevents the resource from being exported.

Last updated on 02/19/2026

Secure downloads and uploads of Azure managed disks

If you use Microsoft Entra ID to control resource access, you can also use it to restrict uploads and downloads of Azure managed disks. When a user tries to upload or download a disk, Azure validates the identity of the requesting user in Microsoft Entra ID, and confirms that user has the required permissions. If a user doesn't have the required permissions, they can't upload or download managed disks.

At a higher level, a system administrator can set a policy at the Azure account or subscription level, to ensure that all disks and snapshots must use Microsoft Entra ID for uploads or downloads. If you have any questions on securing uploads or downloads by using Microsoft Entra ID, reach out to: azuredisks@microsoft .com.

Restrictions

- VHDs can't be uploaded to empty snapshots.
- Azure Backup doesn't currently support disks secured with Microsoft Entra ID.
- Azure Site Recovery doesn't currently support disks secured with Microsoft Entra ID.

Prerequisites

Install the latest Azure PowerShell module or the latest Azure CLI.

Assign RBAC role

To access managed disks secured by using Microsoft Entra ID, users must have either the Data Operator for Managed Disks role or a custom role with the following permissions:

- **Microsoft.Compute/disks/download/action**
- **Microsoft.Compute/disks/upload/action**
- **Microsoft.Compute/snapshots/download/action**
- **Microsoft.Compute/snapshots/upload/action**

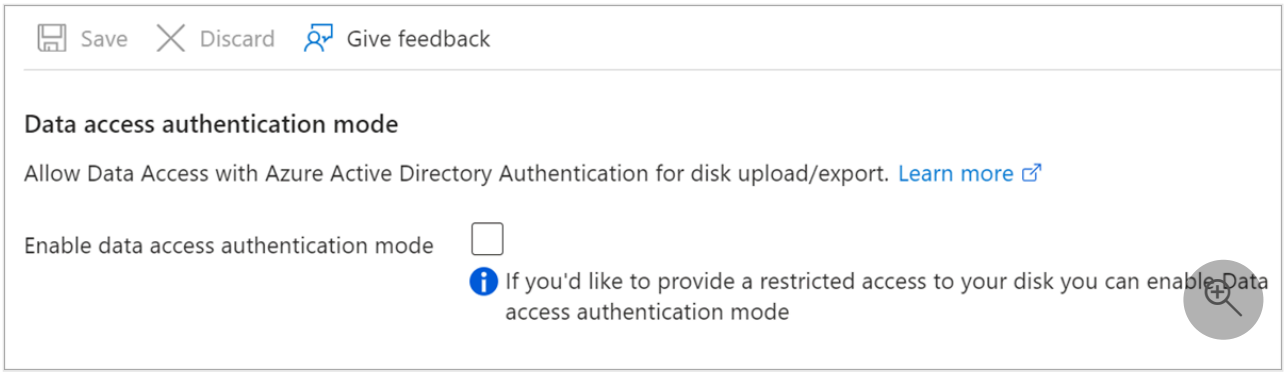
For detailed steps on assigning a role, see the following articles for portal, PowerShell, or CLI. To create or update a custom role, see the following articles for portal, PowerShell, or CLI.

Restrict access to an individual disk

To restrict access to an individual disk, enable **data access authentication mode** on that disk.

Portal

You can enable this setting when creating the disk, or you can enable it on the **Disk Export** page under **Settings** for existing disks.



Assign Azure policy

You can also assign an Azure policy with a remediation task. A policy with a remediation task continuously audits your resources and notifies you when any of them don't comply. The built-in policy definition you'd assign is **Protect your data with authentication requirements when exporting or uploading to a disk or snapshot**. To learn how to assign an Azure policy see the Azure portal, Azure CLI, or Azure PowerShell module articles.

Next steps

- Upload a VHD to Azure or copy a managed disk to another region - Azure CLI
- Upload a VHD to Azure or copy a managed disk to another region - Azure PowerShell
- Download a Linux VHD from Azure
- Download a Windows VHD from Azure

Last updated on 02/19/2026

Azure CLI - Restrict import/export access for managed disks with Private Links

Applies to: ✔ Linux VMs ✔ Flexible scale sets

You can use private endpoints to restrict the export and import of managed disks and securely access data over a Private Link from clients on your Azure virtual network. The private endpoint uses an IP address from the virtual network address space for your managed disks service. Network traffic between clients on their virtual network and managed disks only traverses over the virtual network and a private link on the Microsoft backbone network, eliminating exposure from the public internet.

To use Private Links to export/import managed disks, first you create a disk access resource and link it to a virtual network in the same subscription by creating a private endpoint. Then, associate a disk or a snapshot with an instance of disk access. Finally, set the NetworkAccessPolicy property of the disk or the snapshot to `AllowPrivate`. This will limit access to your virtual network.

You can set the NetworkAccessPolicy property to `DenyAll` to prevent anybody from exporting data of a disk or a snapshot. The default value for the NetworkAccessPolicy property is `AllowAll`.

Limitations

- You can't import or export more than 100 disks or snapshots at the same time with the same disk access resource
- You can't upload to a disk with both a disk access resource and a disk encryption set
- In addition to the scale targets that apply to individual disks, disk access resources have more scale targets that are centered around ingress/egress of data. These limits apply cumulatively to all disks associated with a disk access resource. For details see here

Log in into your subscription and set your variables

Azure CLI

```
subscriptionId=yourSubscriptionId
resourceGroupName=yourResourceGroupName
region=northcentralus
diskAccessName=yourDiskAccessForPrivateLinks
vnetName=yourVNETForPrivateLinks
subnetName=yourSubnetForPrivateLinks
privateEndPointName=yourPrivateLinkForSecureMDExportImport
```

```
privateEndPointConnectionName=yourPrivateLinkConnection

#The name of an existing disk which is the source of the snapshot
sourceDiskName=yourSourceDiskForSnapshot

#The name of the new snapshot which will be secured via Private Links
snapshotNameSecuredWithPL=yourSnapshotNameSecuredWithPL

az login

az account set --subscription $subscriptionId
```

Create a disk access using Azure CLI

Azure CLI

```
az disk-access create -n $diskAccessName -g $resourceGroupName -l $region

diskAccessId=$(az disk-access show -n $diskAccessName -g $resourceGroupName --query
[id] -o tsv)
```

Create a Virtual Network

Network policies like network security groups (NSG) are not supported for private endpoints. In order to deploy a Private Endpoint on a given subnet, an explicit disable setting is required on that subnet.

Azure CLI

```
az network vnet create --resource-group $resourceGroupName \
    --name $vnetName \
    --subnet-name $subnetName
```

Disable subnet private endpoint policies

Azure deploys resources to a subnet within a virtual network, so you need to update the subnet to disable private endpoint network policies.

Azure CLI

```
az network vnet subnet update --resource-group $resourceGroupName \
    --name $subnetName  \
    --vnet-name $vnetName \
    --disable-private-endpoint-network-policies true
```

Create a private endpoint for the disk access object

Azure CLI

```
az network private-endpoint create --resource-group $resourceGroupName \
    --name $privateEndPointName \
    --vnet-name $vnetName  \
    --subnet $subnetName \
    --private-connection-resource-id $diskAccessId \
    --group-ids disks \
    --connection-name $privateEndPointConnectionName
```

Configure the Private DNS Zone

Create a Private DNS Zone for Storage blob domain, create an association link with the Virtual Network and create a DNS Zone Group to associate the private endpoint with the Private DNS Zone.

Azure CLI

```
az network private-dns zone create --resource-group $resourceGroupName \
    --name "privatelink.blob.core.windows.net"

az network private-dns link vnet create --resource-group $resourceGroupName \
    --zone-name "privatelink.blob.core.windows.net" \
    --name yourDNSLink \
    --virtual-network $vnetName \
    --registration-enabled false

az network private-endpoint dns-zone-group create \
   --resource-group $resourceGroupName \
   --endpoint-name $privateEndPointName \
   --name yourZoneGroup \
   --private-dns-zone "privatelink.blob.core.windows.net" \
   --zone-name disks
```

Create a disk protected with Private Links

Azure CLI

```
resourceGroupName=yourResourceGroupName
region=northcentralus
diskAccessName=yourDiskAccessName
diskName=yourDiskName
diskSkuName=Standard_LRS
diskSizeGB=128
```

```
diskAccessId=$(az resource show -n $diskAccessName -g $resourceGroupName --namespace
Microsoft.Compute --resource-type diskAccesses --query [id] -o tsv)

az disk create -n $diskName \
-g $resourceGroupName \
-l $region \
--size-gb $diskSizeGB \
--sku $diskSkuName \
--network-access-policy AllowPrivate \
--disk-access $diskAccessId
```

Create a snapshot of a disk protected with Private Links

Azure CLI

```
resourceGroupName=yourResourceGroupName
region=northcentralus
diskAccessName=yourDiskAccessName
sourceDiskName=yourSourceDiskForSnapshot
snapshotNameSecuredWithPL=yourSnapshotName

diskId=$(az disk show -n $sourceDiskName -g $resourceGroupName --query [id] -o tsv)

diskAccessId=$(az resource show -n $diskAccessName -g $resourceGroupName --namespace
Microsoft.Compute --resource-type diskAccesses --query [id] -o tsv)

az snapshot create -n $snapshotNameSecuredWithPL \
-g $resourceGroupName \
-l $region \
--source $diskId \
--network-access-policy AllowPrivate \
--disk-access $diskAccessId
```

Next steps

- Upload a VHD to Azure or copy a managed disk to another region - Azure CLI or Azure PowerShell module
- Download a VHD - Windows or Linux
- FAQ on Private Links
- Export/Copy managed snapshots as VHD to a storage account in different region with CLI

Last updated on 11/05/2025

Upload a VHD to Azure or copy a managed disk to another region - Azure PowerShell

Summarize this article for me

Applies to: ✔ Windows VMs

This article explains how to either upload a VHD from your local machine to an Azure Managed Disk or copy a managed disk to another region, using the Azure PowerShell module. The process of uploading a managed disk, also known as direct upload, enables you to upload a VHD up to 32 TiB in size directly into a managed disk. Currently, direct upload is supported for Ultra Disks, Premium SSD v2, Premium SSD, Standard SSD, and Standard HDD.

If you're providing a backup solution for IaaS VMs in Azure, you should use direct upload to restore customer backups to managed disks. When uploading a VHD from a source external to Azure, speeds depend on your local bandwidth. When uploading or copying from an Azure VM, your bandwidth would be the same as Standard HDDs.

If you're using Microsoft Entra ID to control resource access, you can use it to restrict uploading of Azure Managed Disks. See Secure downloads and uploads of Azure Managed Disks for details.

Get started

There are two ways you can upload a VHD with the Azure PowerShell module: You can either use the Add-AzVHD command, which will automate most of the process for you, or you can perform the upload manually with AzCopy.

For Premium SSDs, Standard SSDs, and Standard HDDs, you should generally use Add-AzVHD. However, if you're uploading to an Ultra Disk, or a Premium SSD v2, or if you need to upload a VHD that is larger than 50 GiB, you must upload the VHD or VHDX manually with AzCopy. VHDs 50 GiB and larger upload faster using AzCopy and Add-AzVhd doesn't currently support uploading to an Ultra Disk or a Premium SSD v2.

For guidance on how to copy a managed disk from one region to another, see Copy a managed disk.

Use Add-AzVHD

Prerequisites

- Install the Azure PowerShell module.
- A VHD has been prepared for Azure, stored locally.
  - On Windows: You don't need to convert your VHD to VHDx, convert it a fixed size, or resize it to include the 512-byte offset. `Add-AZVHD` performs these functions for you.
    - Hyper-V must be enabled for Add-AzVHD to perform these functions.
  - On Linux: You must perform these actions manually. See Resizing VHDs for details.

Upload a VHD

(Optional) Grant access to the disk

If Microsoft Entra ID is used to enforce upload restrictions on a subscription or at the account level, Add-AzVHD only succeeds if attempted by a user that has the appropriate RBAC role or necessary permissions. You'll need to assign RBAC permissions to grant access to the disk and generate a writeable SAS.

Azure PowerShell

```
New-AzRoleAssignment -SignInName <emailOrUserprincipalname> `
-RoleDefinitionName "Data Operator for Managed Disks" `
-Scope /subscriptions/<subscriptionId>
```

Use Add-AzVHD

The following example uploads a VHD from your local machine to a new Azure Managed Disk using Add-AzVHD. Replace `<your-filepath-here>`, `<your-resource-group-name>`, `<desired-region>`, and `<desired-managed-disk-name>` with your parameters:

> **Note**
>
> If you're using Microsoft Entra ID to secure disk uploads, add `-DataAccessAuthMode 'AzureActiveDirectory'` to the end of your `Add-AzVhd` command.

Azure PowerShell

```
# Required parameters
$path = <your-filepath-here>.vhd
$resourceGroup = <your-resource-group-name>
$location = <desired-region>
$name = <desired-managed-disk-name>

# Optional parameters
```

```
# $Zone = <desired-zone>
# $sku=<desired-SKU>
# -DataAccessAuthMode 'AzureActiveDirectory'
# -DiskHyperVGeneration = V1 or V2. This applies only to OS disks.

# To use $Zone or #sku, add -Zone or -DiskSKU parameters to the command
Add-AzVhd -LocalFilePath $path -ResourceGroupName $resourceGroup -Location
$location -DiskName $name
```

Manual upload

Prerequisites

- Download the latest version of AzCopy v10.
- Install the Azure PowerShell module.
- A fixed size VHD that has been prepared for Azure, stored locally.

To upload your VHD to Azure, you'll need to create an empty managed disk that is configured for this upload process. Before you create one, there's some additional information you should know about these disks.

This kind of managed disk has two unique states:

- ReadyToUpload, which means the disk is ready to receive an upload but, no secure access signature (SAS) has been generated.
- ActiveUpload, which means that the disk is ready to receive an upload and the SAS has been generated.

> **Note**
>
> While in either of these states, the managed disk will be billed at Standard HDD pricing, regardless of the actual type of disk. For example, a P10 will be billed as an S10. This will be true until `revoke-access` is called on the managed disk, which is required in order to attach the disk to a VM.

Create an empty managed disk

Before you can create an empty Standard HDD for uploading, you'll need the file size of the VHD you want to upload, in bytes. The example code will get that for you but, to do it yourself you can use: `$vhdSizeBytes = (Get-Item "<fullFilePathHere>").length`. This value is used when specifying the **-UploadSizeInBytes** parameter.

Now, on your local shell, create an empty Standard HDD for uploading by specifying the **Upload** setting in the **-CreateOption** parameter as well as the **-UploadSizeInBytes** parameter in the New-AzDiskConfig cmdlet. Then call New-AzDisk to create the disk.

Replace `<yourdiskname>`, `<yourresourcegroupname>`, and `<yourregion>` then run the following commands:

> **Important**
>
> If you're creating an OS disk, add `-HyperVGeneration '<yourGeneration>'` to `New-AzDiskConfig`.
>
> If you're using Microsoft Entra ID to secure disk uploads, add `-dataAccessAuthMode 'AzureActiveDirectory'` to `New-AzDiskConfig`.
>
> When uploading to an Ultra Disk or Premium SSD v2 you need to select the correct sector size of the target disk. If you're using a VHD file with a 512 logical sector size, the target disk must be set to 512. If you're using a VHDX file with a 4k logical sector size, the target disk must be set to 4k, and target disk size must match VHDX file size. Note that uploading a VHDX file will result in a size expansion to the next 256 MiB alignment.
>
> VHDX files with logical sector size of 512 aren't supported.

PowerShell

```
$vhdSizeBytes = (Get-Item "<fullFilePathHere>").length

## For Ultra Disks or Premium SSD v2, add -LogicalSectorSize and specify either
4096 or 512, depending on if you're using a VHDX or a VHD

$diskconfig = New-AzDiskConfig -SkuName 'Standard_LRS' -OsType 'Windows' -
UploadSizeInBytes $vhdSizeBytes -Location '<yourregion>' -CreateOption 'Upload'

New-AzDisk -ResourceGroupName '<yourresourcegroupname>' -DiskName '<yourdiskname>'
-Disk $diskconfig
```

If you would like to upload a different disk type, replace **Standard_LRS** with **Premium_LRS**, **Premium_ZRS**, **StandardSSD_ZRS**, **StandardSSD_LRS**, or **UltraSSD_LRS**.

Generate writeable SAS

Now that you've created an empty managed disk that is configured for the upload process, you can upload a VHD to it. To upload a VHD to the disk, you'll need a writeable SAS, so that you can reference it as the destination for your upload.

> **Important**
>
> On February 15th, 2025, the Shared Access Signature (SAS) access time for disks and snapshots will be limited to a maximum of 60 days. Trying to generate a SAS with an expiration longer than 60 days results in an error. Any existing disk or snapshot SAS created with an expiration longer than 60 days may stop working 60 days after the date of creation and will result in a 403 error during authorization.

If a managed disk or snapshot SAS's expiration is longer than 60 days, revoke its access, and generate a new SAS that requests access for 60 days (5,184,000 seconds) or less. Improve your overall security by using SAS with shorter expiration dates. Make these changes before February 15, 2025 to prevent service interruption. The following links can be used to find, revoke, and request a new SAS.

- To check if a disk has an active SAS, you can either use the REST API, the Azure CLI, or the Azure PowerShell module, and examine the **DiskState** property.
- To revoke a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module.
- To create a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module, and set the access duration to 5,184,000 seconds or less.

To generate a writable SAS of your empty managed disk, replace `<yourdiskname>` and `<yourresourcegroupname>`, then use the following commands:

PowerShell

```
$diskSas = Grant-AzDiskAccess -ResourceGroupName '<yourresourcegroupname>' -
DiskName '<yourdiskname>' -DurationInSecond 86400 -Access 'Write'

$disk = Get-AzDisk -ResourceGroupName '<yourresourcegroupname>' -DiskName
'<yourdiskname>'
```

Upload a VHD or VHDX

Now that you have a SAS for your empty managed disk, you can use it to set your managed disk as the destination for your upload command.

Use AzCopy v10 to upload your local VHD or VHDX file to a managed disk by specifying the SAS URI you generated.

> **Note**

If you need to upload VHDx files larger than 2TB (which exceeds the VHD format limit) and can't convert them to VHD due to their size, please be aware that VHDx files are only supported for upload to PremiumSSDv2 and UltraSSD disk SKUs. For files smaller than 2TB, it's recommended to convert them to the VHD format before uploading.

This upload has the same throughput as the equivalent Standard HDD. For example, if you have a size that equates to S4, you'll have a throughput of up to 60 MiB/s. But, if you have a size that equates to S70, you'll have a throughput of up to 500 MiB/s.

```
AzCopy.exe copy "c:\somewhere\mydisk.vhd" $diskSas.AccessSAS --blob-type PageBlob
```

After the upload is complete, and you no longer need to write any more data to the disk, revoke the SAS. Revoking the SAS will change the state of the managed disk and allow you to attach the disk to a VM.

Replace `<yourdiskname>` and `<yourresourcegroupname>`, then run the following command:

PowerShell

```
Revoke-AzDiskAccess -ResourceGroupName '<yourresourcegroupname>' -DiskName
'<yourdiskname>'
```

Copy a managed disk

Direct upload also simplifies the process of copying a managed disk. You can either copy within the same region or copy your managed disk to another region.

The following script will do this for you, the process is similar to the steps described earlier, with some differences, since you're working with an existing disk.

> **ⓘ Important**
>
> You must add an offset of 512 when you're providing the disk size in bytes of a managed disk from Azure. This is because Azure omits the footer when returning the disk size. The copy will fail if you don't do this. The following script already does this for you.

Replace the `<sourceResourceGroupHere>`, `<sourceDiskNameHere>`, `<targetDiskNameHere>`, `<targetResourceGroupHere>`, `<yourOSTypeHere>` and `<yourTargetLocationHere>` (an example of a

location value would be uswest2) with your values, then run the following script in order to copy a managed disk.

> **Tip**
>
> If you're creating an OS disk, add `-HyperVGeneration '<yourGeneration>'` to `New-AzDiskConfig`.

PowerShell

```
$sourceRG = <sourceResourceGroupHere>
$sourceDiskName = <sourceDiskNameHere>
$targetDiskName = <targetDiskNameHere>
$targetRG = <targetResourceGroupHere>
$targetLocate = <yourTargetLocationHere>
$targetVmGeneration = "V1" # either V1 or V2
#Expected value for OS is either "Windows" or "Linux"
$targetOS = <yourOSTypeHere>

$sourceDisk = Get-AzDisk -ResourceGroupName $sourceRG -DiskName $sourceDiskName

# Adding the sizeInBytes with the 512 offset, and the -Upload flag
$targetDiskconfig = New-AzDiskConfig -SkuName 'Standard_LRS' -osType $targetOS -
UploadSizeInBytes $($sourceDisk.DiskSizeBytes+512) -Location $targetLocate -
CreateOption 'Upload' -HyperVGeneration $targetVmGeneration

$targetDisk = New-AzDisk -ResourceGroupName $targetRG -DiskName $targetDiskName -
Disk $targetDiskconfig

$sourceDiskSas = Grant-AzDiskAccess -ResourceGroupName $sourceRG -DiskName
$sourceDiskName -DurationInSecond 86400 -Access 'Read'

$targetDiskSas = Grant-AzDiskAccess -ResourceGroupName $targetRG -DiskName
$targetDiskName -DurationInSecond 86400 -Access 'Write'

azcopy copy $sourceDiskSas.AccessSAS $targetDiskSas.AccessSAS --blob-type PageBlob

Revoke-AzDiskAccess -ResourceGroupName $sourceRG -DiskName $sourceDiskName

Revoke-AzDiskAccess -ResourceGroupName $targetRG -DiskName $targetDiskName
```

Next steps

Now that you've successfully uploaded a VHD to a managed disk, you can attach your disk to a VM and begin using it.

To learn how to attach a data disk to a VM, see our article on the subject: Attach a data disk to a Windows VM with PowerShell. To use the disk as the OS disk, see Create a Windows VM from a specialized disk.

If you have additional questions, see the section on uploading a managed disk in the FAQ.

Last updated on 02/19/2026

Upload a VHD to Azure or copy a managed disk to another region - Azure CLI

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This article explains how to either upload a VHD from your local machine to an Azure Managed Disk or copy a managed disk to another region, using AzCopy. This process, direct upload, enables you to upload a VHD up to 32 TiB in size directly into a managed disk. Currently, direct upload is supported for Ultra Disks, Premium SSD v2, Premium SSD, Standard SSD, and Standard HDD.

If you're providing a backup solution for IaaS VMs in Azure, you should use direct upload to restore customer backups to managed disks. When uploading a VHD from a source external to Azure, speeds depend on your local bandwidth. When uploading or copying from an Azure VM, your bandwidth would be the same as Standard HDDs.

If you're using Microsoft Entra ID to control resource access, you can use it to restrict uploading of Azure Managed Disks. See Secure downloads and uploads of Azure Managed Disks for details.

Get started

If you'd prefer to upload disks through a GUI, you can do so using Azure Storage Explorer. For details refer to: Use Azure Storage Explorer to manage Azure Managed Disks

Prerequisites

- Download the latest version of AzCopy v10.
- Install the Azure CLI.
- If you intend to upload a VHD from on-premises: A fixed size VHD that has been prepared for Azure, stored locally.
- Or, a managed disk in Azure, if you intend to perform a copy action.

To upload your VHD to Azure, you need to create an empty managed disk that is configured for this upload process. Before you create one, there's some additional information you should know about these disks.

This kind of managed disk has two unique states:

- ReadToUpload, which means the disk is ready to receive an upload but, no secure access signature (SAS) has been generated.
- ActiveUpload, which means that the disk is ready to receive an upload and the SAS has been generated.

> **Note**
>
> While in either of these states, the managed disk will be billed at Standard HDD pricing, regardless of the actual type of disk. For example, a P10 will be billed as an S10. This will be true until `revoke-access` is called on the managed disk, which is required in order to attach the disk to a VM.

Create an empty managed disk

Before you can create an empty Standard HDD for uploading, you need the file size of the VHD you want to upload, in bytes. To get that, you can use either `wc -c <yourFileName>.vhd` or `ls -al <yourFileName>.vhd`. This value is used when specifying the **--upload-size-bytes** parameter.

Create an empty Standard HDD for uploading by specifying both the **--for-upload** parameter and the **--upload-size-bytes** parameter in a disk create cmdlet:

Replace `<yourdiskname>`, `<yourresourcegroupname>`, `<yourregion>` with values of your choosing. The `--upload-size-bytes` parameter contains an example value of `34359738880`, replace it with a value appropriate for you.

> **Important**
>
> If you're creating an OS disk, add `--hyper-v-generation <yourGeneration>` to `az disk create`.
>
> If you're using Microsoft Entra ID to secure disk uploads, add `-dataAccessAuthmode 'AzureActiveDirectory'`.
>
> When uploading to an Ultra Disk or Premium SSD v2 you need to select the correct sector size of the target disk. If you're using a VHDX file with a 4k logical sector size, the target disk must be set to 4k. If you're using a VHD file with a 512 logical sector size, the target disk must be set to 512.
>
> VHDX files with logical sector size of 512k aren't supported.

Azure CLI

```
##For Ultra Disk or Premium SSD v2, add --logical-sector-size and specify either
512 or 4096, depending on if you're using a VHD or VHDX

az disk create -n <yourdiskname> -g <yourresourcegroupname> -l <yourregion> --os-
type Linux --for-upload --upload-size-bytes 34359738880 --sku standard_lrs
```

If you would like to upload a different disk type, replace **standard_lrs** with **premium_lrs**, **premium_zrs**, **standardssd_lrs**, **standardssd_zrs**, **premiumv2_lrs**, or **ultrassd_lrs**.

(Optional) Grant access to the disk

If you're using Microsoft Entra ID to secure disk uploads, you need to assign RBAC permissions to grant access to the disk and generate a writeable SAS.

Azure CLI

```
az role assignment create --assignee "{assignee}" \
--role "{Data Operator for Managed Disks}" \
--scope
"/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/{prov
iderName}/{resourceType}/{resourceSubType}/{diskName}"
```

Generate writeable SAS

Now that you've created an empty managed disk that is configured for the upload process, you can upload a VHD to it. To upload a VHD to the disk, you need a writeable SAS, so that you can reference it as the destination for your upload.

> **ⓘ Important**
>
> On February 15th, 2025, the Shared Access Signature (SAS) access time for disks and snapshots will be limited to a maximum of 60 days. Trying to generate a SAS with an expiration longer than 60 days results in an error. Any existing disk or snapshot SAS created with an expiration longer than 60 days may stop working 60 days after the date of creation and will result in a 403 error during authorization.

If a managed disk or snapshot SAS's expiration is longer than 60 days, revoke its access, and generate a new SAS that requests access for 60 days (5,184,000 seconds) or less. Improve your overall security by using SAS with shorter expiration dates. Make these changes before February 15, 2025 to prevent service interruption. The following links can be used to find, revoke, and request a new SAS.

- To check if a disk has an active SAS, you can either use the REST API, the Azure CLI, or the Azure PowerShell module, and examine the **DiskState** property.
- To revoke a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module.
- To create a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module, and set the access duration to 5,184,000 seconds or less.

To generate a writable SAS of your empty managed disk, replace `<yourdiskname>` and `<yourresourcegroupname>`, then use the following command:

Azure CLI

```
az disk grant-access -n <yourdiskname> -g <yourresourcegroupname> --access-level
Write --duration-in-seconds 86400
```

Sample returned value:

Output

```
{
  "accessSas": "https://md-impexp-
t0rdsfgsdfg4.blob.core.windows.net/w2c3mj0ksfgl/abcd?sv=2017-04-
17&sr=b&si=600a9281-d39e-4cc3-91d2-
923c4a696537&sig=xXaT6mFgf139ycT87CADyFxb%2BnPXBElYirYRlbnJZbs%3D"
}
```

Upload a VHD or VHDX

Now that you have a SAS for your empty managed disk, you can use it to set your managed disk as the destination for your upload command.

Use AzCopy v10 to upload your local VHD or VHDX file to a managed disk by specifying the SAS URI you generated.

This upload has the same throughput as the equivalent Standard HDD. For example, if you have a size that equates to S4, you'll have a throughput of up to 60 MiB/s. But, if you have a size that equates to S70, you'll have a throughput of up to 500 MiB/s.

Bash

```
AzCopy.exe copy "c:\somewhere\mydisk.vhd" "sas-URI" --blob-type PageBlob
```

After the upload is complete, and you no longer need to write any more data to the disk, revoke the SAS. Revoking the SAS changes the state of the managed disk and allow you to

attach the disk to a VM.

Replace `<yourdiskname>` and `<yourresourcegroupname>`, then use the following command to make the disk usable:

Azure CLI

```
az disk revoke-access -n <yourdiskname> -g <yourresourcegroupname>
```

Copy a managed disk

Direct upload also simplifies the process of copying a managed disk. You can either copy within the same region or cross-region (to another region).

The following script does this for you. The process is similar to the steps described earlier, with some differences since you're working with an existing disk.

> **ⓘ Important**
>
> You need to add an offset of 512 when you're providing the disk size in bytes of a managed disk from Azure. This is because Azure omits the footer when returning the disk size. The copy will fail if you don't do this. The following script already does this for you.

Replace the `<sourceResourceGroupHere>`, `<sourceDiskNameHere>`, `<targetDiskNameHere>`, `<targetResourceGroupHere>`, and `<yourTargetLocationHere>` (an example of a location value would be uswest2) with your values, then run the following script in order to copy a managed disk.

> **Tip**
>
> If you are creating an OS disk, add `--hyper-v-generation <yourGeneration>` to `az disk create`.

Azure CLI

```
sourceDiskName=<sourceDiskNameHere>
sourceRG=<sourceResourceGroupHere>
targetDiskName=<targetDiskNameHere>
targetRG=<targetResourceGroupHere>
targetLocation=<yourTargetLocationHere>
#Expected value for OS is either "Windows" or "Linux"
targetOS=<yourOSTypeHere>
```

```
sourceDiskSizeBytes=$(az disk show -g $sourceRG -n $sourceDiskName --query
'[diskSizeBytes]' -o tsv)

az disk create -g $targetRG -n $targetDiskName -l $targetLocation --os-type
$targetOS --for-upload --upload-size-bytes $(($sourceDiskSizeBytes+512)) --sku
standard_lrs

targetSASURI=$(az disk grant-access -n $targetDiskName -g $targetRG  --access-level
Write --duration-in-seconds 86400 --query [accessSas] -o tsv)

sourceSASURI=$(az disk grant-access -n $sourceDiskName -g $sourceRG --duration-in-
seconds 86400 --query [accessSas] -o tsv)

azcopy copy $sourceSASURI $targetSASURI --blob-type PageBlob

az disk revoke-access -n $sourceDiskName -g $sourceRG

az disk revoke-access -n $targetDiskName -g $targetRG
```

Next steps

Now that you've successfully uploaded a VHD to a managed disk, you can attach the disk as a data disk to an existing VM or attach the disk to a VM as an OS disk, to create a new VM.

If you've more questions, see the uploading a managed disk section in the FAQ.

Last updated on 02/19/2026

Export/Copy a snapshot to a storage account in different region with CLI

Article • 08/22/2024

This script exports a managed snapshot to a storage account in different region. It first generates the SAS URI of the snapshot and then uses it to copy it to a storage account in different region. Use this script to maintain backup of your managed disks in different region for disaster recovery.

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Run the script

Azure CLI

```
#Provide the subscription Id where snapshot is created
subscriptionId="<subscriptionId>"

#Provide the name of your resource group where snapshot is created
resourceGroupName=myResourceGroupName

#Provide the snapshot name
snapshotName=mySnapshotName

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/azure/storage/storage-dotnet-shared-access-signature-part-1
sasExpiryDuration=3600

#Provide storage account name where you want to copy the snapshot.
storageAccountName=mystorageaccountname

#Name of the storage container where the downloaded snapshot will be stored
storageContainerName=mystoragecontainername
```

```
#Provide the key of the storage account where you want to copy snapshot.
storageAccountKey=mystorageaccountkey

#Provide the name of the VHD file to which snapshot will be copied.
destinationVHDFileName=myvhdfilename

az account set --subscription $subscriptionId

sas=$(az snapshot grant-access --resource-group $resourceGroupName --name
$snapshotName --duration-in-seconds $sasExpiryDuration --query [accessSas] -
o tsv)

az storage blob copy start --destination-blob $destinationVHDFileName --
destination-container $storageContainerName --account-name
$storageAccountName --account-key $storageAccountKey --source-uri $sas
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name myResourceGroupName
```

Sample reference

This script uses following commands to generate SAS URI for a managed snapshot and copies the snapshot to a storage account using SAS URI. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az snapshot grant-access	Generates read-only SAS that is used to copy underlying VHD file to a storage account or download it to on-premises
az storage blob copy start	Copies a blob asynchronously from one storage account to another

Next steps

Create a managed disk from a VHD

Create a virtual machine from a managed disk

For more information on the Azure CLI, see Azure CLI documentation.

Additional virtual machine and managed disks CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Export/Copy a managed disk to a storage account using the Azure CLI

Article • 08/22/2024

This script exports the underlying VHD of a managed disk to a storage account in same or different region. It first generates the SAS URI of the managed disk and then uses it to copy the VHD to a storage account. Use this script to copy managed disks to another region for regional expansion. If you want to publish the VHD file of a managed disk in Azure Marketplace, you can use this script to copy the VHD file to a storage account and then generate a SAS URI of the copied VHD to publish it in the Marketplace.

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Run the script

Azure CLI

```
#Provide the subscription Id where managed disk is created
subscriptionId="<subscriptionId>"

#Provide the name of your resource group where managed disk is created
resourceGroupName=myResourceGroupName

#Provide the managed disk name
diskName=myDiskName

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/azure/storage/storage-dotnet-shared-access-signature-part-1
sasExpiryDuration=3600

#Provide storage account name where you want to copy the underlying VHD file of the managed disk.
```

```
storageAccountName=mystorageaccountname

#Name of the storage container where the downloaded VHD will be stored
storageContainerName=mystoragecontainername

#Provide the key of the storage account where you want to copy the VHD
storageAccountKey=mystorageaccountkey

#Provide the name of the destination VHD file to which the VHD of the
managed disk will be copied.
destinationVHDFileName=myvhdfilename.vhd

az account set --subscription $subscriptionId

sas=$(az disk grant-access --resource-group $resourceGroupName --name
$diskName --duration-in-seconds $sasExpiryDuration --query [accessSas] -o
tsv)

az storage blob copy start --destination-blob $destinationVHDFileName --
destination-container $storageContainerName --account-name
$storageAccountName --account-key $storageAccountKey --source-uri $sas
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name myResourceGroupName
```

Sample reference

This script uses following commands to generate the SAS URI for a managed disk and copies the underlying VHD to a storage account using the SAS URI. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az disk grant-access	Generates read-only SAS that is used to copy the underlying VHD file to a storage account or download it to on-premises
az storage blob copy start	Copies a blob asynchronously from one storage account to another

Next steps

Create a managed disk from a VHD

Create a virtual machine from a managed disk

For more information on the Azure CLI, see Azure CLI documentation.

Additional virtual machine and managed disks CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Export/Copy managed snapshots as VHD to a storage account in different region with PowerShell

Article • 01/23/2025

This script exports a managed snapshot to a storage account in different region. It first generates the SAS URI of the snapshot and then uses it to copy it to a storage account in different region. Use this script to maintain backup of your managed disks in different region for disaster recovery.

If needed, install the Azure PowerShell module using the instructions found in the Azure PowerShell guide, and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the .ssh directory of your user profile.

If you don't have an Azure subscription, create an Azure free account before you begin.

Sample script

PowerShell

```
#Provide the subscription Id of the subscription where snapshot is created
$subscriptionId = "yourSubscriptionId"

#Provide the name of your resource group where snapshot is created
$resourceGroupName ="yourResourceGroupName"

#Provide the snapshot name
$snapshotName = "yourSnapshotName"

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/en-us/Az.Storage/storage-dotnet-shared-access-signature-part-1
$sasExpiryDuration = "3600"

#Provide storage account name where you want to copy the snapshot.
$storageAccountName = "yourstorageaccountName"

#Name of the storage container where the downloaded snapshot will be stored
$storageContainerName = "yourstoragecontainername"

#Provide the key of the storage account where you want to copy snapshot.
$storageAccountKey = 'yourStorageAccountKey'
```

```
#Provide the name of the VHD file to which snapshot will be copied.
$destinationVHDFileName = "yourvhdfilename"


# Set the context to the subscription Id where Snapshot is created
Select-AzSubscription -SubscriptionId $SubscriptionId

#Generate the SAS for the snapshot
$sas = Grant-AzSnapshotAccess -ResourceGroupName $ResourceGroupName -
SnapshotName $SnapshotName  -DurationInSecond $sasExpiryDuration -Access
Read
#Create the context for the storage account which will be used to copy
snapshot to the storage account
$destinationContext = New-AzStorageContext -StorageAccountName
$storageAccountName -StorageAccountKey $storageAccountKey

#Copy the snapshot to the storage account
Start-AzStorageBlobCopy -AbsoluteUri $sas.AccessSAS -DestContainer
$storageContainerName -DestContext $destinationContext -DestBlob
$destinationVHDFileName
```

Script explanation

This script uses following commands to generate SAS URI for a managed snapshot and copies the snapshot to a storage account using SAS URI. Each command in the table links to command specific documentation.

Expand table

Command	Notes
Grant-AzSnapshotAccess	Generates SAS URI for a snapshot that is used to copy it to a storage account.
New-AzStorageContext	Creates a storage account context using the account name and key. This context can be used to perform read/write operations on the storage account.
Start-AzStorageBlobCopy	Copies the underlying VHD of a snapshot to a storage account

Next steps

Create a managed disk from a VHD

Create a virtual machine from a managed disk

For more information on the Azure PowerShell module, see Azure PowerShell documentation.

Additional virtual machine PowerShell script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Export/Copy the VHD of a managed disk to a storage account in different region with PowerShell (Windows)

Article • 08/23/2024

This script exports the VHD of a managed disk to a storage account in different region. It first generates the SAS URI of the managed disk and then uses it to copy the underlying VHD to a storage account in different region. Use this script to copy managed disks to another region for regional expansion.

If needed, install the Azure PowerShell module using the instructions found in the Azure PowerShell guide, and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the .ssh directory of your user profile.

If you don't have an Azure subscription, create an Azure free account before you begin.

Sample script

PowerShell

```
#Provide the subscription Id of the subscription where managed disk is
created
$subscriptionId = "yourSubscriptionId"

#Provide the name of your resource group where managed is created
$resourceGroupName ="yourResourceGroupName"

#Provide the managed disk name
$diskName = "yourDiskName"

#Provide Shared Access Signature (SAS) expiry duration in seconds e.g. 3600.
#Know more about SAS here: https://docs.microsoft.com/en-
us/Az.Storage/storage-dotnet-shared-access-signature-part-1
$sasExpiryDuration = "3600"

#Provide storage account name where you want to copy the underlying VHD of
the managed disk.
$storageAccountName = "yourstorageaccountName"

#Name of the storage container where the downloaded VHD will be stored
$storageContainerName = "yourstoragecontainername"
```

```
#Provide the key of the storage account where you want to copy the VHD of
the managed disk.
$storageAccountKey = 'yourStorageAccountKey'

#Provide the name of the destination VHD file to which the VHD of the
managed disk will be copied.
$destinationVHDFileName = "yourvhdfilename"

#Set the value to 1 to use AzCopy tool to download the data. This is the
recommended option for faster copy.
#Download AzCopy v10 from the link here: https://docs.microsoft.com/en-
us/azure/storage/common/storage-use-azcopy-v10
#Ensure that AzCopy is downloaded in the same folder as this file
#If you set the value to 0 then Start-AzStorageBlobCopy will be used. Azure
storage will asynchronously copy the data.
$useAzCopy = 1

# Set the context to the subscription Id where managed disk is created
Select-AzSubscription -SubscriptionId $SubscriptionId

#Generate the SAS for the managed disk
$sas = Grant-AzDiskAccess -ResourceGroupName $ResourceGroupName -DiskName
$diskName -DurationInSecond $sasExpiryDuration -Access Read

#Create the context of the storage account where the underlying VHD of the
managed disk will be copied
$destinationContext = New-AzStorageContext -StorageAccountName
$storageAccountName -StorageAccountKey $storageAccountKey

#Copy the VHD of the managed disk to the storage account
if($useAzCopy -eq 1)
{
    $containerSASURI = New-AzStorageContainerSASToken -Context
$destinationContext -ExpiryTime(get-date).AddSeconds($sasExpiryDuration) -
FullUri -Name $storageContainerName -Permission rw
    azcopy copy $sas.AccessSAS $containerSASURI

}else{

    Start-AzStorageBlobCopy -AbsoluteUri $sas.AccessSAS -DestContainer
$storageContainerName -DestContext $destinationContext -DestBlob
$destinationVHDFileName
}
```

Script explanation

This script uses the following commands to generate SAS URI of a managed disk and copies the underlying VHD to a storage account using the SAS URI. Each command in the table links to the command specific documentation.

Expand table

Command	Notes
Grant-AzDiskAccess	Generates SAS URI for a managed disk that is used to copy the underlying VHD to a storage account.
New-AzureStorageContext	Creates a storage account context using the account name and key. This context can be used to perform read/write operations on the storage account.
Start-AzureStorageBlobCopy	Copies the underlying VHD of a snapshot to a storage account

Next steps

Create a managed disk from a VHD

Create a virtual machine from a managed disk

For more information on the Azure PowerShell module, see Azure PowerShell documentation.

Additional virtual machine PowerShell script samples can be found in the Azure Windows VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Download a Linux VHD from Azure

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article describes how to download a Linux virtual hard disk (VHD) file from Azure. To download a VHD, the disk can't be attached to a running VM, which means the VM experiences downtime. Some configurations can safely avoid downtime by snapshotting the disk and downloading the VHD from the snapshot.

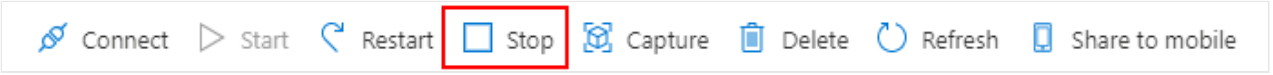
If you're using Microsoft Entra ID to control resource access, you can use it to restrict uploading of Azure Managed Disks. For more information, see Secure downloads and uploads of Azure Managed Disks.

Stop the VM

You can't download a VHD from Azure if the VHD is attached to a running VM. If you want to keep the VM running, you can create a snapshot and then download the snapshot.

To stop the VM:

1. Sign in to the Azure portal.
2. On the left menu, select **Virtual Machines**.
3. Select the VM from the list.
4. On the page for the VM, select **Stop**.



Once the VM is stopped,

Alternative: Snapshot the VM disk

> **Note**
>
> If feasible, stop a VM before taking a snapshot of it. Otherwise, the snapshot isn't clean. Snapshots of running VMs are in the same state as if their VMs were power cycled or crashed when you take a snapshot. Usually, this state is safe but it could cause problems if the running applications at the time aren't crash resistant.

Generally, you should only use snapshots of running VMs if the only disk associated with them is a single OS disk. If a VM has one or more data disks, stop the VM before creating a snapshot of the OS or data disks.

Take a snapshot of the disk to download.

1. Select the VM in the portal.
2. Select **Disks** in the left menu and then select the disk you want to snapshot. The details of the disk are displayed.
3. Select **Create Snapshot** from the menu at the top of the page. The **Create snapshot** page opens.
4. In **Name**, type a name for the snapshot.
5. For **Snapshot type**, select **Full** or **Incremental**.
6. When you're done, select **Review + create**.

After you create your snapshot, you can use it to download a VHD or create another VM.

Generate SAS URL

To download the VHD file, you need to generate a shared access signature (SAS) URL. When the URL is generated, an expiration time is assigned to the URL.

Important

On February 15th, 2025, the Shared Access Signature (SAS) access time for disks and snapshots will be limited to a maximum of 60 days. Trying to generate a SAS with an expiration longer than 60 days results in an error. Any existing disk or snapshot SAS created with an expiration longer than 60 days may stop working 60 days after the date of creation and will result in a 403 error during authorization.

If a managed disk or snapshot SAS's expiration is longer than 60 days, revoke its access, and generate a new SAS that requests access for 60 days (5,184,000 seconds) or less. Improve your overall security by using SAS with shorter expiration dates. Make these changes before February 15, 2025 to prevent service interruption. The following links can be used to find, revoke, and request a new SAS.

- To check if a disk has an active SAS, you can either use the REST API, the Azure CLI, or the Azure PowerShell module, and examine the **DiskState** property.
- To revoke a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module.

- To create a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module, and set the access duration to 5,184,000 seconds or less.

Portal

1. On the menu of the page for the VM, select **Disks**.
2. Select the operating system disk for the VM, and then select **Disk Export**.
3. If necessary, update the value of **URL expires in (seconds)** to give you enough time to complete the download. The default is 3,600 seconds (one hour).
4. Select **Generate URL**.

Download VHD

> **Note**
>
> If you're using Microsoft Entra ID to secure managed disk downloads, the user downloading the VHD needs the appropriate RBAC permissions.

Portal

1. Under the generated URL, select **Download the VHD file**.



2. You might need to select **Save** in the browser to start the download. The default name for the VHD file is *abcd*.

Next steps

- Learn how to upload and create a Linux VM from custom disk with the Azure CLI.
- Manage Azure disks the Azure CLI.

Last updated on 02/19/2026

Download a Windows VHD from Azure

Summarize this article for me

Applies to: ✔ Windows VMs

This article explains how to download a Windows virtual hard disk (VHD) file from Azure. To download a VHD, the disk can't be attached to a running VM, which means the VM experiences downtime. However, some configurations can safely avoid downtime by snapshotting the disk and downloading the VHD from the snapshot.

If you're using Microsoft Entra ID to control resource access, you can use it to restrict uploading of Azure Managed Disks. For more information, see Secure downloads and uploads of Azure Managed Disks.

Optional: Generalize the VM

If you want to use the VHD as an image to create other VMs, use Sysprep to generalize the operating system. Otherwise, you need to make a copy of the disk for each VM you want to create.

To use the VHD as an image to create other VMs, generalize the VM.

1. Sign in to the Azure portal.
2. Connect to the VM.
3. On the VM, open the Command Prompt window as an administrator.
4. Change the directory to *%windir%\system32\sysprep* and run sysprep.exe.
5. In the System Preparation Tool dialog box, select **Enter System Out-of-Box Experience (OOBE)**, and make sure that **Generalize** is selected.
6. In Shutdown Options, select **Shutdown**, and then select **OK**.

If you don't want to generalize your current VM, you can still create a generalized image by first making a snapshot of the OS disk, creating a new VM from the snapshot, and then generalizing the copy.

Stop the VM

You can't download a VHD from Azure if it's attached to a running VM. If you want to keep the VM running, you can create a snapshot and then download the snapshot.

1. On the Hub menu in the Azure portal, select **Virtual Machines**.
2. Select the VM from the list.

3. On the blade for the VM, select **Stop**.

Alternative: Snapshot the VM disk

> **Note**
>
> If feasible, stop a VM before taking a snapshot of it, otherwise the snapshot isn't clean. Snapshots of running VMs are in the same state as if their VMs were power cycled or crashed when you take a snapshot. Usually this state is safe but, it could cause problems if the running applications aren't crash resistant.
>
> Generally, you should only use snapshots of running VMs if the only disk associated with them is a single OS disk. If a VM has one or more data disks, stop the VM before creating a snapshot of the OS or data disks.

Take a snapshot of the disk to download.

1. Select the VM in the portal.
2. Select **Disks** in the left menu and then select the disk you want to snapshot. The details of the disk are displayed.
3. Select **Create Snapshot** from the menu at the top of the page. The **Create snapshot** page opens.
4. In **Name**, type a name for the snapshot.
5. For **Snapshot type**, select **Full** or **Incremental**.
6. When you're done, select **Review + create**.

Your snapshot is created shortly, and you can then use it to download or create another VM.

Generate download URL

To download the VHD file, you need to generate a shared access signature (SAS) URL. When you generate the URL, you assign an expiration time to it.

> **Important**
>
> On February 15th, 2025, the Shared Access Signature (SAS) access time for disks and snapshots will be limited to a maximum of 60 days. Trying to generate a SAS with an expiration longer than 60 days results in an error. Any existing disk or snapshot SAS created with an expiration longer than 60 days may stop working 60 days after the date of creation and will result in a 403 error during authorization.

If a managed disk or snapshot SAS's expiration is longer than 60 days, revoke its access, and generate a new SAS that requests access for 60 days (5,184,000 seconds) or less. Improve your overall security by using SAS with shorter expiration dates. Make these changes before February 15, 2025 to prevent service interruption. The following links can be used to find, revoke, and request a new SAS.

- To check if a disk has an active SAS, you can either use the REST API, the Azure CLI, or the Azure PowerShell module, and examine the **DiskState** property.
- To revoke a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module.
- To create a SAS, you can use either the REST API, the Azure CLI, or the Azure PowerShell module, and set the access duration to 5,184,000 seconds or less.

Portal

1. On the page for the VM, select **Disks** in the left menu.
2. Select the operating system disk for the VM.
3. On the page for the disk, select **Disk Export** from the left menu.
4. The default expiration time of the URL is *3,600 seconds (one hour). You might need to increase this value for Windows OS disks or large data disks. In these situations, **36000** seconds (10 hours) is usually sufficient.
5. Select **Generate URL**.

> **Note**
>
> When downloading a Windows OS disk, you might need a longer expiration time to download a large VHD file. Large VHDs can take up to several hours to download depending on your connection and the size of the VM.
>
> While the SAS URL is active, attempting to start the VM results in the error **There is an active shared access signature outstanding for disk** *diskname*. You can revoke the SAS URL by selecting **Cancel export** on the **Disk Export** page.

Download VHD

> **Note**

If you're using Microsoft Entra ID to secure managed disk downloads, the user downloading the VHD needs the appropriate RBAC permissions.

Portal

1. Under the generated URL, select **Download the VHD file**.
2. You might need to select **Save** in your browser to start the download. The default name for the VHD file is *abcd*.

Next steps

- Learn how to upload a VHD file to Azure.
- Create managed disks from unmanaged disks in a storage account.
- Manage Azure disks with PowerShell.

Last updated on 02/19/2026

Convert the disk type of an Azure Managed Disk

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows

There are five disk types of Azure Managed Disks: Azure Ultra Disks, Premium SSD v2, Premium SSD, Standard SSD, and Standard HDD. You can easily switch between Premium SSD, Standard SSD, and Standard HDD based on your performance needs. Premium SSD and Standard SSD are also available with Zone-redundant storage. For most cases, you can't yet switch from or to an Ultra Disk, you must deploy a new one with a snapshot of an existing disk. However, you can switch from existing disks to a Premium SSD v2. See Convert Premium SSD v2 disks for details.

This functionality isn't supported for unmanaged disks. But you can easily convert an unmanaged disk to a managed disk with CLI or PowerShell to be able to switch between disk types.

Before you begin

Because conversion requires a restart of the virtual machine (VM), schedule the migration of your disk during a pre-existing maintenance window.

Restrictions

- You can only change disk type twice per day.
- You can only change the disk type of managed disks. If your disk is unmanaged, convert it to a managed disk with CLI or PowerShell to switch between disk types.
- You can't migrate a disk to Premium SSD v2 if your original disk was created from an Azure Compute Gallery image.

Change the type of an individual managed disk

For your dev/test workload, you might want a mix of Standard and Premium disks to reduce your costs. You can choose to upgrade only those disks that need better performance. This example shows how to convert a single VM disk from Standard to Premium storage. However, by changing the $storageType variable in this example, you can convert the VM's disks type to Standard SSD or Standard HDD. To use Premium managed disks, your VM must use a VM size that supports Premium storage. You can also use these examples to change a disk from Locally

redundant storage (LRS) disk to a Zone-redundant storage (ZRS) disk or vice-versa. This example also shows how to switch to a size that supports Premium storage:

Azure PowerShell

> **Note**
>
> You can change the type of an existing disk to a Premium SSD v2 the same way you would for other disk types. To learn about the feature see Premium SSD v2 migration.

Azure PowerShell

```
$diskName = 'yourDiskName'
# resource group that contains the managed disk
$rgName = 'yourResourceGroupName'
# Choose between Standard_LRS, StandardSSD_LRS, StandardSSD_ZRS, Premium_ZRS,
and Premium_LRS based on your scenario
$storageType = 'Premium_LRS'
# Premium capable size
$size = 'Standard_DS2_v2'

$disk = Get-AzDisk -DiskName $diskName -ResourceGroupName $rgName

# Get parent VM resource
$vmResource = Get-AzResource -ResourceId $disk.ManagedBy

# Stop and deallocate the VM before changing the storage type
Stop-AzVM -ResourceGroupName $vmResource.ResourceGroupName -Name
$vmResource.Name -Force

$vm = Get-AzVM -ResourceGroupName $vmResource.ResourceGroupName -Name
$vmResource.Name

# Change the VM size to a size that supports Premium storage
# Skip this step if converting storage from Premium to Standard
$vm.HardwareProfile.VmSize = $size
Update-AzVM -VM $vm -ResourceGroupName $rgName

# Update the storage type
$disk.Sku =
[Microsoft.Azure.Management.Compute.Models.DiskSku]::new($storageType)
$disk | Update-AzDisk

Start-AzVM -ResourceGroupName $vm.ResourceGroupName -Name $vm.Name
```

Switch all managed disks of a VM from one account to another

This example shows how to convert all of a VM's disks to premium storage. However, by changing the $storageType variable in this example, you can convert the VM's disks type to Standard SSD or Standard HDD. To use Premium managed disks, your VM must use a VM size that supports Premium storage. This example also switches to a size that supports premium storage:

Azure PowerShell

Azure PowerShell

```
# Name of the resource group that contains the VM
$rgName = 'yourResourceGroup'

# Name of the your virtual machine
$vmName = 'yourVM'

# Choose between Standard_LRS, StandardSSD_LRS, StandardSSD_ZRS, Premium_ZRS,
Premium_LRS, and PremiumV2_LRS based on your scenario
$storageType = 'Premium_LRS'

# Premium capable size
# Required only if converting storage from Standard to Premium
$size = 'Standard_DS2_v2'

# Stop and deallocate the VM before changing the size
Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force

$vm = Get-AzVM -Name $vmName -resourceGroupName $rgName

# Change the VM size to a size that supports Premium storage
# Skip this step if converting storage from Premium to Standard
$vm.HardwareProfile.VmSize = $size
Update-AzVM -VM $vm -ResourceGroupName $rgName

# Get all disks in the resource group of the VM
$vmDisks = Get-AzDisk -ResourceGroupName $rgName

# For disks that belong to the selected VM, convert to Premium storage
foreach ($disk in $vmDisks)
{
    if ($disk.ManagedBy -eq $vm.Id)
    {
        $disk.Sku =
[Microsoft.Azure.Management.Compute.Models.DiskSku]::new($storageType)
        $disk | Update-AzDisk
    }
```

```
}

Start-AzVM -ResourceGroupName $rgName -Name $vmName
```

Convert Premium SSD v2 disks

You can switch existing disks to Premium SSD v2 disks the same way you do for other disk types. Premium SSD v2 disks have some limitations, see the Premium SSD v2 limitations section of their article to learn more.

Switching to Premium SSD v2 disks has some additional limitations:

- You can't switch an OS disk to a Premium SSD v2.
- Existing disks can only be directly switched to 512 sector size Premium SSD v2 disks.
- You can only perform 50 conversions at the same time per subscription per region.
- If your existing disk is a shared disk, detach all VMs before changing to Premium SSD v2.
- If your existing disk is using host caching, set it to none before changing to Premium SSD v2.
- If your existing disk is using bursting, disable it before changing to Premium SSD v2.
- If your existing disk is using double encryption, switch to one of the single encryption options before changing to Premium SSD v2.
- You can't directly switch from a Premium SSD v2 to another disk type. If you want to change a Premium SSD v2 to another disk type, migrate using snapshots.
- You can't directly switch from Ultra Disks to Premium SSD v2 disks, migrate using snapshots.
- If your disk has Azure Site Recovery configured on it, disable it before changing to Premium SSD v2.
- If your disk is attached to a VM with Azure Backup enabled, switch to the Enhanced Backup policy before converting to Premium SSD v2.
- If you're using the rest API, use an API version `2020-12-01` or newer for both the Compute Resource Provider and the Disk Resource Provider.
- Until the conversion process from your previous disk type to Premium SSD v2 is completed, the performance of the disk is degraded, and you can't change or rotate customer-managed keys for the disk if they're in use.
  - You can use the following command to check the conversion process, replace `$diskName` and `$resourceGroupName` with your values: `az disk show -n $diskName -g $resourceGroupName --query [completionPercent] -o tsv`

> **Note**

If you're using Azure Backup and you convert a disk to Premium SSD v2, a full snapshot is taken of the new disk. This is a billable event and you'll be charged for that snapshot.

Disable host caching

If your disk is using host caching, you must disable it before converting to Premium SSD v2. You'll need the LUN of the disk you want to disable host caching on. The following script outputs the name of the disks attached to your VM, and their LUNs. You can use this to identify the LUN of the disk. Replace `yourResourceGroup` and `nameOfYourVM` with your own values, then run the script.

Azure CLI

```
myRG="yourResourceGroup"
myVM="nameOfYourVM"

az vm show -g $myRG -n $myVM --query "[storageProfile.dataDisks[].name,
storageProfile.dataDisks[].lun]"
```

Once you've got the disk's LUN, replace `LunHere` with the LUN and run the following command to disable host caching:

Azure CLI

```
lun=LunHere

az vm update --resource-group $myRG --name $myVM --disk-caching $lun=None
```

Disable bursting

If your disk is using bursting, you must disable it before converting to Premium SSD v2. If you enabled bursting within 12 hours, you have to wait until the 13th hour or later to disable it.

You can use the following command to disable disk bursting: `az disk update --name "yourDiskNameHere" --resource-group "yourRGNameHere" --enable-bursting false`

Disable double encryption

If your disk is using double encryption, you must disable it before converting to Premium SSD v2. You can use the following command to change your disk from double encryption to encryption at rest with customer-managed keys:

Azure CLI

```
az disk-encryption-set update --name "nameOfYourDiskEncryptionSetHere" --resource-group "yourRGNameHere" --key-url yourKeyURL --source-vault "yourKeyVaultName" --encryption-type EncryptionAtRestWithCustomerKey
```

Migrate to Premium SSD v2 or Ultra Disk using snapshots

> **Note**
>
> You can change the type of an existing disk to a Premium SSD v2 the same way you would for other disk types. To learn about the feature see Premium SSD v2 migration.

Premium SSD, Standard SSD, and Standard HDDs support both incremental and full snapshots. However, migration using full snapshots is not supported for these disk types. To create Premium SSD v2 or Ultra Disks from a snapshot, you must use incremental snapshots.

Both Premium SSD v2 disks and Ultra Disks have their own set of restrictions. For example, neither can be used as an OS disk, and also aren't available in all regions. See the Premium SSD v2 limitations and Ultra Disk GA scope and limitations sections of their articles for more information.

> **Important**
>
> When migrating a Standard HDD, Standard SSD, or Premium SSD to either an Ultra Disk or Premium SSD v2, the logical sector size must be 512.

Azure PowerShell

The following script migrates a snapshot of a Standard HDD, Standard SSD, or Premium SSD to either an Ultra Disk or a Premium SSD v2.

PowerShell

```
$diskName = "yourDiskNameHere"
$resourceGroupName = "yourResourceGroupNameHere"
$snapshotName = "yourDesiredSnapshotNameHere"

# Valid values are 1, 2, or 3
$zone = "yourZoneNumber"
```

```
#Provide the size of the disks in GB. It should be greater than the VHD file
size.
$diskSize = '128'

#Provide the storage type. Use PremiumV2_LRS or UltraSSD_LRS.
$storageType = 'PremiumV2_LRS'

#Provide the Azure region (e.g. westus) where Managed Disks will be located.
#This location should be same as the snapshot location
#Get all the Azure location using command below:
#Get-AzLocation

#Select the same location as the current disk
#Note that Premium SSD v2 and Ultra Disks are only supported in a select number
of regions
$location = 'eastus'

#When migrating a Standard HDD, Standard SSD, or Premium SSD to either an Ultra
Disk or Premium SSD v2, the logical sector size must be 512
$logicalSectorSize=512

# Get the disk that you need to backup by creating an incremental snapshot
$yourDisk = Get-AzDisk -DiskName $diskName -ResourceGroupName
$resourceGroupName

# Create an incremental snapshot by setting the SourceUri property with the
value of the Id property of the disk
$snapshotConfig=New-AzSnapshotConfig -SourceUri $yourDisk.Id -Location
$yourDisk.Location -CreateOption Copy -Incremental
$snapshot = New-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName
$snapshotName -Snapshot $snapshotConfig

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -
CreateOption Copy -SourceResourceId $snapshot.Id -DiskSizeGB $diskSize -
LogicalSectorSize $logicalSectorSize -Zone $zone

New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName
$diskName
```

Next steps

Make a read-only copy of a VM by using a snapshot.

Last updated on 03/12/2026

Convert a disk from LRS to ZRS

07/02/2025

This article explains how to convert an existing disk from locally redundant storage (LRS) to zone-redundant storage (ZRS).

For conceptual information on ZRS, see Zone-redundant storage for managed disks

Limitations

ZRS for managed disks is only supported with Premium SSD and Standard SSD managed disks. ZRS for managed disks isn't supported with Premium SSD v2 managed disks or Ultra Disks. Force detach only supports data disks.

Regional availability

ZRS disks are available in all regions that have availability zone support. For a list of these regions, see Azure regions list.

Determine infrastructure redundancy

Before you convert a disk from LRS to ZRS, you must first identify whether the disk is zonal or regional. How you migrate your disk changes based on whether it's zonal or regional.

Portal

1. Sign in to the Azure portal
2. Navigate to your disk and look under **Availability zone** on the **Overview** tab.
   a. If the value is **No infrastructure redundancy required** you have a regional disk. Use Regional migration to migrate your disk.
   b. If the value is anything else, you have a zonal disk. Use Zonal migration to migrate your disk.

Regional migration

You can directly change the disk type of regional disks with the following options.

Portal

Follow these steps:

1. Sign in to the Azure portal.
2. Select the VM from the list of **Virtual machines**.
3. If the VM isn't stopped, select **Stop** at the top of the VM **Overview** pane, and wait for the VM to stop.
4. In the pane for the VM, select **Disks** from the menu.
5. Select the disk that you want to convert.
6. Select **Size + performance** from the menu.
7. Change the **Account type** from the original disk type to the desired disk type.
8. Select **Save**, and close the disk pane.

The disk type conversion is instantaneous. You can start your VM after the conversion.

Zonal migration

In this section, you migrate the data from your current managed disks to ZRS managed disks.

If you have a zonal disk, you can't directly change its type. You must take a snapshot and use that snapshot to create a new ZRS disk.

Step 1: Create your snapshot

The easiest and cleanest way to create a snapshot is to take one while the VM is offline. See Snapshots. If you choose this approach, some downtime should be expected. To create a snapshot of your VM using the Azure portal, PowerShell, or Azure CLI, see Create a snapshot of a virtual hard disk

If you're taking a snapshot of a disk that's attached to a running VM, read the guidance in Snapshots before continuing.

> **Note**
>
> The source managed disks remain intact with their current configurations and you'll continue to be billed for them. To avoid this, you must manually delete the disks once you've finished your migration and confirmed the new disks are working. For more information, see Find and delete unattached Azure managed and unmanaged disks.

Step 2: Migrate the data on your managed disks

Now that you have snapshots of your original disks, you can use them to create ZRS managed disks.

Migrate your data to ZRS managed disks

1. Create a ZRS managed disk from the source disk snapshot by using the following Azure CLI snippet:

Azure CLI

```
# Create a new ZRS Managed Disks using the snapshot Id and the SKU supported
storageType=Premium_ZRS
location=westus2

az disk create --resource-group $resourceGroupName --name $diskName --sku
$storageType --size-gb $diskSize --source $snapshotId
```

Step 3: Create a new VM with your new disks

Now that you have migrated your data to ZRS managed disks or zonal managed disks, create a new VM with these new disks set as the OS and data disks:

Azure CLI

```
    az vm create -g MyResourceGroup -n MyVm --attach-os-disk newZonalOSDiskCopy --
attach-data-disks newZonalDataDiskCopy --os-type linux
```

Copy managed disks to same or different subscription with CLI

Article • 08/22/2024

This article contains two scripts. The first script copies a managed disk that's using platform-managed keys to same or different subscription but in the same region. The second script copies a managed disk that's using customer-managed keys to the same or a different subscription in the same region. Either copy only works when the subscriptions are part of the same Microsoft Entra tenant.

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Disks with platform-managed keys

Azure CLI

```
#Provide the subscription Id of the subscription where managed disk exists
sourceSubscriptionId="<subscriptionId>"

#Provide the name of your resource group where managed disk exists
sourceResourceGroupName=mySourceResourceGroupName

#Provide the name of the managed disk
managedDiskName=myDiskName

#Set the context to the subscription Id where managed disk exists
az account set --subscription $sourceSubscriptionId

#Get the managed disk Id
managedDiskId=$(az disk show --name $managedDiskName --resource-group
$sourceResourceGroupName --query [id] -o tsv)

#If managedDiskId is blank then it means that managed disk does not exist.
echo 'source managed disk Id is: ' $managedDiskId

#Provide the subscription Id of the subscription where managed disk will be
```

```
copied to
targetSubscriptionId=6492b1f7-f219-446b-b509-314e17e1efb0

#Name of the resource group where managed disk will be copied to
targetResourceGroupName=mytargetResourceGroupName

#Set the context to the subscription Id where managed disk will be copied to
az account set --subscription $targetSubscriptionId

#Copy managed disk to different subscription using managed disk Id
az disk create --resource-group $targetResourceGroupName --name
$managedDiskName --source $managedDiskId
```

Disks with customer-managed keys

Azure CLI

```
#Provide the subscription Id of the subscription where managed disk exists
sourceSubscriptionId="<subscriptionId>"

#Provide the name of your resource group where managed disk exists
sourceResourceGroupName=mySourceResourceGroupName

#Provide the name of the managed disk
managedDiskName=myDiskName

#Provide the name of the target disk encryption set
diskEncryptionSetName=myName

#Provide the target disk encryption set resource group
diskEncryptionResourceGroup=myGroup

#Set the context to the subscription Id where managed disk exists
az account set --subscription $sourceSubscriptionId

#Get the managed disk Id
managedDiskId=$(az disk show --name $managedDiskName --resource-group
$sourceResourceGroupName --query [id] -o tsv)

#If managedDiskId is blank then it means that managed disk does not exist.
echo 'source managed disk Id is: ' $managedDiskId

#Get the disk encryption set ID
diskEncryptionSetId=$(az disk-encryption-set show --name
$diskEncryptionSetName --resource-group $diskEncryptionResourceGroup)

#Provide the subscription Id of the subscription where managed disk will be
copied to
targetSubscriptionId=6492b1f7-f219-446b-b509-314e17e1efb0

#Name of the resource group where managed disk will be copied to
targetResourceGroupName=mytargetResourceGroupName
```

```
#Set the context to the subscription Id where managed disk will be copied to
az account set --subscription $targetSubscriptionId

#Copy managed disk to different subscription using managed disk Id and disk
encryption set ID
#Add --location parameter to change the location
az disk create -g $targetResourceGroupName -n $managedDiskName --source
$managedDiskId --disk-encryption-set $diskEncrpytonSetId
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name mySourceResourceGroupName
```

Sample reference

This script uses following commands to create a new managed disk in the target subscription using the `Id` of the source managed disk. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az disk show	Gets all the properties of a managed disk using the name and resource group properties of the managed disk. The `Id` property is used to copy the managed disk to different subscription.
az disk create	Copies a managed disk by creating a new managed disk in different subscription using the `Id` and name the parent managed disk.

Next steps

Create a virtual machine from a managed disk

For more information on the Azure CLI, see Azure CLI documentation.

More virtual machine and managed disks CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Copy snapshot of a managed disk to same or different subscription with CLI

Article • 08/22/2024

This article contains two scripts. The first script copies a snapshot of a managed disk that was using platform-managed keys to the same or a different subscription. The second script copies a snapshot of a managed disk that was using customer-managed keys to the same or a different subscription. These scripts can be used for the following scenarios:

- Migrate a snapshot in Premium storage (Premium_LRS) to Standard storage (Standard_LRS or Standard_ZRS) to reduce your cost.
- Migrate a snapshot from locally redundant storage (Premium_LRS, Standard_LRS) to zone redundant storage (Standard_ZRS) to benefit from the higher reliability of ZRS storage.
- Move a snapshot to different subscription in the same region for longer retention.

> **Note**
>
> Both subscriptions must be located under the same tenant

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.

- When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
- Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Disks with platform-managed keys

Azure CLI

```
#Provide the subscription Id of the subscription where snapshot exists
sourceSubscriptionId="<subscriptionId>"
```

```
#Provide the name of your resource group where snapshot exists
sourceResourceGroupName=mySourceResourceGroupName

#Provide the name of the snapshot
snapshotName=mySnapshotName

#Set the context to the subscription Id where snapshot exists
az account set --subscription $sourceSubscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group
$sourceResourceGroupName --query [id] -o tsv)

#If snapshotId is blank then it means that snapshot does not exist.
echo 'source snapshot Id is: ' $snapshotId

#Provide the subscription Id of the subscription where snapshot will be
copied to
#If snapshot is copied to the same subscription then you can skip this step
targetSubscriptionId=6492b1f7-f219-446b-b509-314e17e1efb0

#Name of the resource group where snapshot will be copied to
targetResourceGroupName=mytargetResourceGroupName

#Set the context to the subscription Id where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
az account set --subscription $targetSubscriptionId

#Copy snapshot to different subscription using the snapshot Id
#We recommend you to store your snapshots in Standard storage to reduce
cost. Please use Standard_ZRS in regions where zone redundant storage (ZRS)
is available, otherwise use Standard_LRS
#Please check out the availability of ZRS here:
https://docs.microsoft.com/azure/storage/common/storage-redundancy-
zrs#support-coverage-and-regional-availability
az snapshot create --resource-group $targetResourceGroupName --name
$snapshotName --source $snapshotId --sku Standard_LRS
```

Disks with customer-managed keys

Azure CLI

```
#Provide the subscription Id of the subscription where snapshot exists
sourceSubscriptionId="<subscriptionId>"

#Provide the name of your resource group where snapshot exists
sourceResourceGroupName=mySourceResourceGroupName

#Provide the name of the target disk encryption set
diskEncryptionSetName=myName
```

```
#Provide the target disk encryption set resource group
diskEncryptionResourceGroup=myGroup

#Provide the name of the snapshot
snapshotName=mySnapshotName

#Set the context to the subscription Id where snapshot exists
az account set --subscription $sourceSubscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group
$sourceResourceGroupName --query [id] -o tsv)

#If snapshotId is blank then it means that snapshot does not exist.
echo 'source snapshot Id is: ' $snapshotId

#Get the disk encryption set ID
diskEncryptionSetId=$(az disk-encryption-set show --name
$diskEncryptionSetName --resource-group $diskEncryptionResourceGroup)

#Provide the subscription Id of the subscription where snapshot will be
copied to
#If snapshot is copied to the same subscription then you can skip this step
targetSubscriptionId=6492b1f7-f219-446b-b509-314e17e1efb0

#Name of the resource group where snapshot will be copied to
targetResourceGroupName=mytargetResourceGroupName

#Set the context to the subscription Id where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
az account set --subscription $targetSubscriptionId

#Copy snapshot to different subscription using the snapshot Id
#We recommend you to store your snapshots in Standard storage to reduce
cost. Please use Standard_ZRS in regions where zone redundant storage (ZRS)
is available, otherwise use Standard_LRS
#Please check out the availability of ZRS here:
https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy-
zrs#support-coverage-and-regional-availability
#To change the region, use the --location parameter
az snapshot create -g $targetResourceGroupName -n $snapshotName --source
$snapshotId --disk-encryption-set $diskEncryptionSetID --sku Standard_LRS --
encryption-type EncryptionAtRestWithCustomerKey
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name mySourceResourceGroupName
```

Sample reference

This script uses following commands to create a snapshot in the target subscription using the `Id` of the source snapshot. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az snapshot show	Gets all the properties of a snapshot using the name and resource group properties of the snapshot. The `Id` property is used to copy the snapshot to different subscription.
az snapshot create	Copies a snapshot by creating a snapshot in different subscription using the `Id` and name of the parent snapshot.

Next steps

Create a virtual machine from a snapshot

For more information on the Azure CLI, see Azure CLI documentation.

More virtual machine and managed disks CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Copy snapshot of a managed disk in same subscription or different subscription with PowerShell (Windows)

Article • 08/23/2024

This script copies a snapshot of a managed disk to same or different subscription. Use this script for the following scenarios:

1. Migrate a snapshot in Premium storage (Premium_LRS) to Standard storage (Standard_LRS or Standard_ZRS) to reduce your cost.
2. Migrate a snapshot from locally redundant storage (Premium_LRS, Standard_LRS) to zone redundant storage (Standard_ZRS) to benefit from the higher reliability of ZRS storage.
3. Move a snapshot to different subscription in the same region for longer retention.

If needed, install the Azure PowerShell module using the instructions found in the Azure PowerShell guide, and then run `Connect-AzAccount` to create a connection with Azure. Also, you need to have an SSH public key named `id_rsa.pub` in the .ssh directory of your user profile.

If you don't have an Azure subscription, create an Azure free account before you begin.

Sample script

PowerShell

```
#Provide the subscription Id of the subscription where snapshot exists
$sourceSubscriptionId='yourSourceSubscriptionId'

#Provide the name of your resource group where snapshot exists
$sourceResourceGroupName='yourResourceGroupName'

#Provide the name of the snapshot
$snapshotName='yourSnapshotName'

#Set the context to the subscription Id where snapshot exists
Select-AzSubscription -SubscriptionId $sourceSubscriptionId

#Get the source snapshot
$snapshot= Get-AzSnapshot -ResourceGroupName $sourceResourceGroupName -Name
$snapshotName
```

```
#Provide the subscription Id of the subscription where snapshot will be
copied to
#If snapshot is copied to the same subscription then you can skip this step
$targetSubscriptionId='yourTargetSubscriptionId'

#Name of the resource group where snapshot will be copied to
$targetResourceGroupName='yourTargetResourceGroupName'

#Set the context to the subscription Id where snapshot will be copied to
#If snapshot is copied to the same subscription then you can skip this step
Select-AzSubscription -SubscriptionId $targetSubscriptionId

#We recommend you to store your snapshots in Standard storage to reduce
cost. Please use Standard_ZRS in regions where zone redundant storage (ZRS)
is available, otherwise use Standard_LRS
#Please check out the availability of ZRS here:
https://docs.microsoft.com/en-us/Az.Storage/common/storage-redundancy-
zrs#support-coverage-and-regional-availability
$snapshotConfig = New-AzSnapshotConfig -SourceResourceId $snapshot.Id -
Location $snapshot.Location -CreateOption Copy -SkuName Standard_LRS

#Create a new snapshot in the target subscription and resource group
New-AzSnapshot -Snapshot $snapshotConfig -SnapshotName $snapshotName -
ResourceGroupName $targetResourceGroupName
```

Script explanation

This script uses following commands to create a snapshot in the target subscription using the Id of the source snapshot. Each command in the table links to command specific documentation.

Expand table

Command	Notes
New-AzSnapshotConfig	Creates snapshot configuration that is used for snapshot creation. It includes the resource Id of the parent snapshot and location that is same as the parent snapshot.
New-AzSnapshot	Creates a snapshot using snapshot configuration, snapshot name, and resource group name passed as parameters.

Next steps

Create a virtual machine from a snapshot

For more information on the Azure PowerShell module, see Azure PowerShell documentation.

Additional virtual machine PowerShell script samples can be found in the Azure Windows VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate from Amazon Web Services (AWS) and other platforms to managed disks in Azure

Article • 08/22/2024

Applies to: ✔ Windows VMs

You can upload VHD files from AWS or on-premises virtualization solutions to Azure to create virtual machines (VMs) that use managed disks. Azure managed disks removes the need to manage storage accounts for Azure IaaS VMs. You just specify the type of disk, and the size of that disk that you need, and Azure creates and manages the disk for you.

You can upload either generalized and specialized VHDs.

- **Generalized VHD** - has had all of your personal account information removed using Sysprep.
- **Specialized VHD** - maintains the user accounts, applications, and other state data from your original VM.

> **ⓘ Important**
>
> Before uploading any VHD to Azure, you should follow **Prepare a Windows VHD or VHDX to upload to Azure**

Expand table

Scenario	Documentation
You have existing AWS EC2 instances that you would like to migrate to Azure VMs using managed disks	Move a VM from Amazon Web Services (AWS) to Azure
You have a VM from another virtualization platform that you would like to use as an image to create multiple Azure VMs.	Upload a generalized VHD and use it to create a new VM in Azure
You have a uniquely customized VM that you would like to recreate in Azure.	Upload a specialized VHD to Azure and create a new VM

Overview of managed disks

Azure managed disks simplifies VM management by removing the need to manage storage accounts. Managed disks also benefit from better reliability of VMs in an Availability Set. It ensures that the disks of different VMs in an Availability Set are sufficiently isolated from each other to avoid a single point of failure. It automatically places disks of different VMs in an Availability Set in different Storage scale units (stamps) which limits the impact of single Storage scale unit failures caused due to hardware and software failures. Based on your needs, you can choose from four types of storage options. To learn about the available disk types, see our article Select a disk type.

Plan for the migration to managed disks

This section helps you to make the best decision on VM and disk types.

If you are planning on migrating from unmanaged disks to managed disks, you should be aware that users with the Virtual Machine Contributor role will not be able to change the VM size (as they could pre-conversion). This is because VMs with managed disks require the user to have the Microsoft.Compute/disks/write permission on the OS disks.

Location

Pick a location where Azure managed disks are available. If you are migrating to premium SSDs, also ensure that premium storage is available in the region where you are planning to migrate to. See Azure Services by Region for up-to-date information on available locations.

VM sizes

If you are migrating to premium SSDs, you have to update the size of the VM to premium storage capable size available in the region where VM is located. Review the VM sizes that are premium storage capable. The Azure VM size specifications are listed in Sizes for virtual machines. Review the performance characteristics of virtual machines that work with premium storage and choose the most appropriate VM size that best suits your workload. Make sure that there is sufficient bandwidth available on your VM to drive the disk traffic.

Disk sizes

For information on available disk types and sizes, see What disk types are available in Azure?.

Disk caching policy

Premium Managed Disks

By default, disk caching policy is *Read-Only* for all the Premium data disks, and *Read-Write* for the Premium operating system disk attached to the VM. This configuration setting is recommended to achieve the optimal performance for your application's IOs. For write-heavy or write-only data disks (such as SQL Server log files), disable disk caching so that you can achieve better application performance.

Pricing

Review the pricing for managed disks.

Next Steps

- Before uploading any VHD to Azure, you should follow Prepare a Windows VHD or VHDX to upload to Azure

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use Site Recovery to migrate to Premium Storage

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

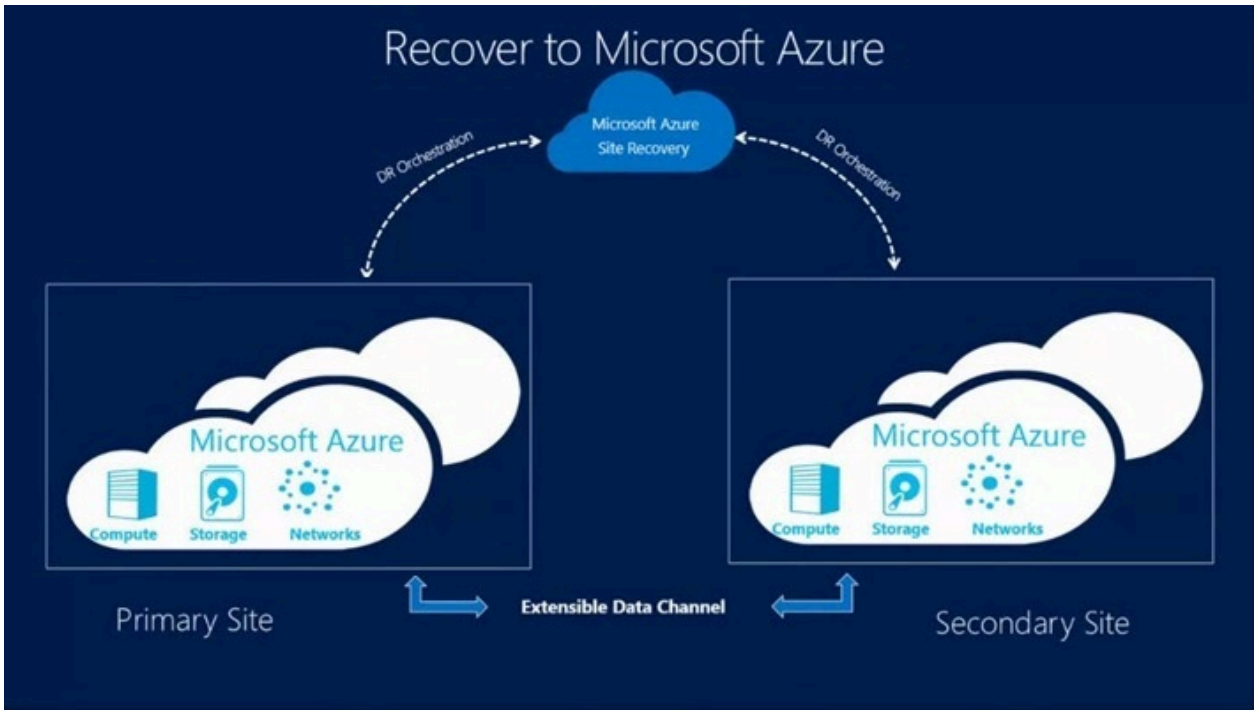
Azure premium SSDs delivers high-performance, low-latency disk support for virtual machines (VMs) that are running I/O-intensive workloads. This guide helps you migrate your VM disks from a standard storage account to a premium storage account by using Azure Site Recovery.

Site Recovery is an Azure service that contributes to your strategy for business continuity and disaster recovery by orchestrating the replication of on-premises physical servers and VMs to the cloud (Azure) or to a secondary datacenter. When outages occur in your primary location, you fail over to the secondary location to keep applications and workloads available. You fail back to your primary location when it returns to normal operation.

Site Recovery provides test failovers to support disaster recovery drills without affecting production environments. You can run failovers with minimal data loss (depending on replication frequency) for unexpected disasters. In the scenario of migrating to Premium Storage, you can use the failover in Site Recovery to migrate target disks to a premium storage account.

We recommend migrating to Premium Storage by using Site Recovery because this option provides minimal downtime. This option also avoids the manual execution of copying disks and creating new VMs. Site Recovery will systematically copy your disks and create new VMs during failover.

Site Recovery supports a number of types of failover with minimal or no downtime. To plan your downtime and estimate data loss, see the types of failover in Site Recovery. If you prepare to connect to Azure VMs after failover, you should be able to connect to the Azure VM by using RDP after failover.



Azure Site Recovery components

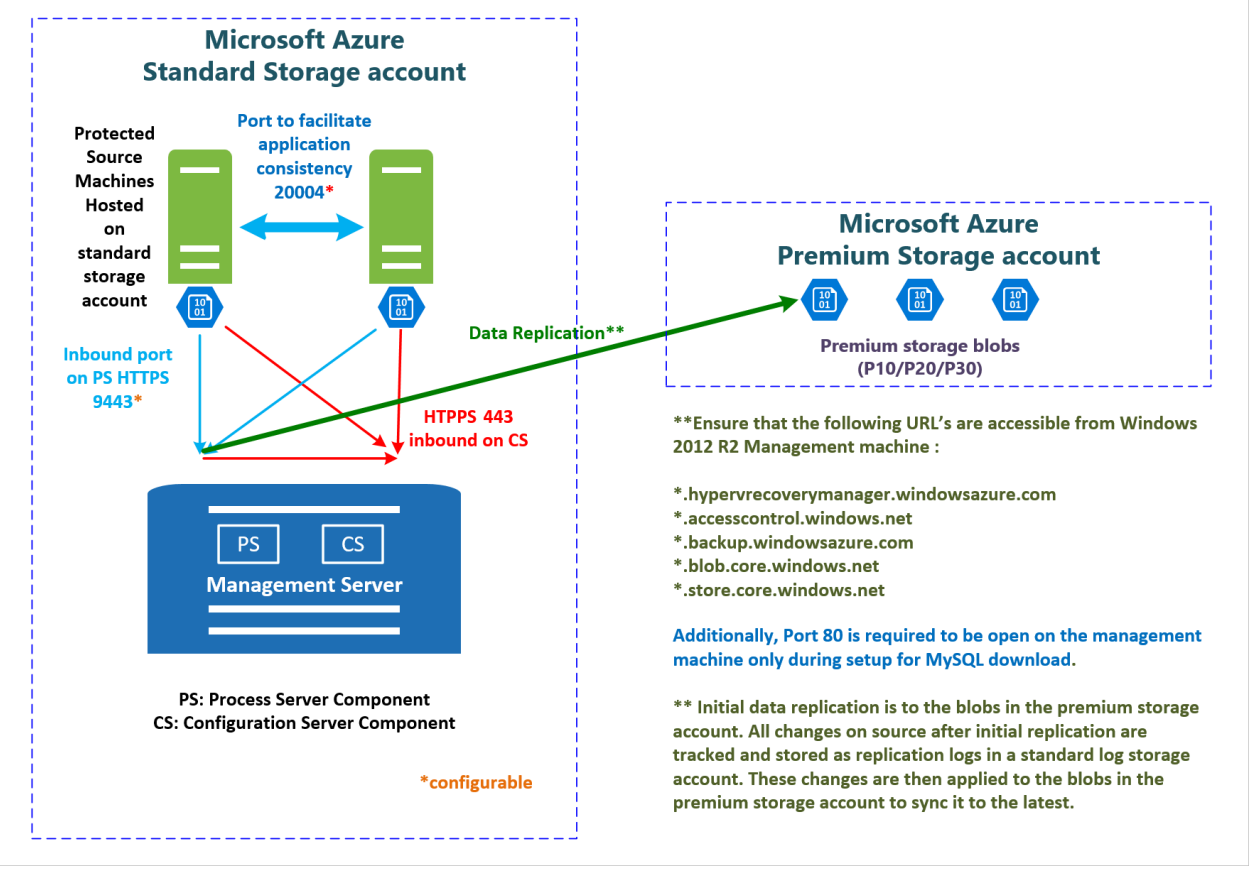
These Site Recovery components are relevant to this migration scenario:

- **Configuration server** is an Azure VM that coordinates communication and manages data replication and recovery processes. On this VM, you run a single setup file to install the configuration server and an additional component, called a process server, as a replication gateway. Read about configuration server prerequisites. You set up the configuration server only once, and you can use it for all migrations to the same region.
- **Process server** is a replication gateway that:

  1. Receives replication data from source VMs.
  2. Optimizes the data with caching, compression, and encryption.
  3. Sends the data to a storage account.

  It also handles push installation of the mobility service to source VMs and performs automatic discovery of source VMs. The default process server is installed on the configuration server. You can deploy additional standalone process servers to scale your deployment. Read about best practices for process server deployment and deploying additional process servers. You set up the process server only once, and you can use it for all migrations to the same region.
- **Mobility service** is a component that is deployed on every standard VM that you want to replicate. It captures data writes on the standard VM and forwards them to the process server. Read about replicated machine prerequisites.

This graphic shows how these components interact:



> **Note**
>
> Site Recovery does not support the migration of Storage Spaces disks.

For additional components for other scenarios, see Scenario architecture.

Azure essentials

These are the Azure requirements for this migration scenario:

- An Azure subscription.
- An Azure premium storage account to store replicated data.
- An Azure virtual network to which VMs will connect when they're created at failover. The Azure virtual network must be in the same region as the one in which Site Recovery runs.
- An Azure standard storage account to store replication logs. This can be the same storage account for the VM disks that are being migrated.

Prerequisites

- Understand the relevant migration scenario components in the preceding section.
- Plan your downtime by learning about failover in Site Recovery.

Setup and migration steps

You can use Site Recovery to migrate Azure IaaS VMs between regions or within same region. The following instructions are tailored for this migration scenario from the article Replicate VMware VMs or physical servers to Azure. Please follow the links for detailed steps in addition to the instructions in this article.

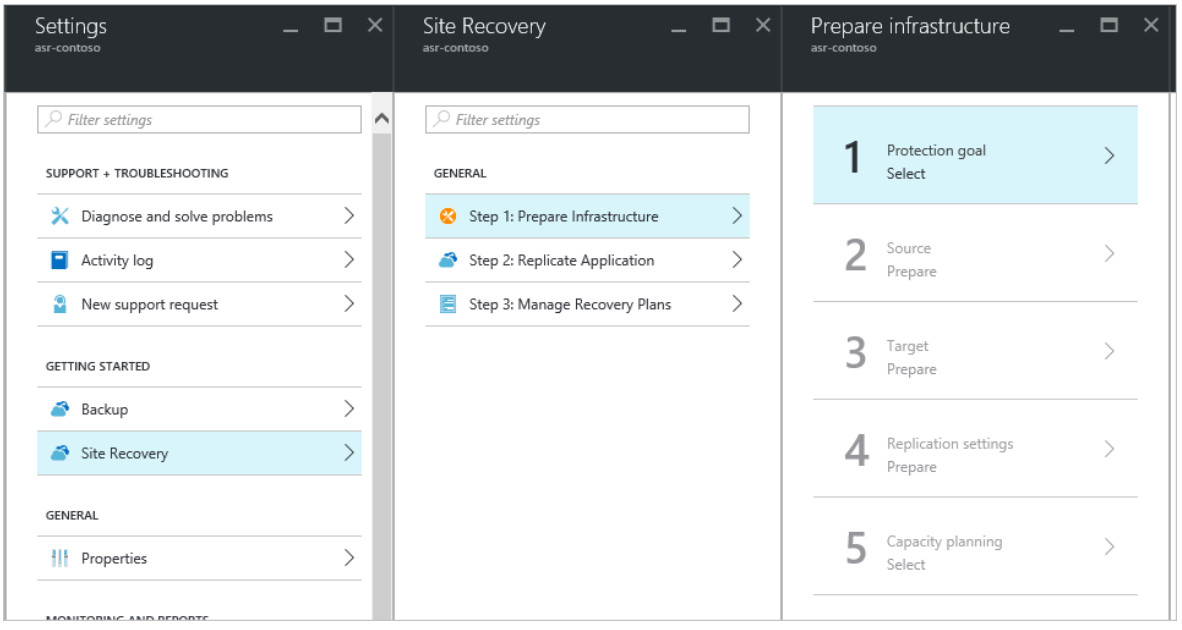
Step 1: Create a Recovery Services vault

1. Open the Azure portal.
2. Select **Create a resource** > **Management** > **Backup** and **Site Recovery (OMS)**. Alternatively, you can select **Browse** > **Recovery Services Vault** > **Add**.
3. Specify a region that VMs will be replicated to. For the purpose of migration in the same region, select the region where your source VMs and source storage accounts are.

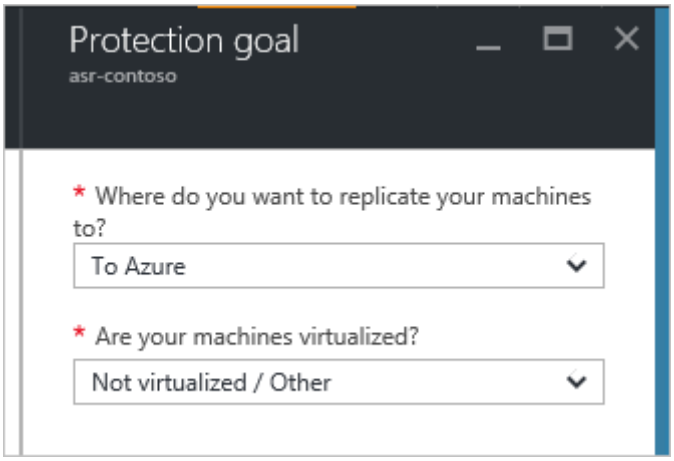
Step 2: Choose your protection goals

1. On the VM where you want to install the configuration server, open the Azure portal.

2. Go to **Recovery Services vaults** > **Settings** > **Site Recovery** > **Step 1: Prepare Infrastructure** > **Protection goal**.



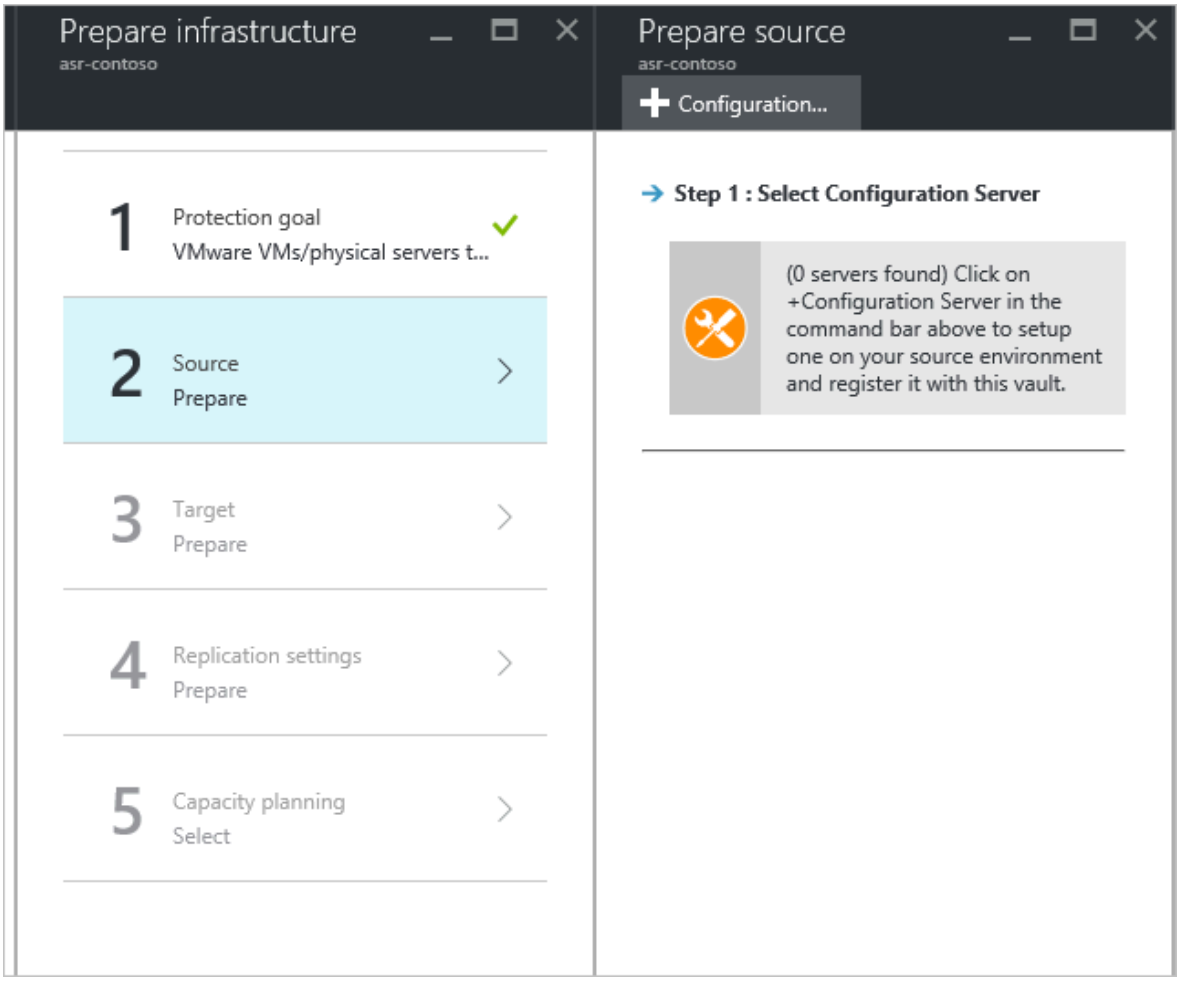
3. Under **Protection goal**, in the first drop-down list, select **To Azure**. In the second drop-down list, select **Not virtualized / Other**, and then select **OK**.



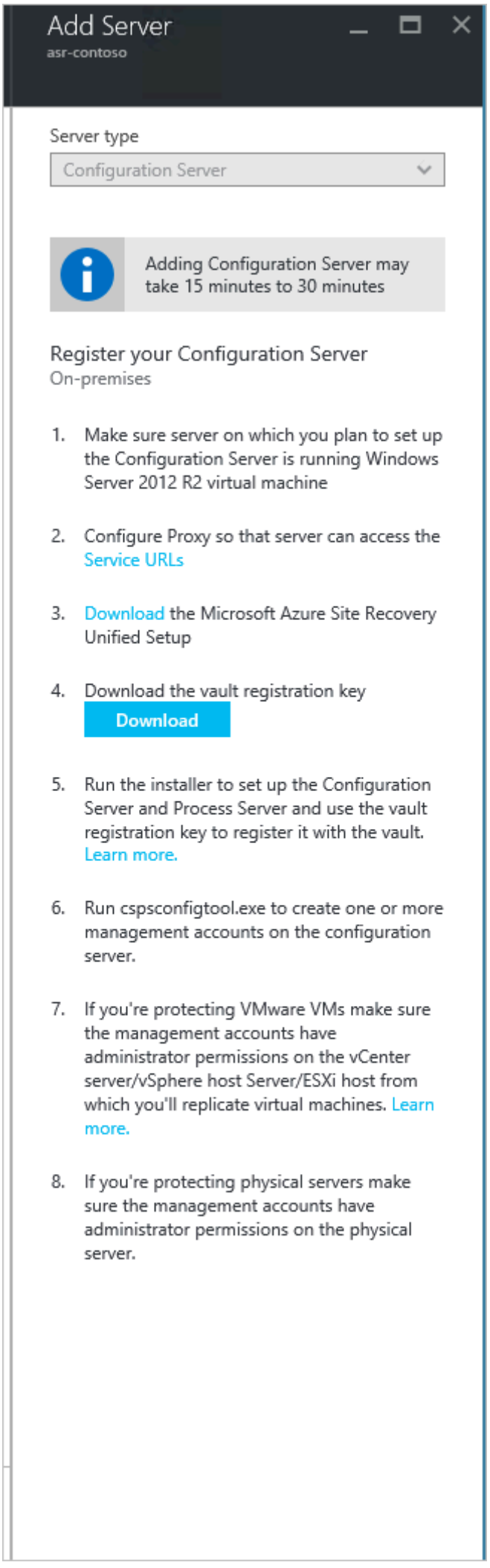
Step 3: Set up the source environment (configuration server)

1. Download **Azure Site Recovery Unified Setup** and the vault registration key by going to the **Prepare infrastructure** > **Prepare source** > **Add Server** panes.

   You will need the vault registration key to run the unified setup. The key is valid for five days after you generate it.

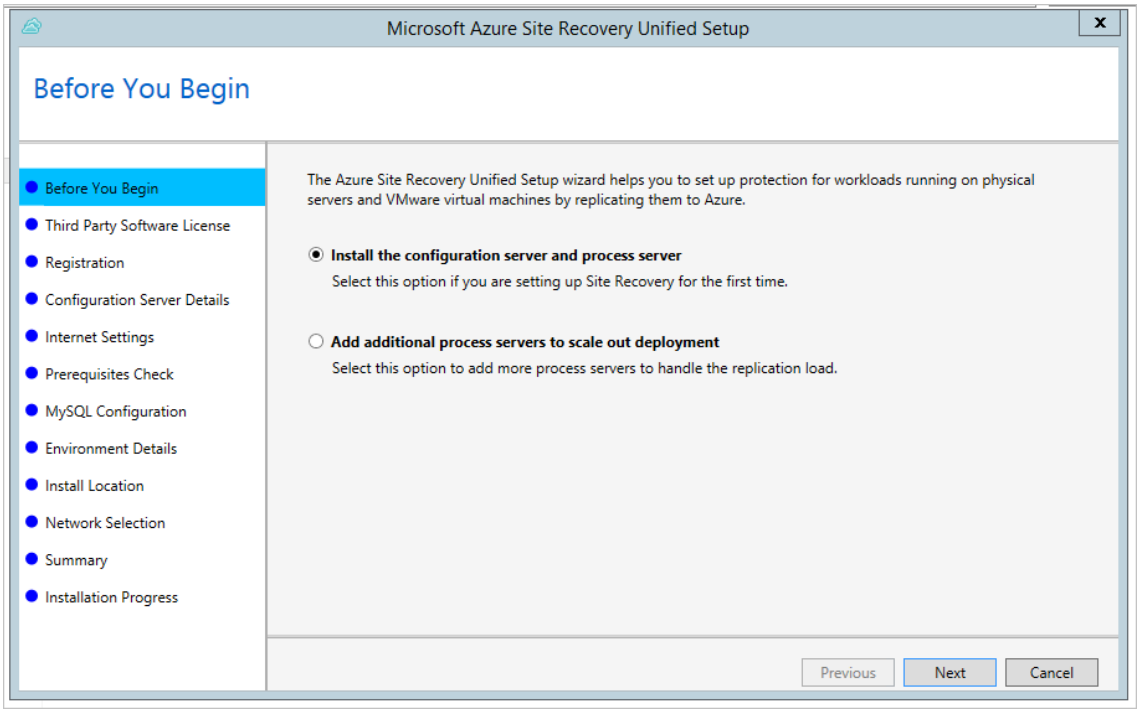


2. In the **Add Server** pane, add a configuration server.

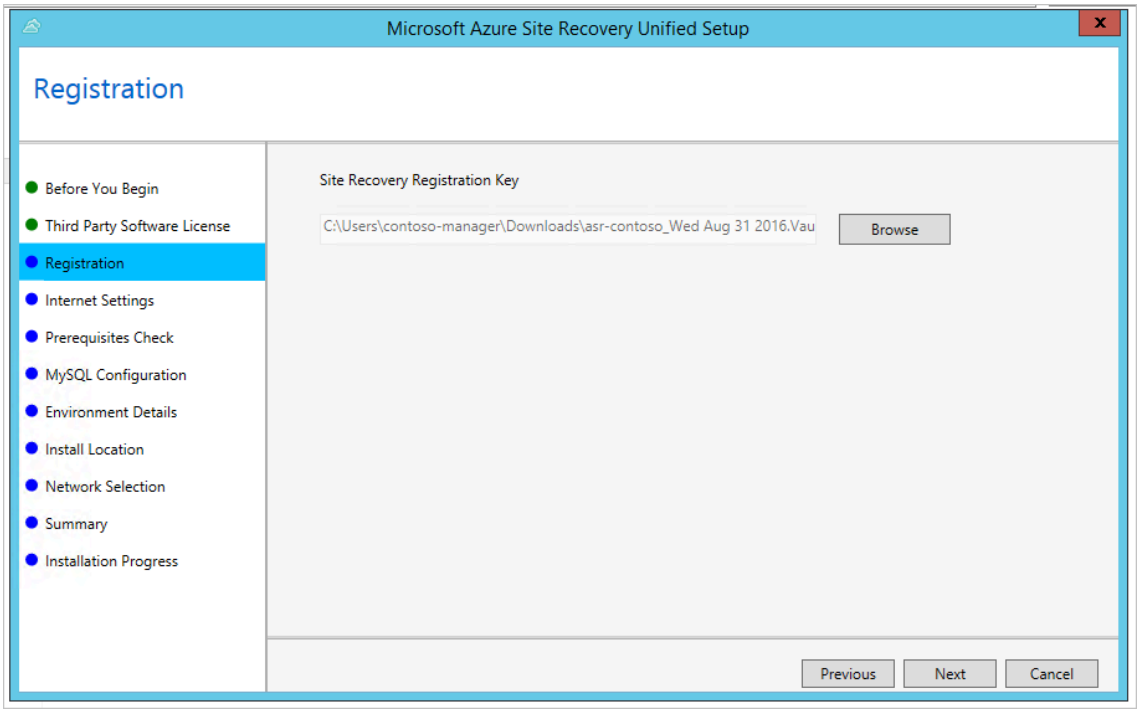

Add Server
asr-contoso
Server type
Configuration Server
Adding Configuration Server may take 15 minutes to 30 minutes
Register your Configuration Server
On-premises
1. Make sure server on which you plan to set up the Configuration Server is running Windows Server 2012 R2 virtual machine
2. Configure Proxy so that server can access the Service URLs
3. Download the Microsoft Azure Site Recovery Unified Setup
4. Download the vault registration key
Download
5. Run the installer to set up the Configuration Server and Process Server and use the vault registration key to register it with the vault. Learn more.
6. Run cspsconfigtool.exe to create one or more management accounts on the configuration server.
7. If you're protecting VMware VMs make sure the management accounts have administrator permissions on the vCenter server/vSphere host Server/ESXi host from which you'll replicate virtual machines. Learn more.
8. If you're protecting physical servers make sure the management accounts have administrator permissions on the physical server.

3. On the VM that you're using as the configuration server, run Unified Setup to install the configuration server and the process server. You can walk through the screenshots to complete the installation. You can refer to the following screenshots for steps specified for this migration scenario.

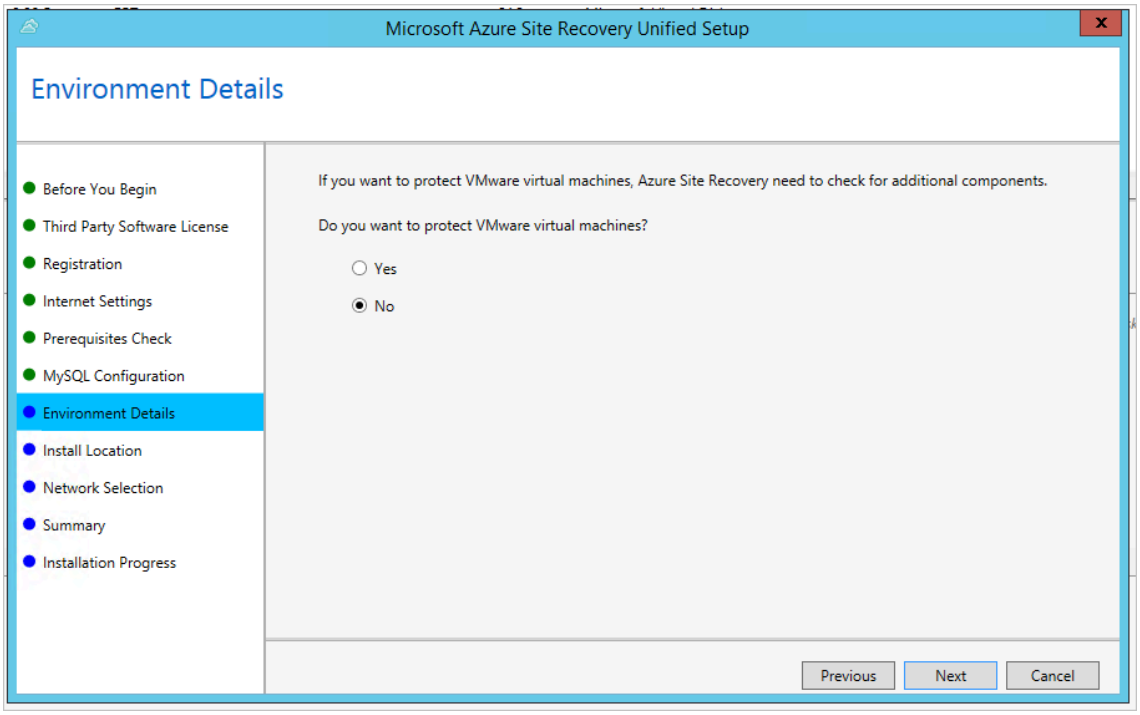
   a. In **Before You Begin**, select **Install the configuration server and process server**.



   b. In **Registration**, browse and select the registration key that you downloaded from the vault.



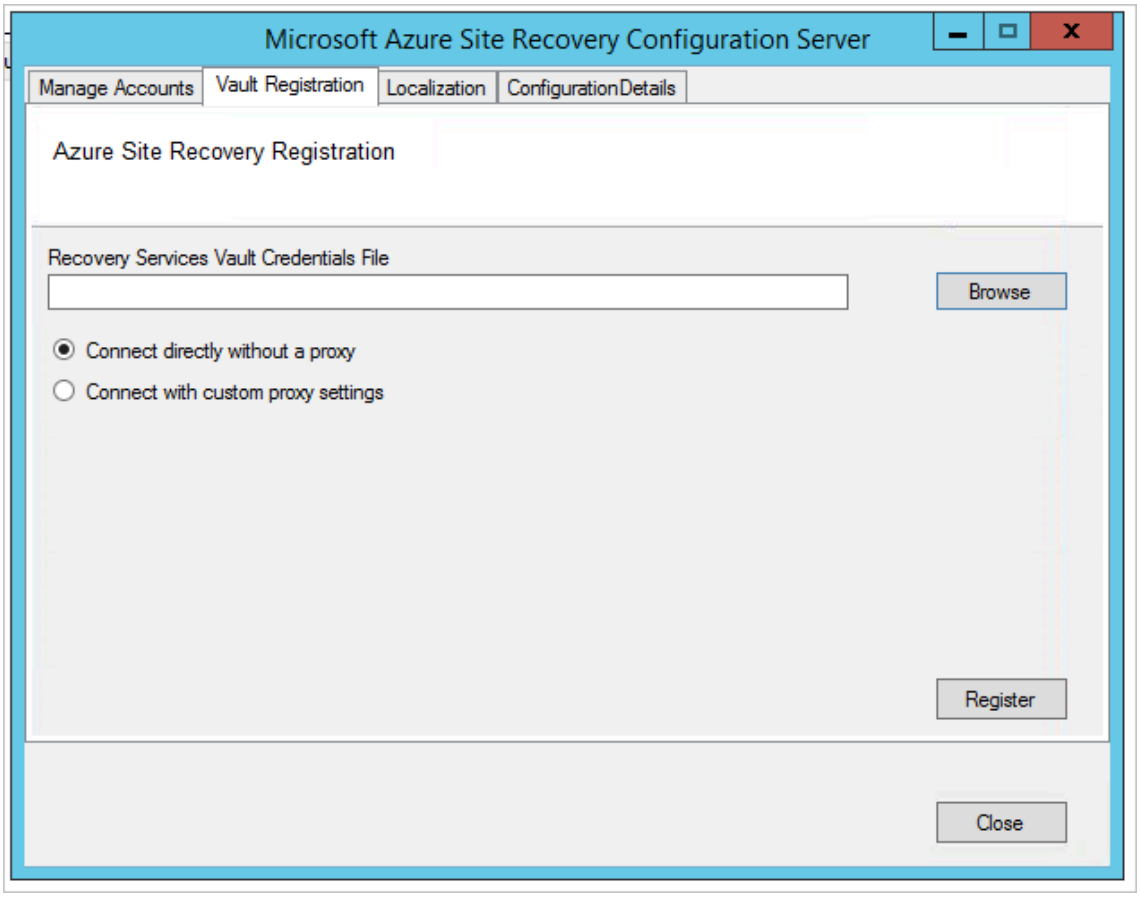
   c. In **Environment Details**, select whether you're going to replicate VMware VMs. For this migration scenario, choose **No**.



4. After the installation is complete, do the following in the **Microsoft Azure Site Recovery Configuration Server** window:

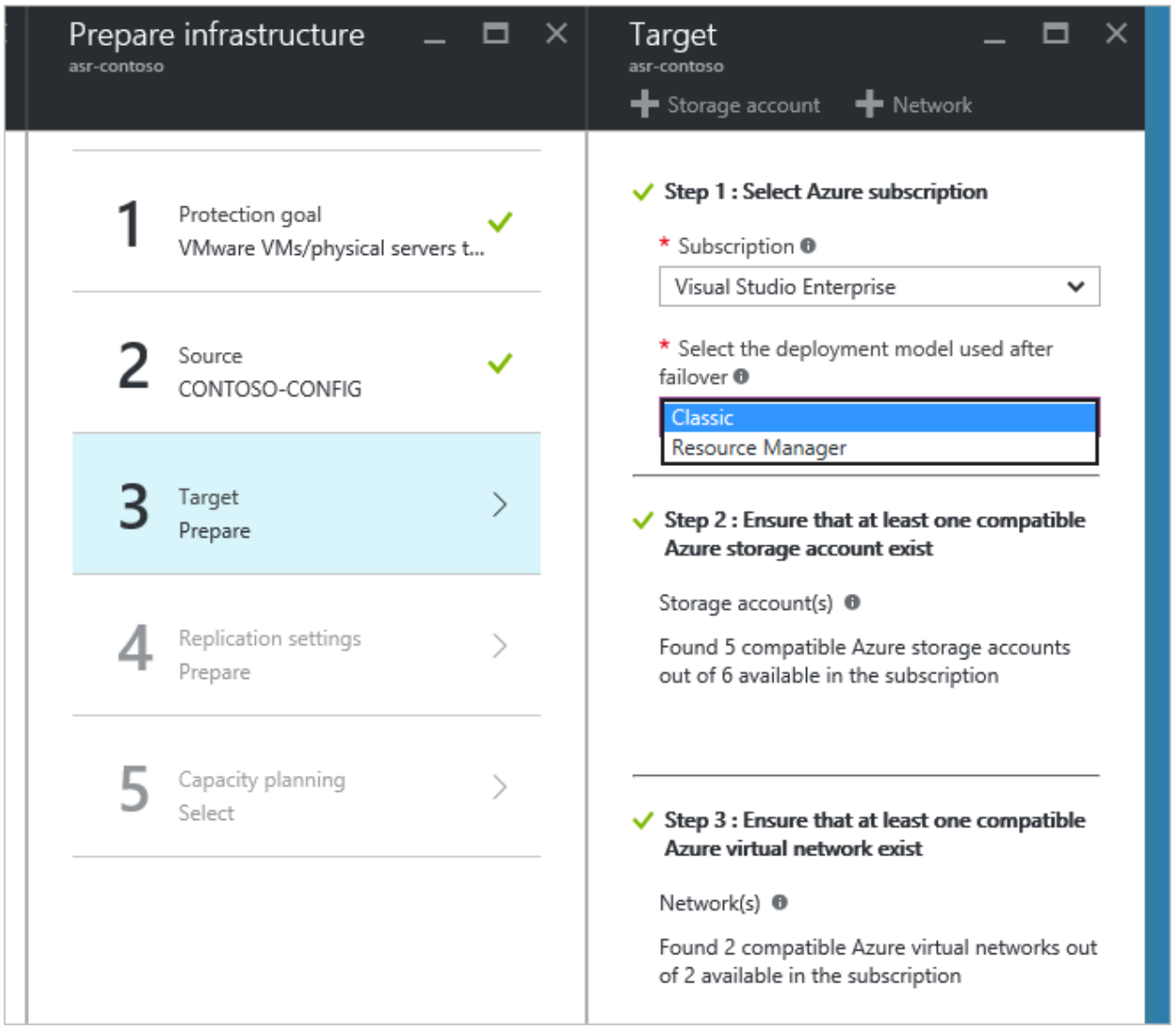
   a. Use the **Manage Accounts** tab to create the account that Site Recovery can use for automatic discovery. (In the scenario about protecting physical machines, setting up the account isn't relevant, but you need at least one account to enable one of the following steps. In this case, you can name the account and password as any.)

   b. Use the **Vault Registration** tab to upload the vault credential file.



Step 4: Set up the target environment

Select **Prepare infrastructure** > **Target**, and specify the deployment model that you want to use for VMs after failover. You can choose **Classic** or **Resource Manager**, depending on your scenario.



Site Recovery checks that you have one or more compatible Azure storage accounts and networks.

> **Note**
>
> If you're using a premium storage account for replicated data, you need to set up an additional standard storage account to store replication logs.

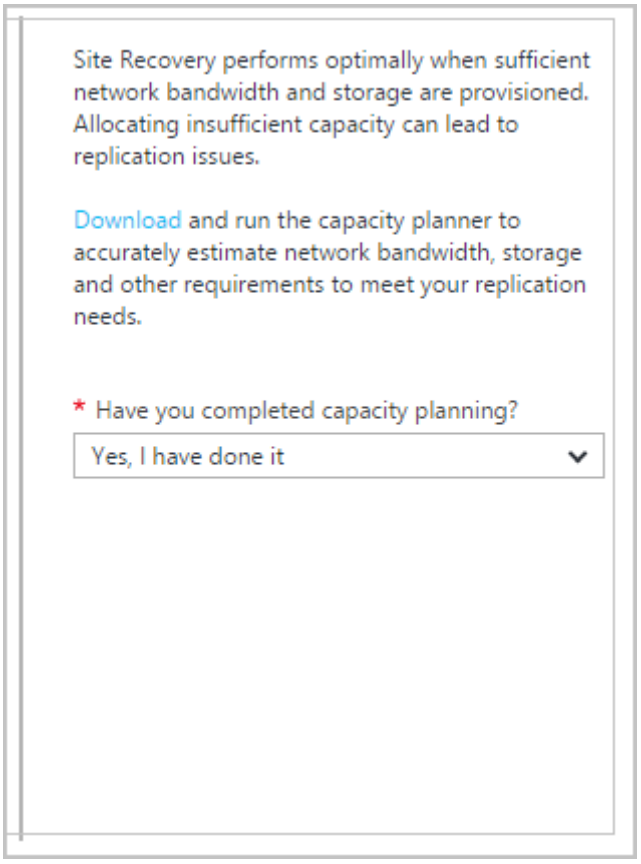
Step 5: Set up replication settings

To verify that your configuration server is successfully associated with the replication policy that you create, follow Set up replication settings.

Step 6: Plan capacity

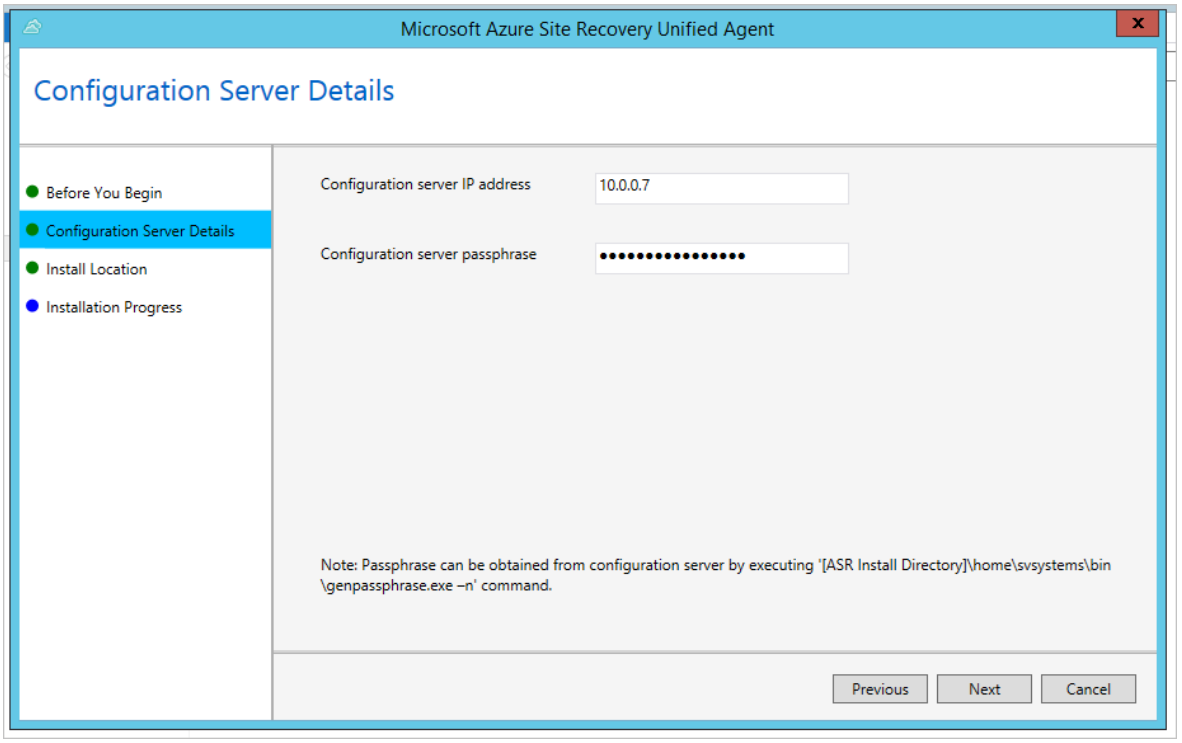
1. Use the capacity planner to accurately estimate network bandwidth, storage, and other requirements to meet your replication needs.

2. When you're done, select **Yes, I have done it** in **Have you completed capacity planning?**.



Step 7: Install the mobility service and enable replication

1. You can choose to push installation to your source VMs or to manually install the mobility service on your source VMs. You can find the requirement of pushing installation and the path of the manual installer in the provided link. If you're doing a manual installation, you might need to use an internal IP address to find the configuration server.

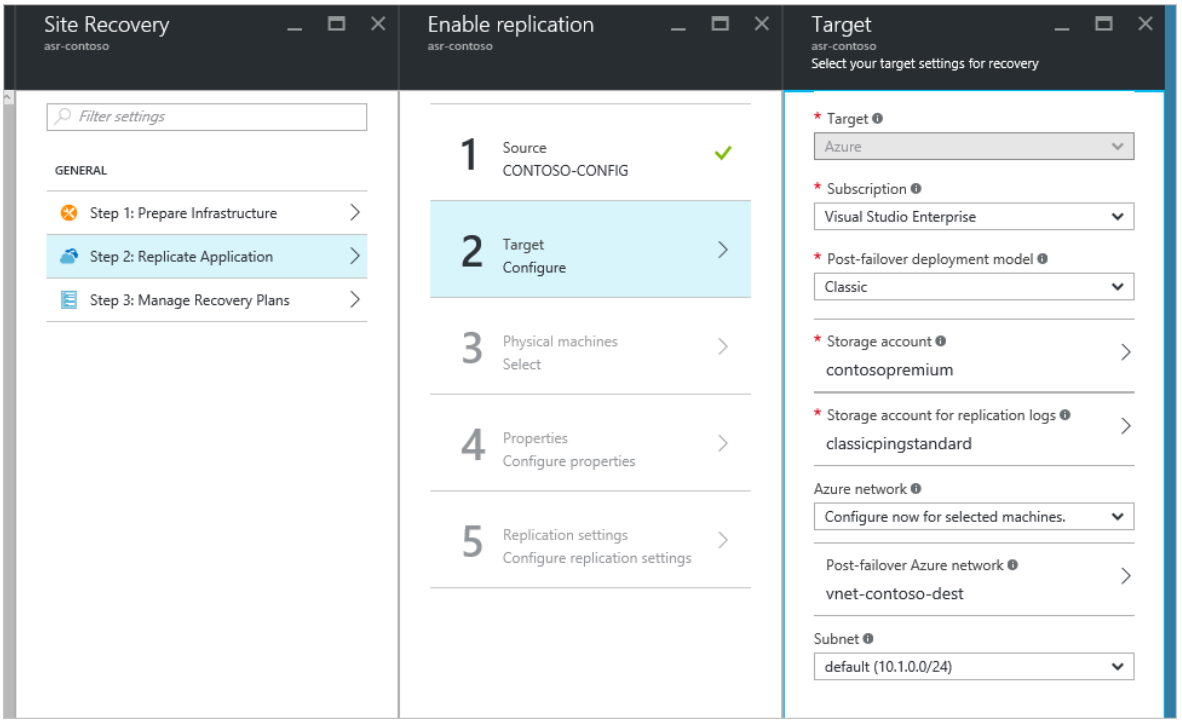


The failed-over VM will have two temporary disks: one from the primary VM and the other created during the provisioning of the VM in the recovery region. To exclude the temporary disk before replication, install the mobility service before you enable replication. To learn more about how to exclude the temporary disk, see Exclude disks from replication.

2. Enable replication as follows:
   a. Select **Replicate Application** > **Source**. After you've enabled replication for the first time, select **+Replicate** in the vault to enable replication for additional machines.
   b. In step 1, set up **Source** as your process server.
   c. In step 2, specify the post-failover deployment model, a premium storage account to migrate to, a standard storage account to save logs, and a virtual network to fail to.
   d. In step 3, add protected VMs by IP address. (You might need an internal IP address to find them.)
   e. In step 4, configure the properties by selecting the accounts that you set up previously on the process server.
   f. In step 5, choose the replication policy that you created previously in "Step 5: Set up replication settings."
   g. Select **OK**.

> **Note**
>
> When an Azure VM is deallocated and started again, there is no guarantee that it will get the same IP address. If the IP address of the configuration

server/process server or the protected Azure VMs changes, the replication in this scenario might not work correctly.



When you design your Azure Storage environment, we recommend that you use separate storage accounts for each VM in an availability set. We recommend that you follow the best practice in the storage layer to use multiple storage accounts for each availability set. Distributing VM disks to multiple storage accounts helps to improve storage availability and distributes the I/O across the Azure storage infrastructure.

If your VMs are in an availability set, instead of replicating disks of all VMs into one storage account, we highly recommend migrating multiple VMs multiple times. That way, the VMs in the same availability set do not share a single storage account. Use the **Enable Replication** pane to set up a destination storage account for each VM, one at a time.

You can choose a post-failover deployment model according to your need. If you choose Azure Resource Manager as your post-failover deployment model, you can fail over a VM (Resource Manager) to a VM (Resource Manager), or you can fail over a VM (classic) to a VM (Resource Manager).

Step 8: Run a test failover

To check whether your replication is complete, select your Site Recovery instance and then select **Settings** > **Replicated Items**. You will see the status and percentage of your replication process.

After initial replication is complete, run a test failover to validate your replication strategy. For detailed steps of a test failover, see Run a test failover in Site Recovery.

> **Note**
>
> Before you run any failover, make sure that your VMs and replication strategy meet the requirements. For more information about running a test failover, see Test failover to Azure in Site Recovery.

You can see the status of your test failover in **Settings** > **Jobs** > *YOUR_FAILOVER_PLAN_NAME*. In the pane, you can see a breakdown of the steps and success/failure results. If the test failover fails at any step, select the step to check the error message.

Step 9: Run a failover

After the test failover is completed, run a failover to migrate your disks to Premium Storage and replicate the VM instances. Follow the detailed steps in Run a failover.

Be sure to select **Shut down VMs and synchronize the latest data**. This option specifies that Site Recovery should try to shut down the protected VMs and synchronize the data so that the latest version of the data will be failed over. If you don't select this option or the attempt doesn't succeed, the failover will be from the latest available recovery point for the VM.

Site Recovery will create a VM instance whose type is the same as or similar to a Premium Storage-capable VM. You can check the performance and price of various VM instances by going to Windows Virtual Machines Pricing or Linux Virtual Machines Pricing.

Post-migration steps

1. **Configure replicated VMs to the availability set if applicable**. Site Recovery does not support migrating VMs along with the availability set. Depending on the deployment of your replicated VM, do one of the following:

    - For a VM created through the classic deployment model: Add the VM to the availability set in the Azure portal. For detailed steps, go to Add an existing virtual machine to an availability set.
    - For a VM created through the Resource Manager deployment model: Save your configuration of the VM and then delete and re-create the VMs in the

availability set.

2. **Delete old VMs and disks**. Make sure that the Premium disks are consistent with source disks and that the new VMs perform the same function as the source VMs. Delete the VM and delete the disks from your source storage accounts in the Azure portal. If there's a problem in which the disk is not deleted even though you deleted the VM, see Troubleshoot storage resource deletion errors.

3. **Clean the Azure Site Recovery infrastructure**. If Site Recovery is no longer needed, you can clean its infrastructure. Delete replicated items, the configuration server, and the recovery policy, and then delete the Azure Site Recovery vault.

Troubleshooting

- Monitor and troubleshoot protection for virtual machines and physical servers
- Microsoft Q&A question page for Microsoft Azure Site Recovery

Next steps

For specific scenarios for migrating virtual machines, see the following resources:

- Migrate Azure Virtual Machines between Storage Accounts
- Upload a Linux virtual hard disk
- Migrating Virtual Machines from Amazon AWS to Microsoft Azure

Also, see the following resources to learn more about Azure Storage and Azure Virtual Machines:

- Azure Storage
- Azure Virtual Machines
- Select a disk type for IaaS VMs

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate to Premium Storage by using Azure Site Recovery

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

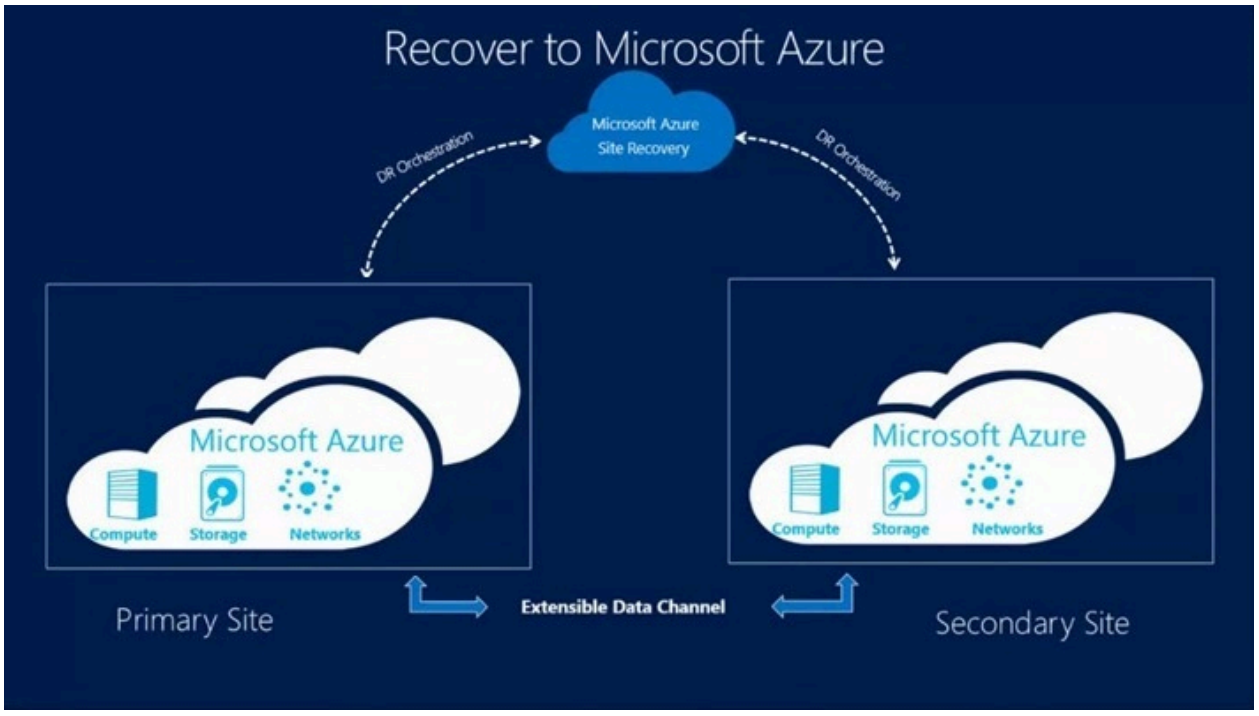
Azure premium SSDs deliver high-performance, low-latency disk support for virtual machines (VMs) that are running I/O-intensive workloads. This guide helps you migrate your VM disks from a standard storage account to a premium storage account by using Azure Site Recovery.

Site Recovery is an Azure service that contributes to your strategy for business continuity and disaster recovery by orchestrating the replication of on-premises physical servers and VMs to the cloud (Azure) or to a secondary datacenter. When outages occur in your primary location, you fail over to the secondary location to keep applications and workloads available. You fail back to your primary location when it returns to normal operation.

Site Recovery provides test failovers to support disaster recovery drills without affecting production environments. You can run failovers with minimal data loss (depending on replication frequency) for unexpected disasters. In the scenario of migrating to Premium Storage, you can use the failover in Site Recovery to migrate target disks to a premium storage account.

We recommend migrating to Premium Storage by using Site Recovery because this option provides minimal downtime. This option also avoids the manual execution of copying disks and creating new VMs. Site Recovery will systematically copy your disks and create new VMs during failover.

Site Recovery supports a number of types of failover with minimal or no downtime. To plan your downtime and estimate data loss, see the types of failover in Site Recovery. If you prepare to connect to Azure VMs after failover, you should be able to connect to the Azure VM by using RDP after failover.



Azure Site Recovery components

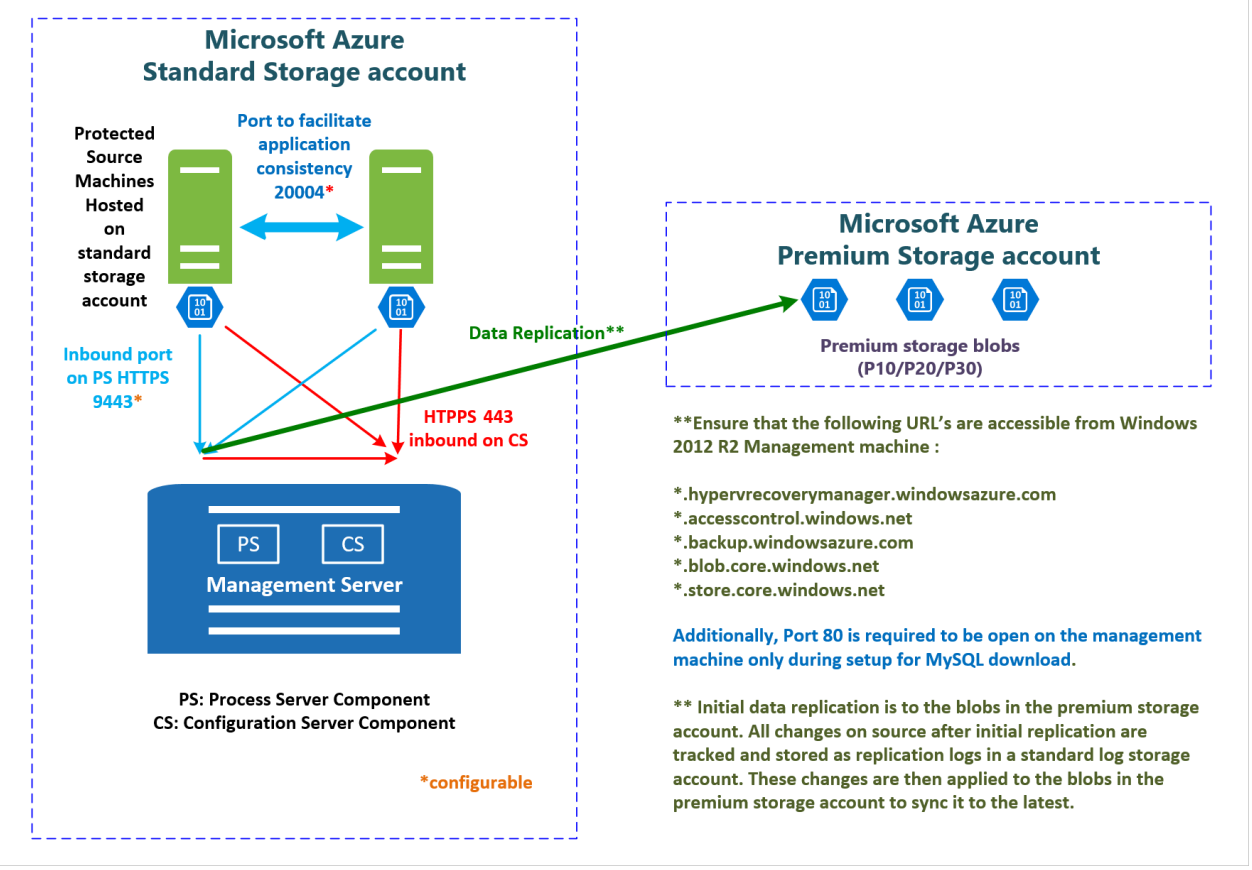
These Site Recovery components are relevant to this migration scenario:

- **Configuration server** is an Azure VM that coordinates communication and manages data replication and recovery processes. On this VM, you run a single setup file to install the configuration server and an additional component, called a process server, as a replication gateway. Read about configuration server prerequisites. You set up the configuration server only once, and you can use it for all migrations to the same region.
- **Process server** is a replication gateway that:

  1. Receives replication data from source VMs.
  2. Optimizes the data with caching, compression, and encryption.
  3. Sends the data to a storage account.

  It also handles push installation of the mobility service to source VMs and performs automatic discovery of source VMs. The default process server is installed on the configuration server. You can deploy additional standalone process servers to scale your deployment. Read about best practices for process server deployment and deploying additional process servers. You set up the process server only once, and you can use it for all migrations to the same region.
- **Mobility service** is a component that is deployed on every standard VM that you want to replicate. It captures data writes on the standard VM and forwards them to the process server. Read about replicated machine prerequisites.

This graphic shows how these components interact:



> **Note**
>
> Site Recovery does not support the migration of Storage Spaces disks.

For additional components for other scenarios, see Scenario architecture.

Azure essentials

These are the Azure requirements for this migration scenario:

- An Azure subscription.
- An Azure premium storage account to store replicated data.
- An Azure virtual network to which VMs will connect when they're created at failover. The Azure virtual network must be in the same region as the one in which Site Recovery runs.
- An Azure standard storage account to store replication logs. This can be the same storage account for the VM disks that are being migrated.

Prerequisites

- Understand the relevant migration scenario components in the preceding section.
- Plan your downtime by learning about failover in Site Recovery.

Setup and migration steps

You can use Site Recovery to migrate Azure IaaS VMs between regions or within same region. The following instructions are tailored for this migration scenario from the article Replicate VMware VMs or physical servers to Azure. Please follow the links for detailed steps in addition to the instructions in this article.

Step 1: Create a Recovery Services vault

1. Open the Azure portal.
2. Select **Create a resource** > **Management** > **Backup and Site Recovery (OMS)**. Alternatively, you can select **Browse** > **Recovery Services Vault** > **Add**.

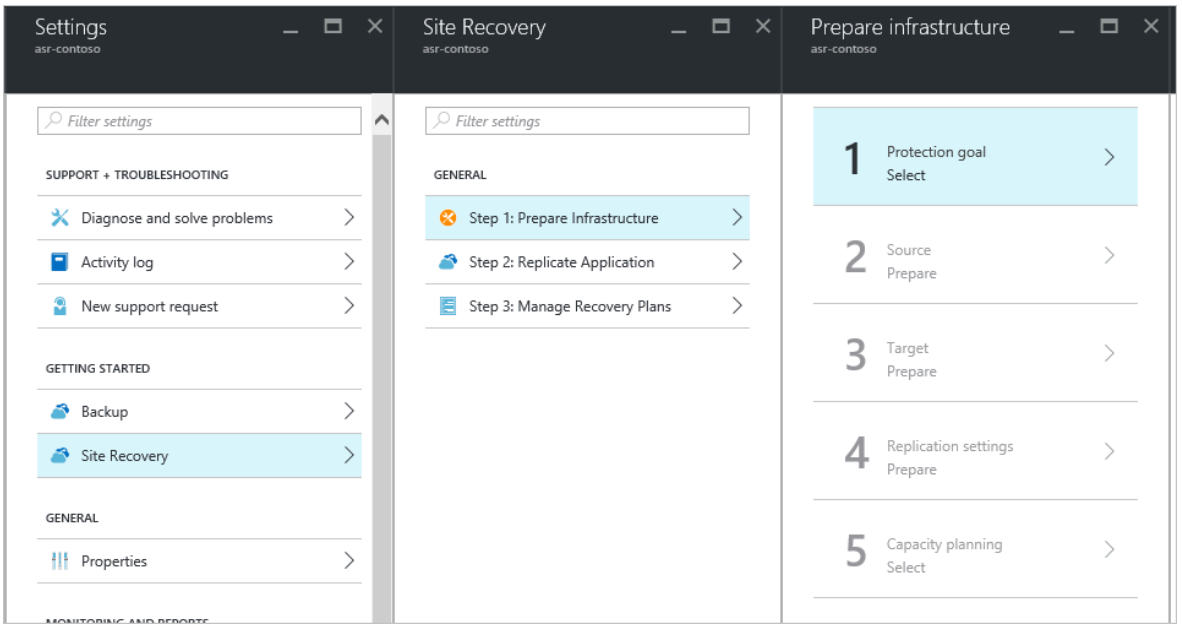
   > **Note**
   >
   > Backup and Site Recovery was formerly part of the OMS suite.

3. Specify a region that VMs will be replicated to. For the purpose of migration in the same region, select the region where your source VMs and source storage accounts are.

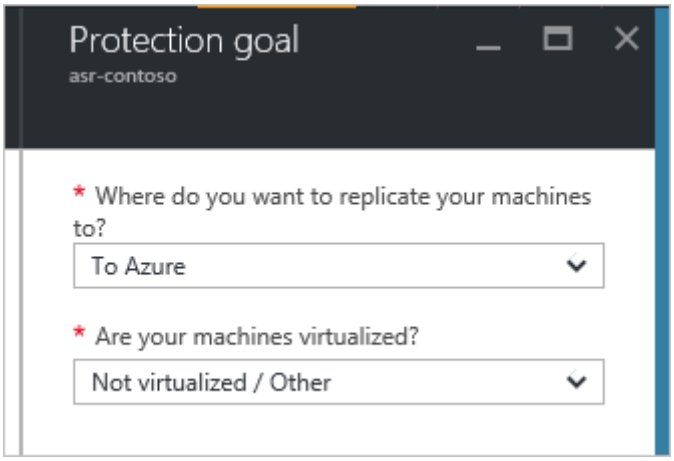
Step 2: Choose your protection goals

1. On the VM where you want to install the configuration server, open the Azure portal.

2. Go to **Recovery Services vaults** > **Settings** > **Site Recovery** > **Step 1: Prepare Infrastructure** > **Protection goal**.



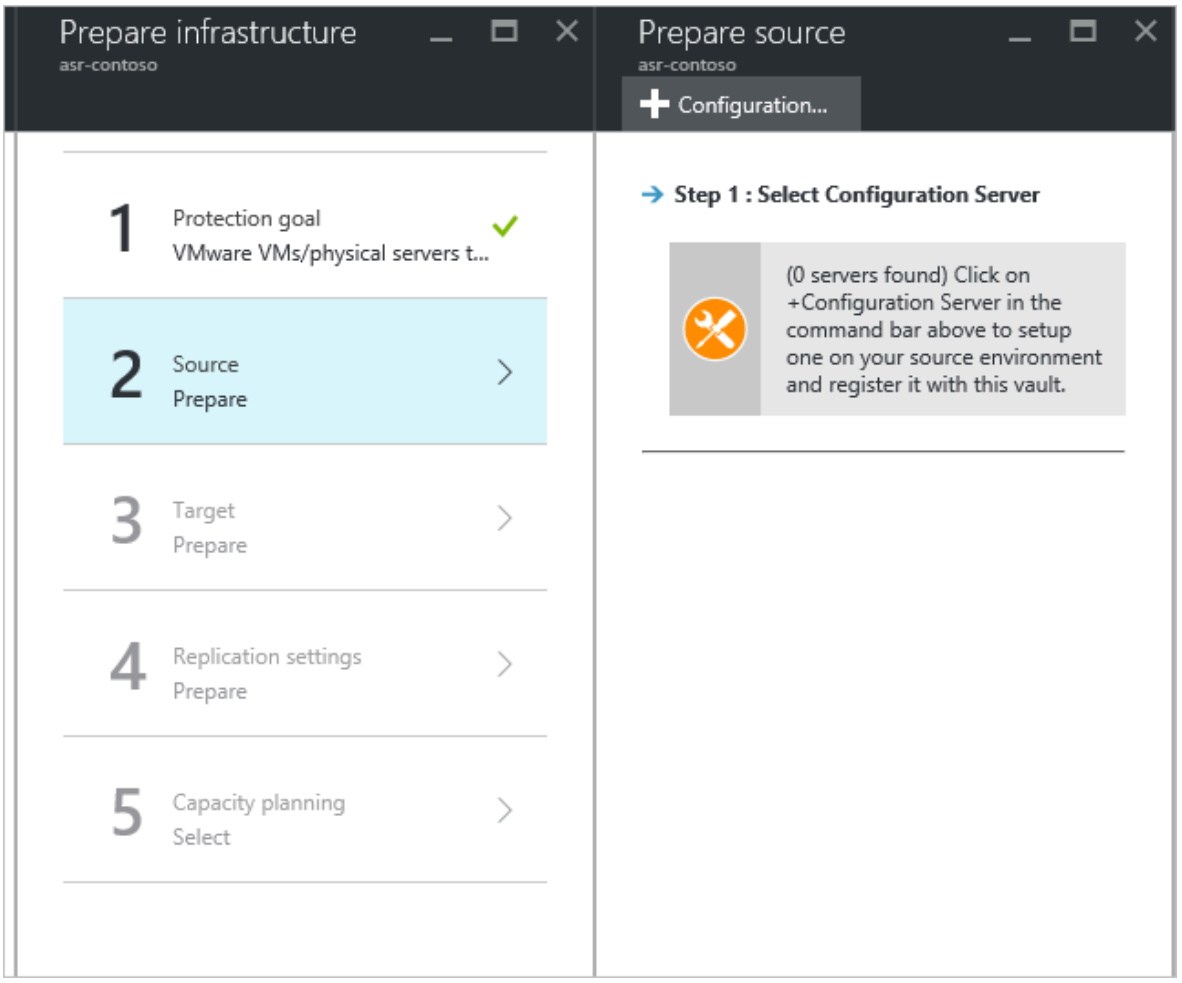
3. Under **Protection goal**, in the first drop-down list, select **To Azure**. In the second drop-down list, select **Not virtualized / Other**, and then select **OK**.



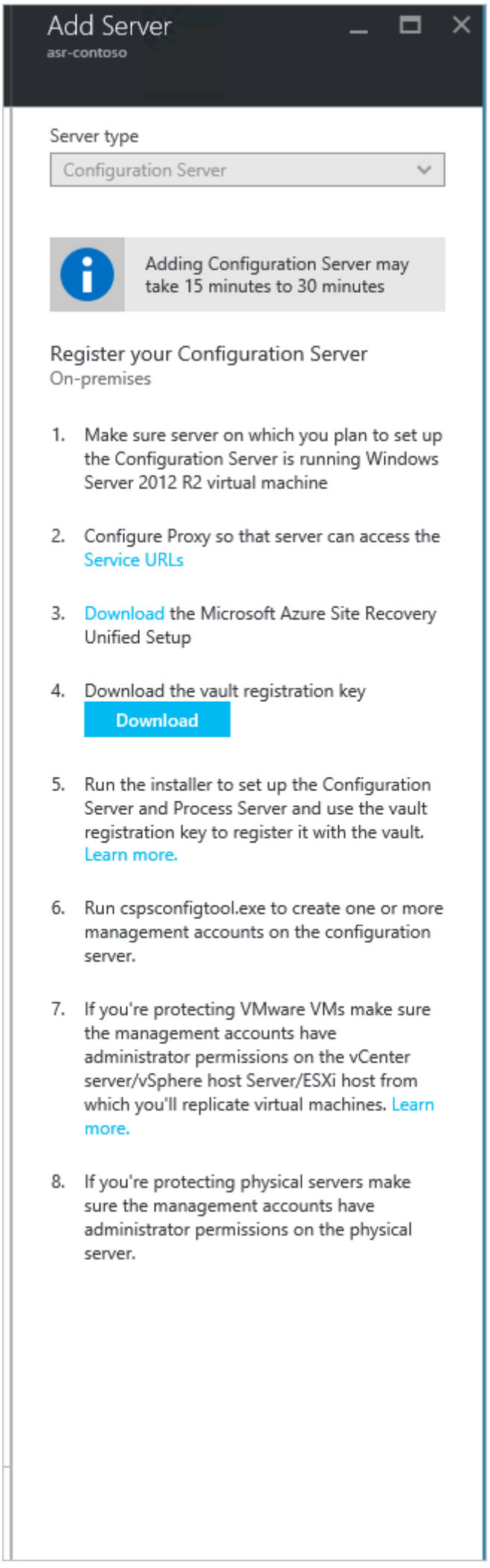
Step 3: Set up the source environment (configuration server)

1. Download **Azure Site Recovery Unified Setup** and the vault registration key by going to the **Prepare infrastructure** > **Prepare source** > **Add Server** panes.

   You will need the vault registration key to run the unified setup. The key is valid for five days after you generate it.

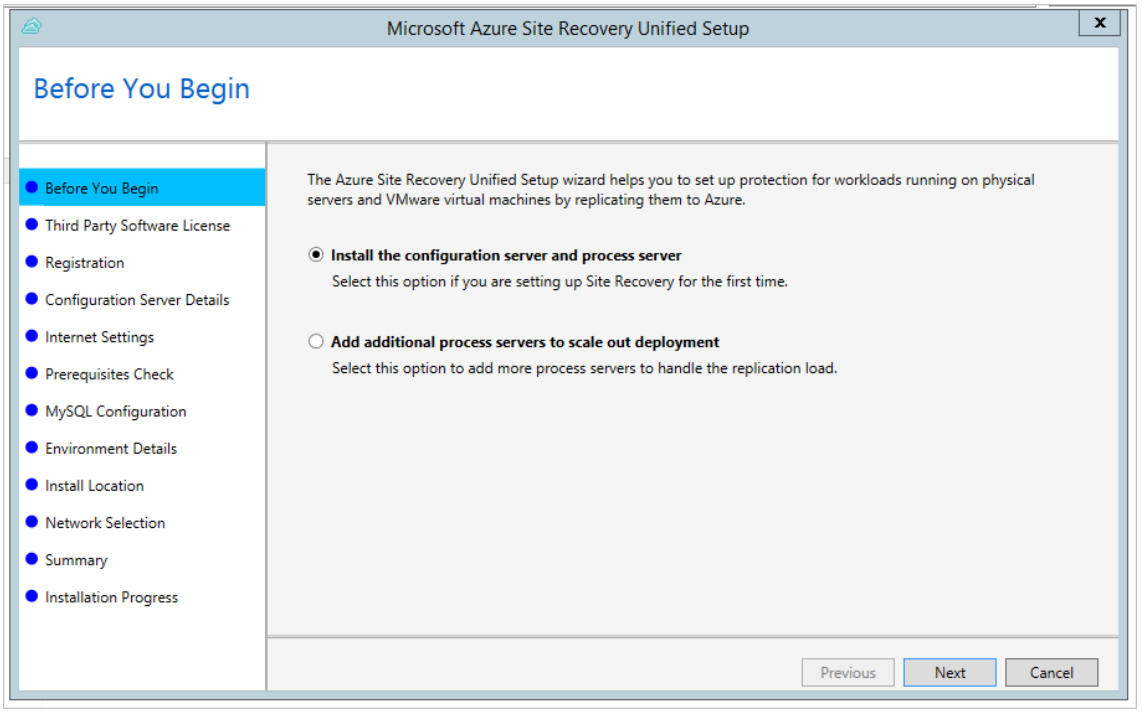


2. In the **Add Server** pane, add a configuration server.

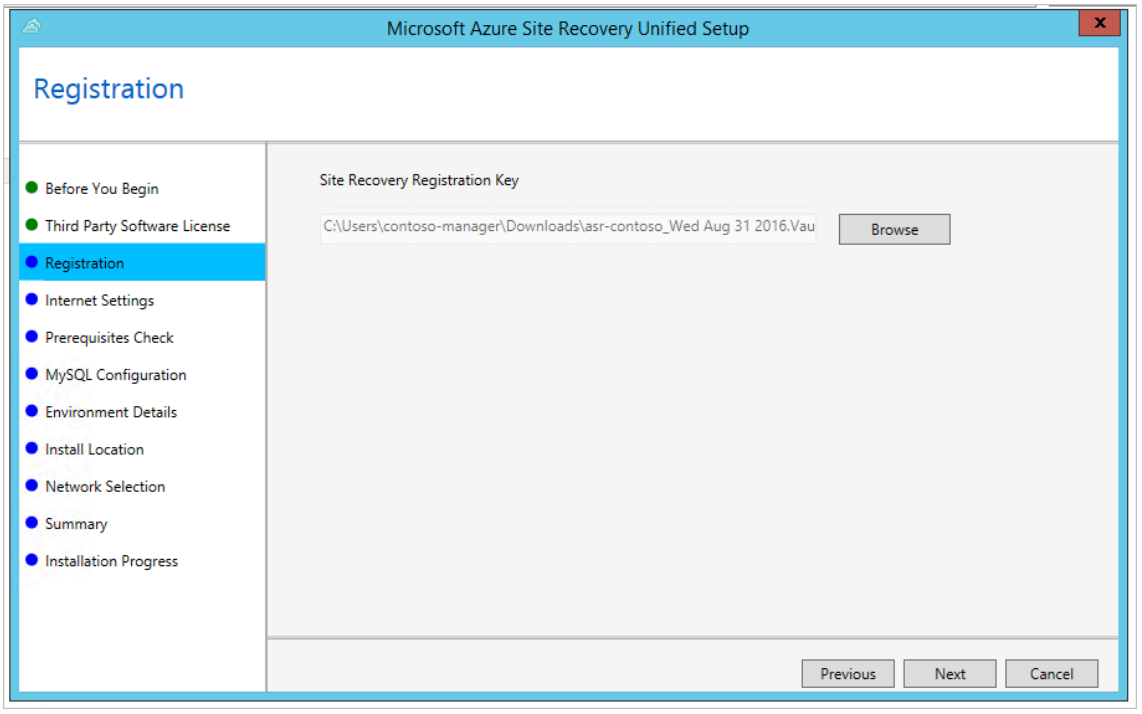

Add Server
asr-contoso
Server type
Configuration Server
Adding Configuration Server may take 15 minutes to 30 minutes
Register your Configuration Server
On-premises
1. Make sure server on which you plan to set up the Configuration Server is running Windows Server 2012 R2 virtual machine
2. Configure Proxy so that server can access the Service URLs
3. Download the Microsoft Azure Site Recovery Unified Setup
4. Download the vault registration key
Download
5. Run the installer to set up the Configuration Server and Process Server and use the vault registration key to register it with the vault. Learn more.
6. Run cspsconfigtool.exe to create one or more management accounts on the configuration server.
7. If you're protecting VMware VMs make sure the management accounts have administrator permissions on the vCenter server/vSphere host Server/ESXi host from which you'll replicate virtual machines. Learn more.
8. If you're protecting physical servers make sure the management accounts have administrator permissions on the physical server.

3. On the VM that you're using as the configuration server, run Unified Setup to install the configuration server and the process server. You can walk through the screenshots to complete the installation. You can refer to the following screenshots for steps specified for this migration scenario.

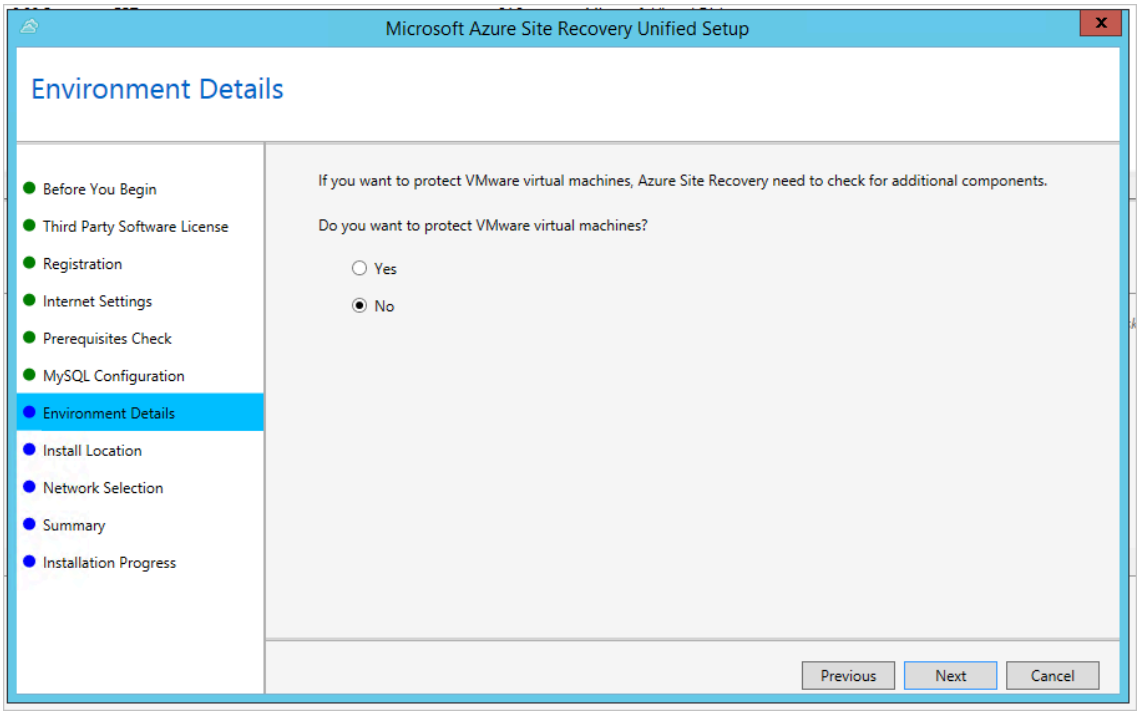
   a. In **Before You Begin**, select **Install the configuration server and process server**.



   b. In **Registration**, browse and select the registration key that you downloaded from the vault.



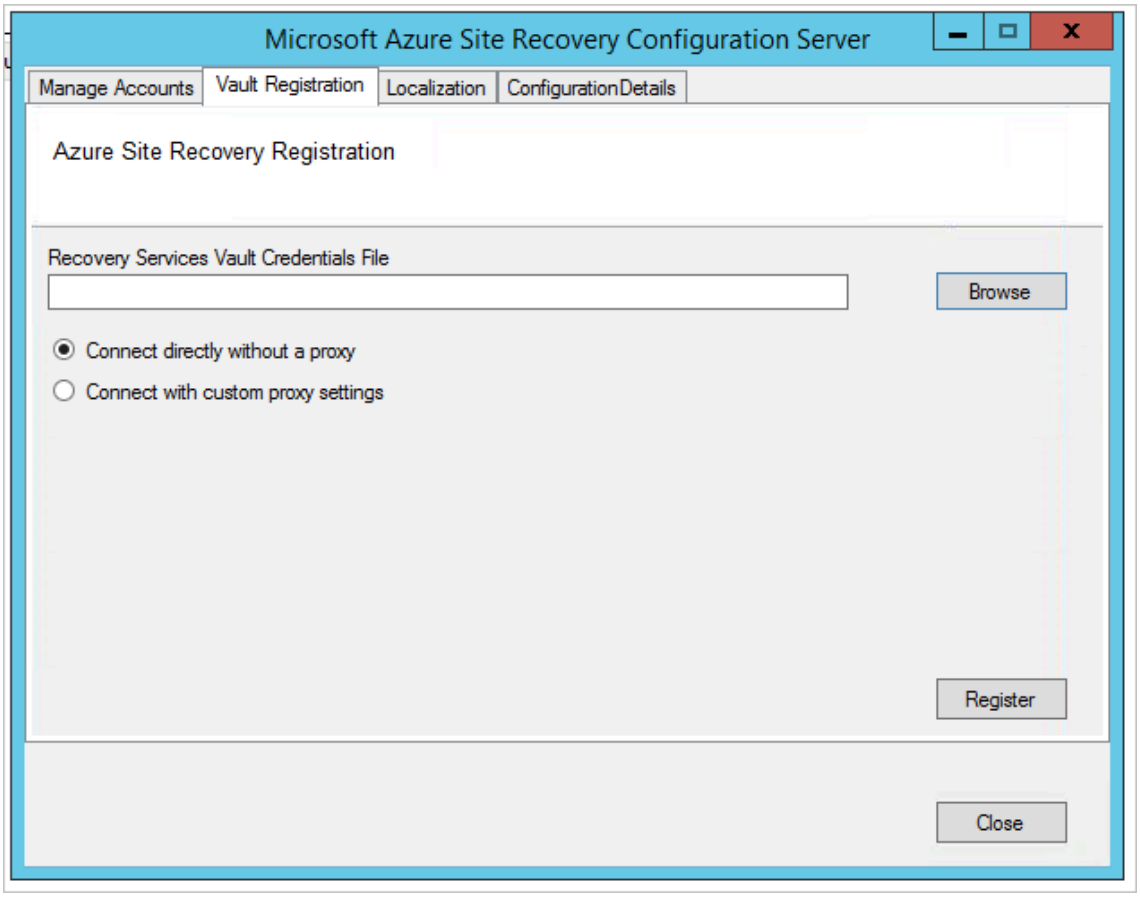
   c. In **Environment Details**, select whether you're going to replicate VMware VMs. For this migration scenario, choose **No**.



4. After the installation is complete, do the following in the **Microsoft Azure Site Recovery Configuration Server** window:

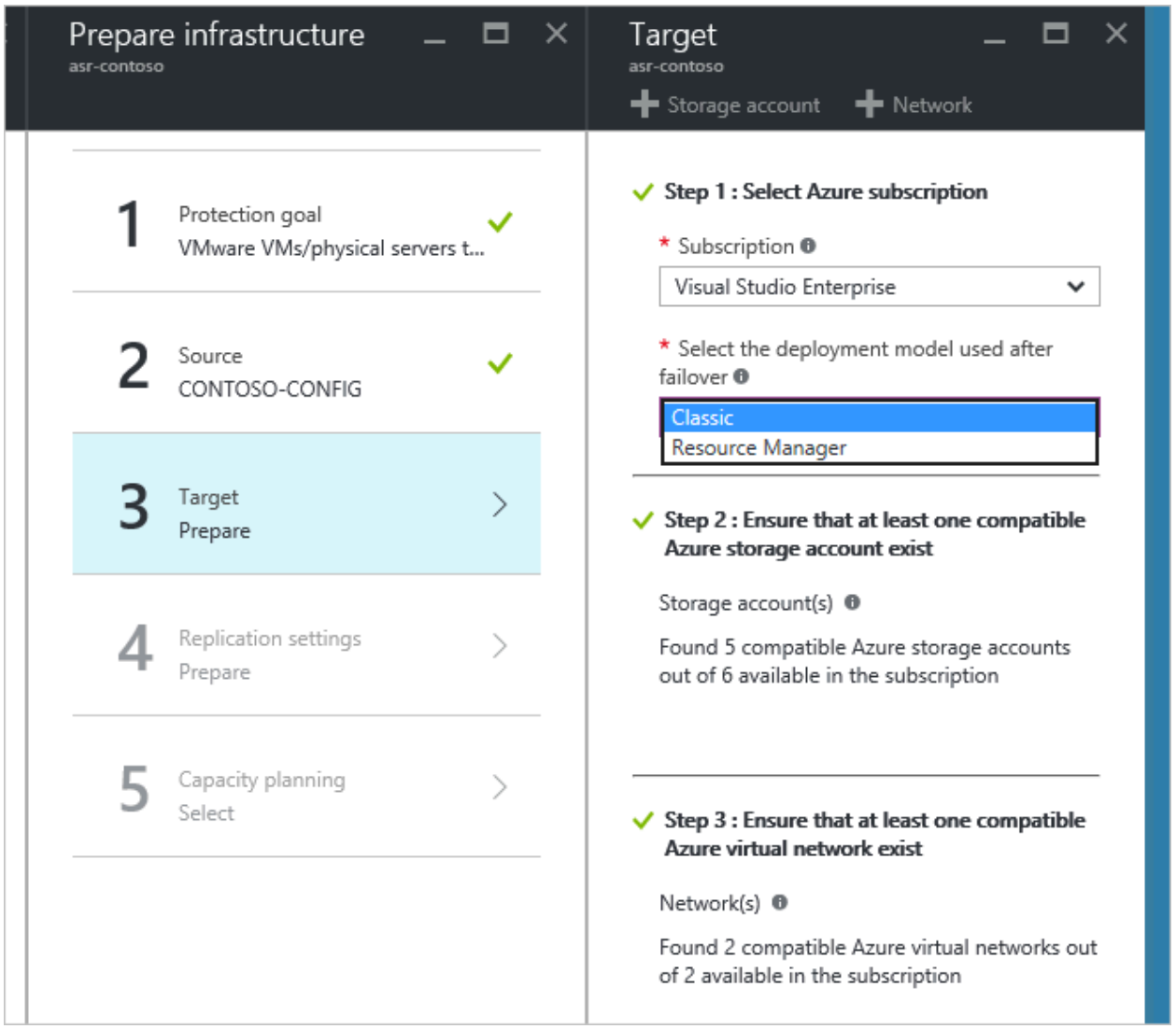
   a. Use the **Manage Accounts** tab to create the account that Site Recovery can use for automatic discovery. (In the scenario about protecting physical machines, setting up the account isn't relevant, but you need at least one account to enable one of the following steps. In this case, you can name the account and password as any.)

   b. Use the **Vault Registration** tab to upload the vault credential file.



Step 4: Set up the target environment

Select **Prepare infrastructure** > **Target**, and specify the deployment model that you want to use for VMs after failover. You can choose **Classic** or **Resource Manager**, depending on your scenario.



Site Recovery checks that you have one or more compatible Azure storage accounts and networks.

> **Note**
>
> If you're using a premium storage account for replicated data, you need to set up an additional standard storage account to store replication logs.

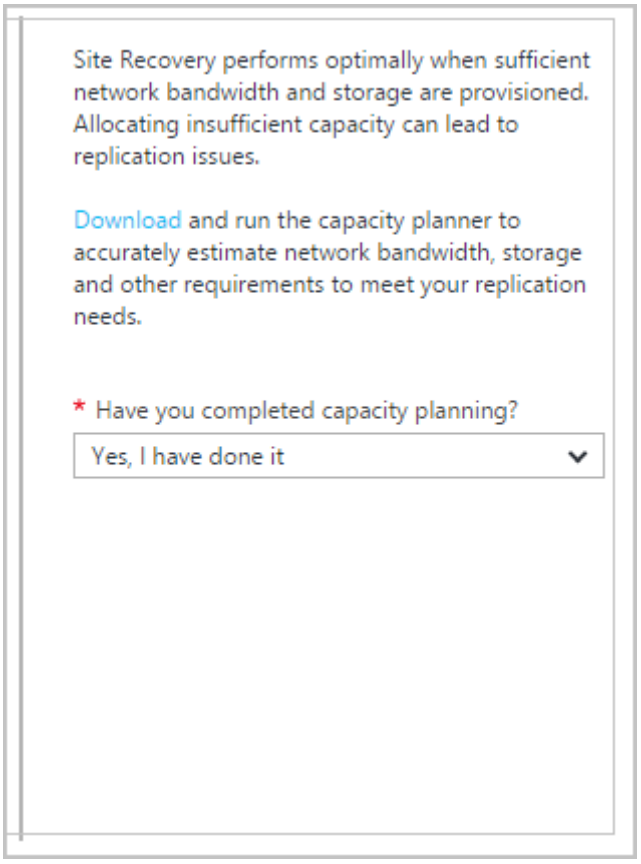
Step 5: Set up replication settings

To verify that your configuration server is successfully associated with the replication policy that you create, follow Set up replication settings.

Step 6: Plan capacity

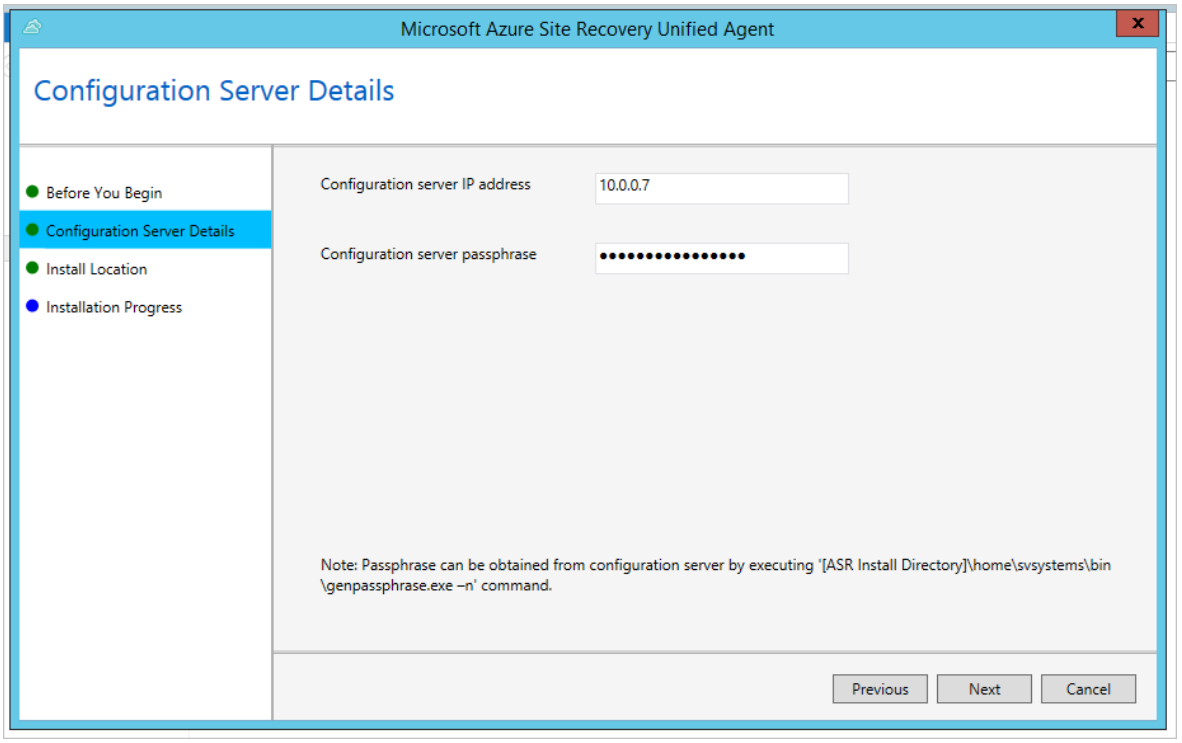
1. Use the capacity planner to accurately estimate network bandwidth, storage, and other requirements to meet your replication needs.

2. When you're done, select **Yes, I have done it** in **Have you completed capacity planning?**.



Step 7: Install the mobility service and enable replication

1. You can choose to push installation to your source VMs or to manually install the mobility service on your source VMs. You can find the requirement of pushing installation and the path of the manual installer in the provided link. If you're doing a manual installation, you might need to use an internal IP address to find the configuration server.

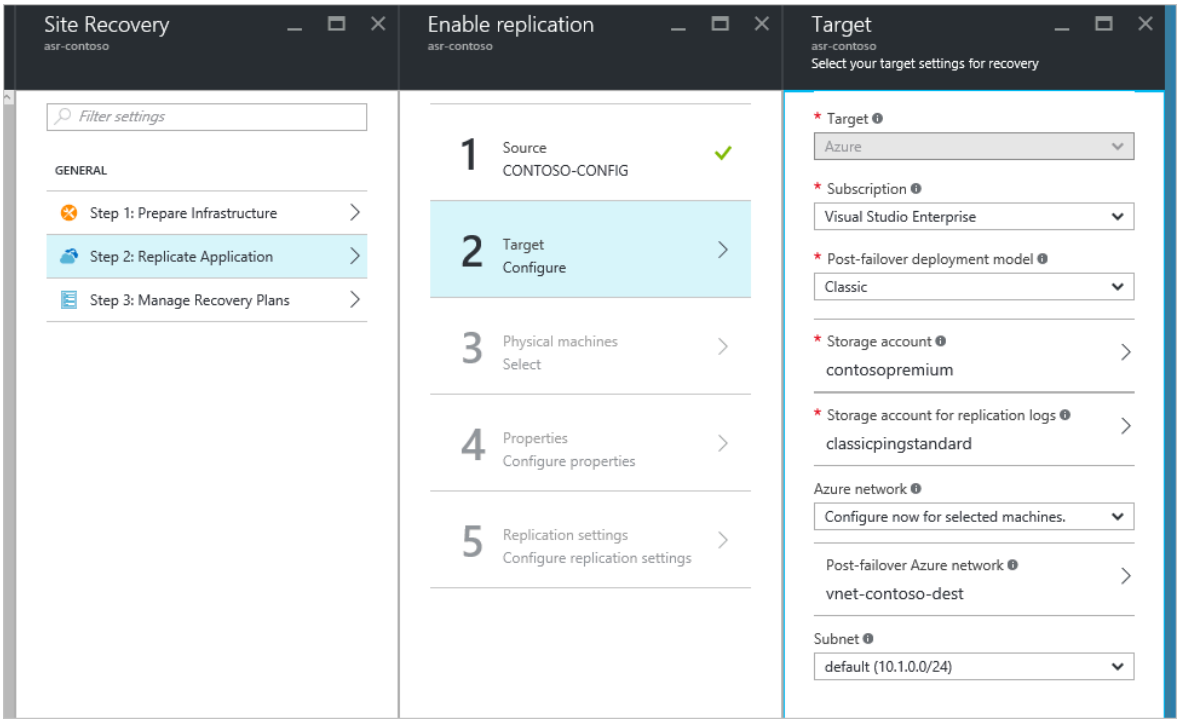


The failed-over VM will have two temporary disks: one from the primary VM and the other created during the provisioning of the VM in the recovery region. To exclude the temporary disk before replication, install the mobility service before you enable replication. To learn more about how to exclude the temporary disk, see Exclude disks from replication.

2. Enable replication as follows:
   a. Select **Replicate Application** > **Source**. After you've enabled replication for the first time, select **+Replicate** in the vault to enable replication for additional machines.
   b. In step 1, set up **Source** as your process server.
   c. In step 2, specify the post-failover deployment model, a premium storage account to migrate to, a standard storage account to save logs, and a virtual network to fail to.
   d. In step 3, add protected VMs by IP address. (You might need an internal IP address to find them.)
   e. In step 4, configure the properties by selecting the accounts that you set up previously on the process server.
   f. In step 5, choose the replication policy that you created previously in "Step 5: Set up replication settings."
   g. Select **OK**.

> **Note**
>
> When an Azure VM is deallocated and started again, there is no guarantee that it will get the same IP address. If the IP address of the configuration

server/process server or the protected Azure VMs changes, the replication in this scenario might not work correctly.



When you design your Azure Storage environment, we recommend that you use separate storage accounts for each VM in an availability set. We recommend that you follow the best practice in the storage layer to use multiple storage accounts for each availability set. Distributing VM disks to multiple storage accounts helps to improve storage availability and distributes the I/O across the Azure storage infrastructure.

If your VMs are in an availability set, instead of replicating disks of all VMs into one storage account, we highly recommend migrating multiple VMs multiple times. That way, the VMs in the same availability set do not share a single storage account. Use the **Enable Replication** pane to set up a destination storage account for each VM, one at a time.

You can choose a post-failover deployment model according to your need. If you choose Azure Resource Manager as your post-failover deployment model, you can fail over a VM (Resource Manager) to a VM (Resource Manager), or you can fail over a VM (classic) to a VM (Resource Manager).

Step 8: Run a test failover

To check whether your replication is complete, select your Site Recovery instance and then select **Settings** > **Replicated Items**. You will see the status and percentage of your replication process.

After initial replication is complete, run a test failover to validate your replication strategy. For detailed steps of a test failover, see Run a test failover in Site Recovery.

> **Note**
>
> Before you run any failover, make sure that your VMs and replication strategy meet the requirements. For more information about running a test failover, see Test failover to Azure in Site Recovery.

You can see the status of your test failover in **Settings** > **Jobs** > *YOUR_FAILOVER_PLAN_NAME*. In the pane, you can see a breakdown of the steps and success/failure results. If the test failover fails at any step, select the step to check the error message.

Step 9: Run a failover

After the test failover is completed, run a failover to migrate your disks to Premium Storage and replicate the VM instances. Follow the detailed steps in Run a failover.

Be sure to select **Shut down VMs and synchronize the latest data**. This option specifies that Site Recovery should try to shut down the protected VMs and synchronize the data so that the latest version of the data will be failed over. If you don't select this option or the attempt doesn't succeed, the failover will be from the latest available recovery point for the VM.

Site Recovery will create a VM instance whose type is the same as or similar to a Premium Storage-capable VM. You can check the performance and price of various VM instances by going to Windows Virtual Machines Pricing or Linux Virtual Machines Pricing.

Post-migration steps

1. **Configure replicated VMs to the availability set if applicable**. Site Recovery does not support migrating VMs along with the availability set. Depending on the deployment of your replicated VM, do one of the following:

    - For a VM created through the classic deployment model: Add the VM to the availability set in the Azure portal. For detailed steps, go to Add an existing virtual machine to an availability set.
    - For a VM created through the Resource Manager deployment model: Save your configuration of the VM and then delete and re-create the VMs in the

availability set.

2. **Delete old VMs and disks**. Make sure that the Premium disks are consistent with source disks and that the new VMs perform the same function as the source VMs. Delete the VM and delete the disks from your source storage accounts in the Azure portal. If there's a problem in which the disk is not deleted even though you deleted the VM, see Troubleshoot storage resource deletion errors.

3. **Clean the Azure Site Recovery infrastructure**. If Site Recovery is no longer needed, you can clean its infrastructure. Delete replicated items, the configuration server, and the recovery policy, and then delete the Azure Site Recovery vault.

Troubleshooting

- Monitor and troubleshoot protection for virtual machines and physical servers
- Microsoft Q&A question page for Microsoft Azure Site Recovery

Next steps

For specific scenarios for migrating virtual machines, see the following resources:

- Migrate Azure Virtual Machines between Storage Accounts
- Create and upload a Windows Server VHD to Azure
- Migrating Virtual Machines from Amazon AWS to Microsoft Azure

Also, see the following resources to learn more about Azure Storage and Azure Virtual Machines:

- Azure Storage
- Azure Virtual Machines

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate Azure VMs to managed disks in Azure

Applies to: ✔ Linux VMs ✔ Windows VMs

Unmanaged disks are no longer supported. If you're using unmanaged disks, migrate to managed disks. For details, see Migrate unmanaged disks to managed disks.

Azure managed disks simplifies your storage management by removing the need to separately manage storage accounts. Managed disks provide numerous benefits over unmanaged disks, such as high availability, better scalability, large disks, bursting, and shared disks. You can also migrate your existing Azure VMs to managed disks to benefit from better reliability of VMs in an Availability Set. It ensures that the disks of different VMs in an Availability Set are sufficiently isolated from each other to avoid single point of failures. It automatically places disks of different VMs in an Availability Set in different Storage scale units (stamps) which limits the impact of single Storage scale unit failures caused due to hardware and software failures. Based on your needs, you can choose from four types of storage options. To learn about the available disk types, see Select a disk type

Migration scenarios

You can migrate to managed disks in following scenarios:

Expand table

Scenario	Article
Convert stand alone VMs and VMs in an availability set to managed disks	Convert VMs to use managed disks
Convert a single VM from classic to Resource Manager on managed disks	Create a VM from a classic VHD
Convert all the VMs in a vNet from classic to Resource Manager on managed disks	Migrate IaaS resources from classic to Resource Manager and then Convert a VM from unmanaged disks to managed disks
Upgrade VMs with standard unmanaged disks to VMs with managed premium disks	First, Convert a Windows virtual machine from unmanaged disks to managed disks. Then Update the storage type of a managed disk.

> **Important**
>
> VMs created through the classic deployment model will be retired on September 1, 2023.
>
> If you use IaaS resources from Azure Service Management, please complete your migration by September 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.
>
> For more information, see Migrate your IaaS resources to Azure Resource Manager by September 1, 2023.

Next steps

- Learn more about managed disks
- Review the pricing for managed disks.

Last updated on 02/26/2026

Migrate unmanaged disks to managed disks

Applies to: ✔ Linux VMs ✔ Windows VMs

In 2017, we launched Azure managed disks. We've been enhancing capabilities ever since. Because Azure managed disks now have the full capabilities of unmanaged disks and other advancements, we announced the retirement for unmanaged disks on September 13, 2022. Unmanaged disks were fully retired on March 31, 2026 (extended from previously published date of September 30, 2025). If you're still using unmanaged disks, migrate to managed disks as soon as possible.

By using managed disks, you don't need to manage storage accounts for creating a disk, because Azure manages the storage accounts under the hood. This abstraction reduces maintenance overhead for you. Also, it allows managed disks to provide numerous benefits over unmanaged disks, such as high availability, better scalability, large disks, bursting, and shared disks. If you use unmanaged disks, start planning your Windows or Linux migration now.

How am I affected?

After March 31, 2026, you can't start IaaS VMs that use unmanaged disks. Any VMs that use unmanaged disks and are running or allocated will be stopped and deallocated.

What is being retired?

Unmanaged disks are a type of page blob in Azure that is used for storing Virtual Hard Disk (VHD) files associated with virtual machines (VM). When a page blob VHD is attached to a VM, it functions as a virtual disk for that VM. The VM's operating system can read from and write to the attached page blob as if it were a SCSI volume. This retirement only affects page blobs being used as virtual disks that are directly attached to VMs.

Page blobs accessed directly via HTTP or HTTPS REST APIs are standalone entities and have no dependencies on any specific VM. Clients can interact with these page blobs by using standard HTTP or HTTPS protocols, making requests to read from or write to the blobs by using Storage REST APIs. Since these page blobs aren't attached as virtual disks, this retirement doesn't affect them.

Third party storage offerings on Azure that use page blobs via HTTP or HTTPS REST APIs as their underlying storage solution might not be affected by this retirement.

What actions should I take?

Start planning your migration to Azure managed disks today.

- Make a list of all affected VMs:
  - On the Azure portal's VM pane, add the **Uses managed disks** filter and set its value to **No**, to get a list of all the affected VMs within the subscription.
  - You can also query Azure Resource Graph by using the portal to view the list of all flagged VMs and related information for the selected subscriptions.
  - On February 28, 2020, we sent out emails to subscription owners with a list of all subscriptions that contain these VMs. Use them to build this list.
- Now that you have a list of all the affected VMs, see Migrate Azure VMs to managed disks in Azure to learn how to migrate your unmanaged disks to managed disks. If you have other questions, see Frequently asked questions about migrating to managed disks.
- If you have further technical questions, problems, and need help with adding subscriptions to the allow list, contact support.

Complete your migration as soon as possible to prevent business impact and to take advantage of the improved reliability, scalability, security, and new features of Azure managed disks.

What resources are available for this migration?

- Microsoft Q&A: Microsoft and community support for migration.
- Azure Migration Support: Dedicated support team for technical assistance during migration.
- If your company or organization has partnered with Microsoft or works with Microsoft representatives such as cloud solution architects (CSAs) or technical account managers (TAMs), work with them for more resources for migration.

Last updated on 03/31/2026

Migrate your Standard HDD OS disks by September 08, 2028

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs

On September 8, 2028, the ability to use Standard HDDs as OS disks will be retired. After that date, all Standard HDDs being used as OS disks will be automatically converted to Standard SSDs of equivalent size. To avoid potential service disruptions during the conversion to Standard SSD after the retirement date, convert your Standard HDD OS disks to either Standard SSD or Premium SSDs before the retirement date.

If you're using Standard HDD as OS disks, begin planning a migration now. Generally, Standard SSD provides the closest price to performance ratio as Standard HDDs. If you need higher performance, migrate to Premium SSD.

How am I affected?

After September 8, 2028, any existing virtual machines using Standard HDD OS Disks will have their OS disks automatically converted to a Standard SSD of equivalent size. Workloads on these disks may experience a service disruption if they're not migrated before September 8, 2028.

What is being retired?

This retirement is only for the ability to use Standard HDDs as OS disks. None of the other managed disk types or ephemeral OS disks will be affected by this retirement.

What actions should I take?

Stop creating new virtual machines with Standard HDD OS disks. Use Standard SSD or Premium SSD for new OS disks instead.

To start planning a migration to either Standard SSD or Premium SSD OS disks, first, make a list of all affected OS disks and VMs.

If you have multiple subscriptions, use the Disk Storage Center to create a list. Add two filters, one for **Storage type** which should be equal to **Standard HDD LRS** and one for **OS type** which should equal to **Linux and Windows**. This produces a list of all Standard HDD OS disks

across all your subscriptions. The **Owner** column is the name of the virtual machine that uses the listed Standard HDD OS disks.

The Disk Storage Center still works for individual subscriptions. But, if you prefer, you can use the Azure CLI or the Azure PowerShell module to get the same list for an individual subscription. For the Azure PowerShell module, use `Get-AzDisk | Where-Object { $_.Sku.Name -eq "Standard_LRS" -and $_.OsType }` and for the Azure CLI, use `az disk list --query "[?sku.name=='Standard_LRS' && osType!=null]" --output table`.

Once you have a list of Standard HDD OS disks, convert your Standard HDD OS disks to either Standard SSD or Premium SSDs. Review Azure Disk Storage billing guidance to choose the most cost-effective disk type for your workloads.

For technical questions and issues, contact support.

What resources are available for this migration?

Microsoft Q&A: Microsoft and community support for migration.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 02/26/2026

Migrate a Linux virtual machine from unmanaged disks to managed disks

Article • 08/23/2024

Applies to: ✔ Linux VMs

If you have existing Linux virtual machines (VMs) that use unmanaged disks, you can migrate the VMs to use Azure Managed Disks. This process converts both the OS disk and any attached data disks.

This article shows you how to migrate VMs by using the Azure CLI. If you need to install or upgrade it, see Install Azure CLI.

Before you begin

- Review the FAQ about migration to Managed Disks.
- The migration will restart the VM, so schedule the migration of your VMs during a pre-existing maintenance window.
- The migration isn't reversible.
- Any users with the Virtual Machine Contributor role won't be able to change the VM size (as they could pre-migration). This is because VMs with managed disks require the user to have the Microsoft.Compute/disks/write permission on the OS disks.
- Be sure to test the migration. Migrate a test virtual machine before you perform the migration in production.
- During the migration, you deallocate the VM. The VM receives a new IP address when it's started after the migration. If needed, you can assign a static IP address to the VM.
- Review the minimum version of the Azure VM agent required to support the migration process. For information on how to check and update your agent version, see Minimum version support for VM agents in Azure
- The original VHDs and the storage account used by the VM before migration are not deleted. They continue to incur charges. To avoid being billed for these artifacts, delete the original VHD blobs after you verify that the migration is

complete. If you need to find these unattached disks in order to delete them, see our article Find and delete unattached Azure managed and unmanaged disks.

Migrate single-instance VMs

This section covers how to migrate single-instance Azure VMs from unmanaged disks to managed disks. (If your VMs are in an availability set, see the next section.) You can use this process to migrate the VMs from premium (SSD) unmanaged disks to premium managed disks, or from standard (HDD) unmanaged disks to standard managed disks.

1. Deallocate the VM by using az vm deallocate. The following example deallocates the VM named `myVM` in the resource group named `myResourceGroup`:

   Azure CLI

   ```
   az vm deallocate --resource-group myResourceGroup --name myVM
   ```

2. Migrate the VM to managed disks by using az vm convert. The following process converts the VM named `myVM`, including the OS disk and any data disks:

   Azure CLI

   ```
   az vm convert --resource-group myResourceGroup --name myVM
   ```

3. Start the VM after the migration to managed disks by using az vm start. The following example starts the VM named `myVM` in the resource group named `myResourceGroup`.

   Azure CLI

   ```
   az vm start --resource-group myResourceGroup --name myVM
   ```

Migrate VMs in an availability set

If the VMs that you want to migrate to managed disks are in an availability set, you first need to migrate the availability set to a managed availability set.

All VMs in the availability set must be deallocated before you migrate the availability set. Plan to migrate all VMs to managed disks after the availability set itself has been converted to a managed availability set. Then, start all the VMs and continue operating as normal.

1. List all VMs in an availability set by using az vm availability-set list. The following example lists all VMs in the availability set named `myAvailabilitySet` in the resource group named `myResourceGroup`:

   Azure CLI

   ```
   az vm availability-set show \
       --resource-group myResourceGroup \
       --name myAvailabilitySet \
       --query [virtualMachines[*].id] \
       --output table
   ```

2. Deallocate all the VMs by using az vm deallocate. The following example deallocates the VM named `myVM` in the resource group named `myResourceGroup`:

   Azure CLI

   ```
   az vm deallocate --resource-group myResourceGroup --name myVM
   ```

3. Migrate the availability set by using az vm availability-set convert. The following example converts the availability set named `myAvailabilitySet` in the resource group named `myResourceGroup`:

   Azure CLI

   ```
   az vm availability-set convert \
       --resource-group myResourceGroup \
       --name myAvailabilitySet
   ```

4. Migrate all the VMs to managed disks by using az vm convert. The following process converts the VM named `myVM`, including the OS disk and any data disks:

   Azure CLI

   ```
   az vm convert --resource-group myResourceGroup --name myVM
   ```

5. Start all the VMs after the migration to managed disks by using az vm start. The following example starts the VM named `myVM` in the resource group named `myResourceGroup`:

   Azure CLI

   ```
   az vm start --resource-group myResourceGroup --name myVM
   ```

Migrate using the Azure portal

You can also migrate unmanaged disks to managed disks using the Azure portal.

1. Sign in to the Azure portal.
2. Select the VM from the list of VMs in the portal.
3. In the blade for the VM, select **Disks** from the menu.
4. At the top of the **Disks** blade, select **Migrate to managed disks**.
5. If your VM is in an availability set, there will be a warning on the **Migrate to managed disks** blade that you need to migrate the availability set first. The warning should have a link you can click to migrate the availability set. Once the availability set is converted or if your VM is not in an availability set, click **Migrate** to start the process of migrating your disks to managed disks.

The VM will be stopped and restarted after migration is complete.

Next steps

For more information about storage options, see Azure Managed Disks overview.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate a Windows virtual machine from unmanaged disks to managed disks

Summarize this article for me

Applies to: ✔ Windows VMs

On March 31st, 2026, unmanaged disks are being retired. If you're using unmanaged disks, migrate to managed disks before that date. For details on the retirement, see Migrate your Azure unmanaged disks by March 31st, 2026.

If you have existing Windows virtual machines (VMs) that use unmanaged disks, you can migrate the VMs to use managed disks through the Azure Managed Disks service. This process converts both the operating system (OS) disk and any attached data disks.

Before you begin

- Review Plan for the migration to Managed Disks.
- Review the FAQ about migration to Managed Disks.
- Ensure the VM is in a healthy sate before converting.
- The migration will restart the VM, so schedule the migration of your VMs during a pre-existing maintenance window.
- The migration isn't reversible.
- Be sure to test the migration. Migrate a test virtual machine before you perform the migration in production.
- During the migration, you deallocate the VM. The VM receives a new IP address when it's started after the migration. If needed, you can assign a static IP address to the VM.
- Review the minimum version of the Azure VM agent required to support the migration process. For information on how to check and update your agent version, see Minimum version support for VM agents in Azure
- The original VHDs and the storage account used by the VM before migration are not deleted. They continue to incur charges. To avoid being billed for these artifacts, delete the original VHD blobs after you verify that the migration is complete. If you need to find these unattached disks in order to delete them, see our article Find and delete unattached Azure managed and unmanaged disks.

Migrate single-instance VMs

This section covers how to migrate single-instance Azure VMs from unmanaged disks to managed disks. (If your VMs are in an availability set, see the next section.)

1. Deallocate the VM by using the Stop-AzVM cmdlet. The following example deallocates the VM named `myVM` in the resource group named `myResourceGroup`:

   Azure PowerShell

   ```
   $rgName = "myResourceGroup"
   $vmName = "myVM"
   Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force
   ```

2. Migrate the VM to managed disks by using the ConvertTo-AzVMManagedDisk cmdlet. The following process converts the previous VM, including the OS disk and any data disks, and starts the Virtual Machine:

   Azure PowerShell

   ```
   ConvertTo-AzVMManagedDisk -ResourceGroupName $rgName -VMName $vmName
   ```

Migrate VMs in an availability set

If the VMs that you want to migrate to managed disks are in an availability set, you first need to migrate the availability set to a managed availability set.

1. Migrate the availability set by using the Update-AzAvailabilitySet cmdlet. The following example updates the availability set named `myAvailabilitySet` in the resource group named `myResourceGroup`:

   Azure PowerShell

   ```
   $rgName = 'myResourceGroup'
   $avSetName = 'myAvailabilitySet'

   $avSet = Get-AzAvailabilitySet -ResourceGroupName $rgName -Name $avSetName
   Update-AzAvailabilitySet -AvailabilitySet $avSet -Sku Aligned
   ```

   If the region where your availability set is located has only 2 managed fault domains but the number of unmanaged fault domains is 3, this command shows an error similar to "The specified fault domain count 3 must fall in the range 1 to 2." To resolve the error, update the fault domain to 2 and update `Sku` to `Aligned` as follows:

Azure PowerShell

```
$avSet.PlatformFaultDomainCount = 2
Update-AzAvailabilitySet -AvailabilitySet $avSet -Sku Aligned
```

2. Deallocate and migrate the VMs in the availability set. The following script deallocates each VM by using the Stop-AzVM cmdlet, converts it by using ConvertTo-AzVMManagedDisk, and restarts it automatically as apart of the migration process:

Azure PowerShell

```
$avSet = Get-AzAvailabilitySet -ResourceGroupName $rgName -Name $avSetName

foreach($vmInfo in $avSet.VirtualMachinesReferences)
{
  $vm = Get-AzVM -ResourceGroupName $rgName | Where-Object {$_.Id -eq
$vmInfo.id}
  Stop-AzVM -ResourceGroupName $rgName -Name $vm.Name -Force
  ConvertTo-AzVMManagedDisk -ResourceGroupName $rgName -VMName $vm.Name
}
```

Troubleshooting

- Before converting, make sure all the VM extensions are in the 'Provisioning succeeded' state or the migration will fail with the error code 409.
- If there is an error during migration, or if a VM is in a failed state because of issues in a previous migration, run the `ConvertTo-AzVMManagedDisk` cmdlet again. A simple retry usually unblocks the situation.
- If you are converting a Linux VM to managed disks, use the latest version of the Azure Linux Agent. Operations using Azure Linux Agent versions '2.2.0' and earlier will likely fail. Running the migration on a generalized VM or a VM that belongs to a classic availability set is also not supported.
- If the migration fails with the "SnapshotCountExceeded" error, delete some snapshots and attempt the operation again.

Migrate using the Azure portal

You can also migrate unmanaged disks to managed disks using the Azure portal.

1. Sign in to the Azure portal.
2. Select the VM from the list of VMs in the portal.
3. In the blade for the VM, select **Disks** from the menu.
4. At the top of the **Disks** blade, select **Migrate to managed disks**.

5. If your VM is in an availability set, there will be a warning on the **Migrate to managed disks** blade that you need to migrate the availability set first. The warning should have a link you can click to migrate the availability set. Once the availability set is converted or if your VM is not in an availability set, click **Migrate** to start the process of migrating your disks to managed disks.

The VM will be stopped and restarted after migration is complete.

Next steps

Change the disk type of an Azure Managed Disk.

Take a read-only copy of a VM by using snapshots.

Last updated on 02/22/2026

Create a managed disk from a VHD file in a storage account in the same subscription with CLI (Linux)

Article • 08/22/2024

This script creates a managed disk from a VHD file in a storage account in the same subscription. Use this script to import a specialized (not generalized/sysprepped) VHD to managed OS disk to create a virtual machine. Or, use it to import a data VHD to managed data disk.

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Run the script

Azure CLI

```
#Provide the subscription Id
subscriptionId="<subscriptionId>"

#Provide the name of your resource group.
#Ensure that resource group is already created
resourceGroupName=myResourceGroupName

#Provide the name of the Managed Disk
diskName=myDiskName

#Provide the size of the disks in GB. It should be greater than the VHD file
size.
diskSize=128


#Provide the URI of the VHD file that will be used to create Managed Disk.
```

```
# VHD file can be deleted as soon as Managed Disk is created.
# e.g.
https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd
vhdUri=https://contosostorageaccount1.blob.core.windows.net/vhds/contosoumd7
8620170425131836.vhd

#Provide the storage type for the Managed Disk. Premium_LRS or Standard_LRS.
storageType=Premium_LRS


#Provide the Azure location (e.g. westus) where Managed Disk will be
located.
#The location should be same as the location of the storage account where
VHD file is stored.
#Get all the Azure location supported for your subscription using command
below:
#az account list-locations
location=westus

#If you're creating an OS disk, uncomment the following lines and replace
the values
#osType = 'yourOSType' #Acceptable values are either Windows or Linux
#hyperVGeneration = 'yourHyperVGen' #Acceptable values are either v1 or v2

#Set the context to the subscription Id where Managed Disk will be created
az account set --subscription $subscriptionId

#Create the Managed disk from the VHD file
#If you're creating an OS disk, add the following: --os-type $osType -hyper-
v-generation $hyperVGeneration
az disk create --resource-group $resourceGroupName --name $diskName --sku
$storageType --location $location --size-gb $diskSize --source $vhdUri
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name myResourceGroupName
```

Sample reference

This script uses following commands to create a managed disk from a VHD. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az disk create	Creates a managed disk using URI of a VHD in a storage account in the same subscription

Next steps

Create a virtual machine by attaching a managed disk as OS disk

For more information on the Azure CLI, see Azure CLI documentation.

Additional virtual machine and managed disks CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a managed disk from a snapshot with CLI (Linux)

Article • 08/22/2024

This article contains two scripts for creating a managed disk from a snapshot. The first script is for a managed disk with platform-managed keys and the second script is for a managed disk with customer-managed keys. Use these scripts to restore a virtual machine from snapshots of OS and data disks. Create OS and data managed disks from respective snapshots and then create a new virtual machine by attaching managed disks. You can also restore data disks of an existing VM by attaching data disks created from snapshots.

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Disks with platform-managed keys

Azure CLI

```
#Provide the subscription Id of the subscription where you want to create
Managed Disks
subscriptionId="<subscriptionId>"

#Provide the name of your resource group
resourceGroupName=myResourceGroupName

#Provide the name of the snapshot that will be used to create Managed Disks
snapshotName=mySnapshotName

#Provide the name of the new Managed Disks that will be create
diskName=myDiskName

#Provide the size of the disks in GB. It should be greater than the VHD file
size.
diskSize=128
```

```
#Provide the storage type for Managed Disk. Acceptable values are
Standard_LRS, Premium_LRS, PremiumV2_LRS, StandardSSD_LRS, UltraSSD_LRS,
Premium_ZRS, and StandardSSD_ZRS.
storageType=Premium_LRS

#Required for Premium SSD v2 and Ultra Disks
#Provide the Availability Zone you'd like the disk to be created in, default
is 1
zone=1

#Set the context to the subscription Id where Managed Disk will be created
az account set --subscription $subscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group
$resourceGroupName --query [id] -o tsv)

#Create a new Managed Disks using the snapshot Id
#Note that managed disk will be created in the same location as the snapshot
#If you're creating a Premium SSD v2 or an Ultra Disk, add "--zone $zone" to
the end of the command
az disk create --resource-group $resourceGroupName --name $diskName --sku
$storageType --size-gb $diskSize --source $snapshotId
```

Disks with customer-managed keys

Azure CLI

```
#Provide the subscription Id of the subscription where you want to create
Managed Disks
subscriptionId="<subscriptionId>"

#Provide the name of your resource group
resourceGroupName=myResourceGroupName

#Provide the name of the snapshot that will be used to create Managed Disks
snapshotName=mySnapshotName

#Provide the name of the new Managed Disks that will be create
diskName=myDiskName

#Provide the size of the disks in GB. It should be greater than the VHD file
size.
diskSize=128

#Provide the storage type for Managed Disk. Premium_LRS or Standard_LRS.
storageType=Premium_LRS

#Provide the name of the target disk encryption set
diskEncryptionSetName=myName
```

```
#Provide the target disk encryption set resource group
diskEncryptionResourceGroup=myGroup

#Required for Premium SSD v2 and Ultra Disks
#Provide the Availability Zone you'd like the disk to be created in, default
is 1
zone=1

#Set the context to the subscription Id where Managed Disk will be created
az account set --subscription $subscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group
$resourceGroupName --query [id] -o tsv)

#Get the disk encryption set ID
diskEncryptionSetId=$(az disk-encryption-set show --name
$diskEncryptionSetName --resource-group $diskEncryptionResourceGroup)

#Create a new Managed Disks using the snapshot Id
#Note that managed disk will be created in the same location as the snapshot
#To change the location, add the --location parameter
#If you're creating a Premium SSD v2 or an Ultra Disk, add "--zone $zone" to
the end of the command
az disk create -g $resourceGroupName -n $diskName --source $snapshotId --
disk-encryption-set $diskEncryptionSetID --location eastus2euap
```

Performance impact - background copy process

When you create a managed disk from a snapshot, it starts a background copy process. You can attach a disk to a VM while this process is running but you'll experience performance impact (4k disks experience read impact, 512e experience both read and write impact) with higher latency, lower IOPS and throughput until background copy completes. For Ultra Disks and Premium SSD v2, you can check the status of the background copy process with the following commands:

> **ⓘ Important**
>
> You can't use the following sections to get the status of the background copy process for disk types other than Ultra Disk or Premium SSD v2. Other disk types will always report 100%.

Azure CLI

```
subscriptionId=yourSubscriptionID
resourceGroupName=yourResourceGroupName
diskName=yourDiskName
az account set --subscription $subscriptionId
az disk show -n $diskName -g $resourceGroupName --query [completionPercent]
-o tsv
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name myResourceGroupName
```

Sample reference

This script uses following commands to create a managed disk from a snapshot. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az snapshot show	Gets all the properties of a snapshot using the name and resource group properties of the snapshot. ID property is used to create managed disk.
az disk create	Creates a managed disk using snapshot ID of a managed snapshot

Next steps

Create a virtual machine by attaching a managed disk as OS disk

For more information on the Azure CLI, see Azure CLI documentation.

More virtual machine and managed disks CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a managed disk from a VHD file in a storage account in same or different subscription with PowerShell

Article • 12/19/2024

This script creates a managed disk from a VHD file in a storage account in same or different subscription. Use this script to import a specialized (not generalized/sysprepped) VHD to managed OS disk to create a virtual machine. Also, use it to import a data VHD to managed data disk.

Don't create multiple identical managed disks from a VHD file in small amount of time. To create managed disks from a vhd file, blob snapshot of the vhd file is created and then it is used to create managed disks. Only one blob snapshot can be created in a minute that causes disk creation failures due to throttling. To avoid this throttling, create a managed snapshot from the vhd file and then use the managed snapshot to create multiple managed disks in short amount of time.

If you don't have an Azure subscription, create an Azure free account before you begin.

Sample script

PowerShell

```
<#

.DESCRIPTION

This sample demonstrates how to create a Managed Disk from a VHD file.
Create Managed Disks from VHD files in following scenarios:
1. Create a Managed OS Disk from a specialized VHD file. A specialized VHD
is a copy of VHD from an exisitng VM that maintains the user accounts,
applications and other state data from your original VM.
   Attach this Managed Disk as OS disk to create a new virtual machine.
2. Create a Managed data Disk from a VHD file. Attach the Managed Disk to an
existing VM or attach it as data disk to create a new virtual machine.

.NOTES

1. Before you use this sample, please install the latest version of Azure
PowerShell from here: http://go.microsoft.com/?linkid=9811175&clcid=0x409
2. Provide the appropriate values for each variable. Note: The angled
brackets should not be included in the values you provide.
```

```
#>

#Provide the subscription Id
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group
$resourceGroupName ='yourResourceGroupName'

#Provide the name of the Managed Disk
$diskName = 'yourDiskName'

#Provide the size of the disks in GB. It should be greater than the VHD file
size.
$diskSize = '128'

#Provide the URI of the VHD file that will be used to create Managed Disk.
# VHD file can be deleted as soon as Managed Disk is created.
# e.g. https://contosostorageaccount1.blob.core.windows.net/vhds/contoso-um-
vm120170302230408.vhd
$vhdUri =
'https://contososts torageaccount1.blob.core.windows.net/vhds/contosovhd123.v
hd'

#Provide the resource Id of the storage account where VHD file is stored.
#e.g. /subscriptions/6472s1g8-h217-446b-b509-
314e17e1efb0/resourceGroups/MDDemo/providers/Microsoft.Storage/storageAccoun
ts/contosostorageaccount
$storageAccountId =
'/subscriptions/yourSubscriptionId/resourceGroups/yourResourceGroupName/prov
iders/Microsoft.Storage/storageAccounts/yourStorageAccountName'

#Provide the storage type for the Managed Disk. PremiumLRS or StandardLRS.
$sku = 'Premium_LRS'

#Provide the Azure location (e.g. westus) where Managed Disk will be
located.
#The location should be same as the location of the storage account where
VHD file is stored.
#Get all the Azure location using command below:
#Get-AzureRmLocation
$location = 'westus'

#Set the context to the subscription Id where Managed Disk will be created
Set-AzContext -Subscription $subscriptionId

#If you're creating an OS disk, add the following lines
#Acceptable values are either Windows or Linux
#$OSType = 'yourOSType'
#Acceptable values are either V1 or V2
#$HyperVGeneration = 'yourHyperVGen'

#If you're creating an OS disk, add -HyperVGeneration and -OSType parameters
$diskConfig = New-AzDiskConfig -SkuName $sku -Location $location -DiskSizeGB
```

```
$diskSize -StorageAccountId $storageAccountId -SourceUri $vhdUri -
CreateOption Import

#Create Managed disk
New-AzDisk -DiskName $diskName -Disk $diskConfig -ResourceGroupName
$resourceGroupName
```

Script explanation

This script uses following commands to create a managed disk from a VHD in different subscription. Each command in the table links to command specific documentation.

Expand table

Command	Notes
New-AzDiskConfig	Creates disk configuration that is used for disk creation. It includes storage type, location, resource ID of the storage account where the parent VHD is stored, VHD URI of the parent VHD.
New-AzDisk	Creates a disk using disk configuration, disk name, and resource group name passed as parameters.

Next steps

Create a virtual machine by attaching a managed disk as OS disk

For more information on the Azure PowerShell module, see Azure PowerShell documentation.

Additional virtual machine PowerShell script samples can be found in the Azure Windows VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a managed disk from a snapshot with PowerShell

This script creates a managed disk from a snapshot. Use it to restore a virtual machine from snapshots of OS and data disks. Create OS and data managed disks from respective snapshots and then create a new virtual machine by attaching managed disks. You can also restore data disks of an existing VM by attaching data disks created from snapshots.

If you don't have an Azure account, create a free account before you begin.

Sample script

PowerShell

```
#Provide the subscription Id
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group
$resourceGroupName ='yourResourceGroupName'

#Provide the name of the snapshot that will be used to create Managed Disks
$snapshotName = 'yourSnapshotName'

#Provide the name of the Managed Disk
$diskName = 'yourManagedDiskName'

#Provide the size of the disks in GB. It should be greater than the VHD file size.
$diskSize = '128'

#Provide the storage type for Managed Disk. Acceptable values are Standard_LRS,
Premium_LRS, PremiumV2_LRS, StandardSSD_LRS, UltraSSD_LRS, Premium_ZRS and
StandardSSD_ZRS.
$storageType = 'Premium_LRS'

#Required for Premium SSD v2 and Ultra Disks
#Provide the Availability Zone you'd like the disk to be created in, default is 1
$zone=1

#Provide the Azure region (e.g. westus) where Managed Disks will be located.
#This location should be same as the snapshot location
#Get all the Azure location using command below:
#Get-AzLocation
$location = 'westus'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$snapshot = Get-AzSnapshot -ResourceGroupName $resourceGroupName -SnapshotName
$snapshotName
```

```
#If you're creating a Premium SSD v2 or an Ultra Disk, add "-Zone $zone" to the end
of the command
$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -
CreateOption Copy -SourceResourceId $snapshot.Id -DiskSizeGB $diskSize

New-AzDisk -Disk $diskConfig -ResourceGroupName $resourceGroupName -DiskName
$diskName
```

Performance impact - background copy process

When you create a managed disk from a snapshot, it starts a background copy process. You can attach a disk to a VM while this process is running but you'll experience performance impact (4k disks experience read impact, 512e experience both read and write impact). For Ultra Disks and Premium SSD v2, you can check the status of the background copy process with the Azure CLI or Azure PowerShell module. You can only check the status of the background copy process for Ultra Disks or Premium SSD v2 disks, other disk types always report 100%.

Script explanation

This script uses following commands to create a managed disk from a snapshot. Each command in the table links to command specific documentation.

Expand table

Command	Notes
Get-AzSnapshot	Gets snapshot properties.
New-AzDiskConfig	Creates disk configuration that is used for disk creation. It includes the resource Id of the parent snapshot, location that is same as the location of parent snapshot and the storage type.
New-AzDisk	Creates a disk using disk configuration, disk name, and resource group name passed as parameters.

Next steps

Create a virtual machine from a managed disk

For more information on the Azure PowerShell module, see Azure PowerShell documentation.

Additional virtual machine PowerShell script samples can be found in the Azure Windows VM documentation.

Last updated on 01/06/2026

Create a snapshot from a VHD to create multiple identical managed disks in small amount of time with PowerShell (Windows)

Article • 08/22/2024

This script creates a snapshot from a VHD file in a storage account in same or different subscription. Use this script to import a specialized (not generalized/sysprepped) VHD to a snapshot and then use the snapshot to create multiple identical managed disks in small amount of time. Also, use it to import a data VHD to a snapshot and then use the snapshot to create multiple managed disks in small amount of time.

If you don't have an Azure subscription, create an Azure free account before you begin.

Sample script

PowerShell

```
#Provide the subscription Id where snapshot will be created
$subscriptionId = 'yourSubscriptionId'

#Provide the name of your resource group where snapshot will be created.
$resourceGroupName ='yourResourceGroupName'

#Provide the name of the snapshot
$snapshotName = 'yourSnapshotName'

#Provide the storage type for snapshot. PremiumLRS or StandardLRS.
$storageType = 'StandardLRS'

#Provide the Azure region (e.g. westus) where snapshot will be located.
#This location should be same as the storage account location where VHD file
is stored
#Get all the Azure location using command below:
#Get-AzLocation
$location = 'westus'

#Provide the URI of the VHD file (page blob) in a storage account. Please
not that this is NOT the SAS URI of the storage container where VHD file is
stored.
#e.g.
https://contosostorageaccount1.blob.core.windows.net/vhds/contosovhd123.vhd
#Note: VHD file can be deleted as soon as Managed Disk is created.
```

```
$sourceVHDURI =
'https://yourStorageAccountName.blob.core.windows.net/vhds/yourVHDName.vhd'

#Provide the resource Id of the storage account where VHD file is stored.
#e.g. /subscriptions/6582b1g7-e212-446b-b509-
314e17e1efb0/resourceGroups/MDDemo/providers/Microsoft.Storage/storageAccoun
ts/contosostorageaccount1
#This is an optional parameter if you are creating snapshot in the same
subscription
$storageAccountId =
'/subscriptions/yourSubscriptionId/resourceGroups/yourResourceGroupName/prov
iders/Microsoft.Storage/storageAccounts/yourStorageAccountName'

#Set the context to the subscription Id where Managed Disk will be created
Select-AzSubscription -SubscriptionId $SubscriptionId

$snapshotConfig = New-AzSnapshotConfig -AccountType $storageType -Location
$location -CreateOption Import -StorageAccountId $storageAccountId -
SourceUri $sourceVHDURI

New-AzSnapshot -Snapshot $snapshotConfig -ResourceGroupName
$resourceGroupName -SnapshotName $snapshotName
```

Next steps

Create a managed disk from snapshot

Create a virtual machine by attaching a managed disk as OS disk

For more information on the Azure PowerShell module, see Azure PowerShell documentation.

Additional virtual machine PowerShell script samples can be found in the Azure Windows VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a virtual machine from a snapshot with CLI

Article • 08/22/2024

This script creates a virtual machine from a snapshot of an OS disk.

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Run the script

Azure CLI

```
#Provide the subscription Id of the subscription where you want to create
Managed Disks
subscriptionId="<subscriptionId>"

#Provide the name of your resource group
resourceGroupName=myResourceGroupName

#Provide the name of the snapshot that will be used to create Managed Disks
snapshotName=mySnapshotName

#Provide the name of the Managed Disk
osDiskName=myOSDiskName

#Provide the size of the disks in GB. It should be greater than the VHD file
size.
diskSize=128

#Provide the storage type for Managed Disk. Premium_LRS or Standard_LRS.
storageType=Premium_LRS

#Provide the OS type
osType=linux

#Provide the name of the virtual machine
virtualMachineName=myVirtualMachineName
```

```
#Set the context to the subscription Id where Managed Disk will be created
az account set --subscription $subscriptionId

#Get the snapshot Id
snapshotId=$(az snapshot show --name $snapshotName --resource-group
$resourceGroupName --query [id] -o tsv)

#Create a new Managed Disks using the snapshot Id
az disk create --resource-group $resourceGroupName --name $osDiskName --sku
$storageType --size-gb $diskSize --source $snapshotId

#Create VM by attaching created managed disks as OS
az vm create --name $virtualMachineName --resource-group $resourceGroupName
--attach-os-disk $osDiskName --os-type $osType
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name myResourceGroupName
```

Sample reference

This script uses the following commands to create a managed disk, virtual machine, and all related resources. Each command in the table links to command specific documentation.

Expand table

Command	Notes
az snapshot show	Gets snapshot using snapshot name and resource group name. Id property of the returned object is used to create a managed disk.
az disk create	Creates managed disks from a snapshot using snapshot Id, disk name, storage type, and size
az vm create	Creates a VM using a managed OS disk

Next steps

For more information on the Azure CLI, see Azure CLI documentation.

Additional virtual machine CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a virtual machine using an existing managed OS disk with CLI

Article • 08/22/2024

This script creates a virtual machine by attaching an existing managed disk as OS disk. Use this script in preceding scenarios:

- Create a VM from an existing managed OS disk that was copied from a managed disk in different subscription
- Create a VM from an existing managed disk that was created from a specialized VHD file
- Create a VM from an existing managed OS disk that was created from a snapshot

If you don't have an Azure subscription, create an Azure free account before you begin.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

Launch Azure Cloud Shell

The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com.

When Cloud Shell opens, verify that **Bash** is selected for your environment. Subsequent sessions will use Azure CLI in a Bash environment, Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press **Enter** to run it.

Sign in to Azure

Cloud Shell is automatically authenticated under the initial account signed-in with. Use the following script to sign in using a different subscription, replacing `<Subscription ID>` with your Azure Subscription ID. If you don't have an Azure subscription, create an Azure free account before you begin.

Azure CLI

```
subscription="<subscriptionId>" # add subscription here

az account set -s $subscription # ...or use 'az login'
```

For more information, see set active subscription or log in interactively

Run the script

Azure CLI

```
#Provide the subscription Id
subscriptionId="<subscriptionId>"

#Provide the name of your resource group
resourceGroupName=myResourceGroupName

#Provide the name of the Managed Disk
managedDiskName=myDiskName

#Provide the OS type
osType=linux

#Provide the name of the virtual machine
virtualMachineName=myVirtualMachineName123

#Set the context to the subscription Id where Managed Disk exists and where
```

```
VM will be created
az account set --subscription $subscriptionId

#Get the resource Id of the managed disk
managedDiskId=$(az disk show --name $managedDiskName --resource-group
$resourceGroupName --query [id] -o tsv)

#Create VM by attaching existing managed disks as OS
az vm create --name $virtualMachineName --resource-group $resourceGroupName
--attach-os-disk $managedDiskId --os-type $osType
```

Clean up resources

Run the following command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name myResourceGroupName
```

Sample reference

This script uses the following commands to get managed disk properties, attach a managed disk to a new VM and create a VM. Each item in the table links to command specific documentation.

Expand table

Command	Notes
az disk show	Gets managed disk properties using disk name and resource group name. Id property is used to attach a managed disk to a new VM
az vm create	Creates a VM using a managed OS disk

Next steps

For more information on the Azure CLI, see Azure CLI documentation.

Additional virtual machine CLI script samples can be found in the Azure Linux VM documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Add a disk to a Linux VM

09/18/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to attach a persistent disk to your virtual machine (VM) to preserve your data, even if your VM is reprovisioned due to maintenance or resizing.

Attach a new disk to a VM

If you want to add a new, empty data disk to your VM, use the az vm disk attach command with the `--new` parameter. If your VM is in an Availability Zone, the disk is automatically created in the same zone as the VM. For more information, see Overview of Availability Zones. The following example creates a disk named *myDataDisk* that is 50 GB in size:

Azure CLI

```
az vm disk attach \
   -g myResourceGroup \
   --vm-name myVM \
   --name myDataDisk \
   --new \
   --size-gb 50
```

Lower latency

In select regions, disk attach latency is reduced. In those regions, there's an improvement of up to 15%. This improvement is useful if you have planned or unplanned failovers between VMs, you're scaling your workload, or you're running a high-scale stateful workload such as Azure Kubernetes Service. However, this improvement is limited to the explicit disk attach command, `az vm disk attach`. You won't see the performance improvement if you call a command that might implicitly perform an attach, like `az vm update`. You don't need to take any action other than calling the explicit attach command to see this improvement.

Lower latency is currently available in every public region except for:

- Canada Central
- Central US
- East US
- East US 2
- South Central US
- West US 2

- Germany North
- India West
- North Europe
- West Europe

Attach an existing disk

To attach an existing disk, find the disk ID and pass the ID to the az vm disk attach command. The following example queries for a disk named *myDataDisk* in *myResourceGroup*, then attaches it to the VM named *myVM*:

Azure CLI

```
diskId=$(az disk show -g myResourceGroup -n myDataDisk --query 'id' -o tsv)

az vm disk attach -g myResourceGroup --vm-name myVM --name $diskId
```

Identifying disks

Azure Linux VMs use different disk interfaces depending on the VM size and generation:

- VM sizes v6 and newer: Use NVMe interface for improved performance
- VM sizes v5 and older: Use SCSI interface for disk management

For details about SCSI vs NVMe differences, see SCSI to NVMe conversion.

Connect to the virtual machine

To identify disks associated with your Linux VM, connect to the VM by using SSH. For details, see How to use SSH with Linux on Azure. The following example connects to a VM with the public IP address of 10.123.123.25 with the username azureuser:

Bash

```
ssh azureuser@10.123.123.25
```

> **Note**
>
> Before identifying specific disks, determine whether your VM uses SCSI, NVMe, or a combination of both interfaces.

Azure-VM-Utils

The azure-vm-utils package provides utilities to optimize the Linux experience on Azure VMs, making disk identification more reliable across different VM configurations.

Use the following commands to list disks on the VM:

Bash

```
# List all disks
sudo azure-disk-list

# List NVMe disks with detailed information
sudo azure-nvme-id
```

The output from `azure-nvme-id` is similar to:

```
/dev/nvme0n1: type=os
/dev/nvme0n2: type=data, lun=0
/dev/nvme1n1: type=local, index=1, name=nvme-50G-1
```

Next Steps

- Format and mount the disks based on your requirements and use case. Review instructions for formatting and mounting managed disks and temporary disks.
- Learn about Azure-VM-Utils.

Attach a data disk to a Windows VM with PowerShell

Article • 12/09/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

This article shows you how to attach both new and existing disks to a Windows virtual machine by using PowerShell.

First, review these tips:

- The size of the virtual machine controls how many data disks you can attach. For more information, see Sizes for virtual machines.
- To use premium SSDs, you'll need a premium storage-enabled VM type, like the DS-series or GS-series virtual machine.

This article uses PowerShell within the Azure Cloud Shell, which is constantly updated to the latest version. To open the Cloud Shell, select **Try it** from the top of any code block.

Lower latency

In select regions, the disk attach latency has been reduced, so you'll see an improvement of up to 15%. This is useful if you have planned/unplanned failovers between VMs, you're scaling your workload, or are running a high scale stateful workload such as Azure Kubernetes Service. However, this improvement is limited to the explicit disk attach command, `Add-AzVMDataDisk`. You won't see the performance improvement if you call a command that may implicitly perform an attach, like `Update-AzVM`. You don't need to take any action other than calling the explicit attach command to see this improvement.

Lower latency is currently available in every public region except for:

- Canada Central
- Central US
- East US
- East US 2
- South Central US
- West US 2
- Germany North
- India West
- North Europe

- West Europe

Add an empty data disk to a virtual machine

This example shows how to add an empty data disk to an existing virtual machine.

Using managed disks

Azure PowerShell

```
$rgName = 'myResourceGroup'
$vmName = 'myVM'
$location = 'East US'
$storageType = 'Premium_LRS'
$dataDiskName = $vmName + '_datadisk1'

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -
CreateOption Empty -DiskSizeGB 128
$dataDisk1 = New-AzDisk -DiskName $dataDiskName -Disk $diskConfig -
ResourceGroupName $rgName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName
$vm = Add-AzVMDataDisk -VM $vm -Name $dataDiskName -CreateOption Attach -
ManagedDiskId $dataDisk1.Id -Lun 1

Update-AzVM -VM $vm -ResourceGroupName $rgName
```

Using managed disks in an Availability Zone

To create a disk in an Availability Zone, use New-AzDiskConfig with the `-Zone` parameter. The following example creates a disk in zone *1*.

PowerShell

```
$rgName = 'myResourceGroup'
$vmName = 'myVM'
$location = 'East US 2'
$storageType = 'Premium_LRS'
$dataDiskName = $vmName + '_datadisk1'

$diskConfig = New-AzDiskConfig -SkuName $storageType -Location $location -
CreateOption Empty -DiskSizeGB 128 -Zone 1
$dataDisk1 = New-AzDisk -DiskName $dataDiskName -Disk $diskConfig -
ResourceGroupName $rgName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName
$vm = Add-AzVMDataDisk -VM $vm -Name $dataDiskName -CreateOption Attach -
ManagedDiskId $dataDisk1.Id -Lun 1
```

```
Update-AzVM -VM $vm -ResourceGroupName $rgName
```

Initialize the disk

After you add an empty disk, you'll need to initialize it. To initialize the disk, you can sign in to a VM and use disk management. If you enabled WinRM and a certificate on the VM when you created it, you can use remote PowerShell to initialize the disk. You can also use a custom script extension:

Azure PowerShell

```
    $location = "location-name"
    $scriptName = "script-name"
    $fileName = "script-file-name"
    Set-AzVMCustomScriptExtension -ResourceGroupName $rgName -Location
$locName -VMName $vmName -Name $scriptName -TypeHandlerVersion "1.4" -
StorageAccountName "mystore1" -StorageAccountKey "primary-key" -FileName
$fileName -ContainerName "scripts"
```

The script file can contain code to initialize the disks, for example:

> **Note**
>
> The example script uses MBR partition style. If your disk is two tebibytes (TiB) or larger, you must use GPT partitioning. If it's under two TiB, you can use either MBR or GPT.

Azure PowerShell

```
    $disks = Get-Disk | Where partitionstyle -eq 'raw' | sort number

    $letters = 70..89 | ForEach-Object { [char]$_ }
    $count = 0
    $labels = "data1","data2"

    foreach ($disk in $disks) {
        $driveLetter = $letters[$count].ToString()
        $disk |
        Initialize-Disk -PartitionStyle MBR -PassThru |
        New-Partition -UseMaximumSize -DriveLetter $driveLetter |
        Format-Volume -FileSystem NTFS -NewFileSystemLabel $labels[$count] -
Confirm:$false -Force
    $count++
    }
```

Attach an existing data disk to a VM

You can attach an existing managed disk to a VM as a data disk.

Azure PowerShell

```
$rgName = "myResourceGroup"
$vmName = "myVM"
$dataDiskName = "myDisk"
$disk = Get-AzDisk -ResourceGroupName $rgName -DiskName $dataDiskName

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName

$vm = Add-AzVMDataDisk -CreateOption Attach -Lun 0 -VM $vm -ManagedDiskId
$disk.Id

Update-AzVM -VM $vm -ResourceGroupName $rgName
```

Next steps

You can also deploy managed disks using templates. For more information, see Using Managed Disks in Azure Resource Manager Templates or the quickstart template for deploying multiple data disks.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use the D: drive as a data drive on a Windows VM

Article • 08/22/2024

Applies to: ✔ Windows VMs

If your application needs to use the D drive to store data, follow these instructions to use a different drive letter for the temporary disk. Never use the temporary disk to store data that you need to keep.

If you resize or **Stop (Deallocate)** a virtual machine, this may trigger placement of the virtual machine to a new hypervisor. A planned or unplanned maintenance event may also trigger this placement. In this scenario, the temporary disk will be reassigned to the first available drive letter. If you have an application that specifically requires the D: drive, you need to follow these steps to temporarily move the pagefile.sys, attach a new data disk and assign it the letter D and then move the pagefile.sys back to the temporary drive. Once complete, Azure will not take back the D: if the VM moves to a different hypervisor.

For more information about how Azure uses the temporary disk, see Understanding the temporary drive on Microsoft Azure Virtual Machines

Attach the data disk

First, you'll need to attach the data disk to the virtual machine. To do this using the portal, see How to attach a managed data disk in the Azure portal.

Temporarily move pagefile.sys to C drive

1. Connect to the virtual machine.
2. Right-click the **Start** menu and select **System**.
3. In the left-hand menu, search for and select **View advanced system settings**.
4. In the **Performance** section, select **Settings**.
5. Select the **Advanced** tab.
6. In the **Virtual memory** section, select **Change**.
7. Select the **C** drive and then click **System managed size** and then click **Set**.
8. Select the **D** drive and then click **No paging file** and then click **Set**.
9. Click Apply. You will get a warning that the computer needs to be restarted for the changes to take affect.

10. Restart the virtual machine.

Change the drive letters

1. Once the VM restarts, log back on to the VM.
2. Click the **Start** menu and type **diskmgmt.msc** and hit Enter. Disk Management will start.
3. Right-click on **D**, the Temporary Storage drive, and select **Change Drive Letter and Paths**.
4. Under Drive letter, select a new drive such as **T** and then click **OK**.
5. Right-click on the data disk, and select **Change Drive Letter and Paths**.
6. Under Drive letter, select drive **D** and then click **OK**.

Move pagefile.sys back to the temporary storage drive

1. Right-click the **Start** menu and select **System**
2. In the left-hand menu, search for and select **View advanced system settings**.
3. In the **Performance** section, select **Settings**.
4. Select the **Advanced** tab.
5. In the **Virtual memory** section, select **Change**.
6. Select the OS drive **C** and click **No paging file** and then click **Set**.
7. Select the temporary storage drive **T** and then click **System managed size** and then click **Set**.
8. Click **Apply**. You will get a warning that the computer needs to be restarted for the changes to take affect.
9. Restart the virtual machine.

Next steps

- You can increase the storage available to your virtual machine by attaching an additional data disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Format and mount managed disks

09/18/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article covers how to format, mount, and persist managed disks on Azure Linux virtual machines (VMs). Managed disks are persistent storage attached to your VM that can use either SCSI or NVMe interfaces depending on your VM size.

Prerequisites

Before formatting and mounting a data disk, ensure you have:

- Identify the correct disk to avoid data loss
- SSH access to your VM
- Root or sudo privileges

> ⚠ **Warning**
>
> Always verify you're working with the correct disk before formatting. Formatting the wrong disk can result in data loss.

Format the disk

Use the latest version of `parted` available for your distribution. If the disk size is 2 tebibytes (TiB) or larger, use GPT partitioning. If the disk size is under 2 TiB, then you can use either MBR or GPT partitioning.

Once you've identified the correct disk (for example, `/dev/nvme0n2` for a data disk), you can format it:

SCSI

The following example uses `parted` on `/dev/sdc`, which is where the first data disk typically is on most VMs. Replace `sdc` with the correct option for your disk. It also formats the disk by using the XFS file system.

Bash

```
sudo parted /dev/sdc --script mklabel gpt mkpart xfspart xfs 0% 100%
sudo partprobe /dev/sdc
```

```
sudo mkfs.xfs /dev/sdc1
```

Use the partprobe utility to ensure the kernel is aware of the new partition and file system. If you don't use `partprobe`, then the `blkid` and the `lsblk` commands won't immediately return the UUID for the new file system.

Mount the disk

Now, create a directory to mount the file system by using `mkdir`. The following example creates a directory at `/datadrive`:

Bash

```
sudo mkdir /datadrive
```

Use `mount` to then mount the file system. The following example mounts the `/dev/sdc1` (for SCSI) or `/dev/nvme0n2p1` (for NVMe) partition to the `/datadrive` mount point:

SCSI

Bash

```
# Mount the disk identified earlier (replace sdc1 with your identified disk's partition)
sudo mount /dev/sdc1 /datadrive
```

You can also use the Azure device path:

Bash

```
sudo mount /dev/disk/azure/scsi1/lun0-part1 /datadrive
```

Persist the mount

To ensure that the drive is remounted automatically after a reboot, add it to the `/etc/fstab` file.

It's highly recommended that the UUID (Universally Unique Identifier) is used in `/etc/fstab` to refer to the drive rather than the device path (such as /dev/sdc1). Device paths aren't persistent

and change on reboot. To find the UUID of the new drive, use the `blkid` utility:

Bash

```
sudo blkid
```

The output looks similar to the following example:

```
/dev/sda1: LABEL="cloudimg-rootfs" UUID="11111111-1b1b-1c1c-1d1d-1e1e1e1e1e1e"
TYPE="ext4" PARTUUID="1a1b1c1d-11aa-1234-1a1a1a1a1a1a"
/dev/sda15: LABEL="UEFI" UUID="BCD7-96A6" TYPE="vfat" PARTUUID="1e1g1cg1h-11aa-
1234-1u1u1a1a1u1u"
/dev/sdb1: UUID="22222222-2b2b-2c2c-2d2d-2e2e2e2e2e2e" TYPE="ext4" TYPE="ext4"
PARTUUID="1a2b3c4d-01"
/dev/sda14: PARTUUID="2e2g2cg2h-11aa-1234-1u1u1a1a1u1u"
/dev/sdc1: UUID="33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e" TYPE="xfs"
PARTLABEL="xfspart" PARTUUID="c1c2c3c4-1234-cdef-asdf3456ghjk"
```

> **⚠ Warning**
>
> Improperly editing the `/etc/fstab` file could result in an unbootable system. If you're unsure, see your distribution's documentation for information on how to properly edit this file. You should also create a backup of the `/etc/fstab` file before editing.

Next, open the `/etc/fstab` file in a text editor. Add a line to the end of the file, by using the UUID value for the disk that was created in the previous steps, and the mountpoint of `/datadrive`. Using the example from this article, the new line would look like the following:

```
UUID=33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e   /datadrive   xfs   defaults,nofail   1
2
```

When you're done editing the file, save and close the editor.

Alternatively, you can run the following command to add the disk to the `/etc/fstab` file:

Bash

```
# For SCSI disks
UUID=$(sudo blkid -s UUID -o value /dev/sdc1)
echo "UUID=$UUID   /datadrive   xfs   defaults,nofail   1   2" | sudo tee -a
/etc/fstab
```

```
# For NVMe disks
UID=$(sudo blkid -s UUID -o value /dev/nvme1n1p1)
echo "UUID=$UUID   /datadrive   xfs   defaults,nofail   1   2" | sudo tee -a
/etc/fstab
```

> ⓘ Note
>
> Later removing a data disk without editing fstab could cause the VM to fail to boot. Most distributions provide either the `nofail` and/or `nobootwait` fstab options. These options allow a system to boot even if the disk fails to mount at boot time. See your distribution's documentation for details on these parameters.

The `nofail` option ensures that the VM starts even if the filesystem is corrupt or the disk doesn't exist at boot time. Without this option, you might encounter behavior as described in Cannot SSH to Linux VM due to FSTAB errors

The Azure VM Serial Console can be used for console access to your VM if modifying fstab results in a boot failure. More details are available in the Serial Console documentation.

TRIM/UNMAP support for Linux in Azure

Some Linux kernels support TRIM/UNMAP operations to discard unused blocks on the disk. This feature is primarily useful to inform Azure that deleted pages are no longer valid and can be discarded. This feature can save money on disks that are billed based on the amount of consumed storage, such as unmanaged standard disks and disk snapshots.

There are two ways to enable TRIM support in your Linux VM. As usual, consult your distribution for the recommended approach:

• Use the `discard` mount option in `/etc/fstab`, for example:

```
UUID=33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e   /datadrive   xfs   defaults,discard
1   2
```

• In some cases, the `discard` option might have performance implications. Alternatively, you can run the `fstrim` command manually from the command line, or add it to your crontab to run regularly:

Ubuntu

Bash

```
sudo apt install util-linux
sudo fstrim /datadrive
```

Troubleshooting

When adding data disks to a Linux VM, you may encounter errors if a disk does not exist at LUN 0. If you are adding a disk manually using the `az vm disk attach -new` command and you specify a LUN (`--lun`) rather than allowing the Azure platform to determine the appropriate LUN, take care that a disk already exists / will exist at LUN 0.

Consider the following example showing a snippet of the output from `lsscsi`:

Bash

```
[5:0:0:0]    disk    Msft     Virtual Disk     1.0   /dev/sdc
[5:0:0:1]    disk    Msft     Virtual Disk     1.0   /dev/sdd
```

The two data disks exist at LUN 0 and LUN 1 (the first column in the `lsscsi` output details `[host:channel:target:lun]`). Both disks should be accessible from within the VM. If you had manually specified the first disk to be added at LUN 1 and the second disk at LUN 2, you may not see the disks correctly from within your VM.

> **Note**
>
> The Azure `host` value is 5 in these examples, but this may vary depending on the type of storage you select.

This disk behavior is not an Azure problem, but the way in which the Linux kernel follows the SCSI specifications. When the Linux kernel scans the SCSI bus for attached devices, a device must be found at LUN 0 in order for the system to continue scanning for additional devices. As such:

- Review the output of `lsscsi` after adding a data disk to verify that you have a disk at LUN 0.
- If your disk does not show up correctly within your VM, verify a disk exists at LUN 0.

Format and mount temporary disk to Azure Linux VMs

09/18/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article covers how to format and mount temporary disks (also known as resource disks) on Azure Linux virtual machines (VMs). Depending on your VM series, temporary disks use either SCSI or NVMe interfaces. Temporary disks aren't managed disks, and aren't persistent.

Store important data on managed disks instead of local temporary disks. Temporary disks are generally meant to store items like page files, swap files, or SQL Server tempdb files.

Prerequisites

Before formatting temporary disks:

- Identify the correct disk to avoid data loss
- Understand that data isn't persistent across VM stops/deallocations
- Have SSH access to your VM with root or sudo privileges

Format disks

> ⚠ **Warning**
>
> Formatting will permanently erase all data on the disk. Ensure you're working with the correct disk and that no important data exists on it.

Use the latest version of `parted` available for your distribution. If the disk size is 2 tebibytes (TiB) or larger, you must use GPT partitioning. If the disk size is under 2 TiB, then you can use either MBR or GPT partitioning.

SCSI disks

The following example uses `parted` on `/dev/sdb`, which is typically where the SCSI temporary disks appear. Replace `sdb` with the correct device for your disk. We're using the XFS file system for better performance.

Bash

```
sudo parted /dev/disk/azure/resource --script mklabel gpt mkpart xfspart xfs
0% 100%
sudo partprobe /dev/sdb
sudo mkfs.xfs /dev/sdb1
```

Use the partprobe utility to make sure the kernel is aware of the new partition and file system. If you don't use `partprobe`, then `blkid` or `lsblk` commands might not return the UUID for the new file system immediately.

Mount temporary disks

Now, create a directory to mount the file system by using `mkdir`. For temporary storage, common mount points include `/mnt`, `/tmp`, or application-specific directories.

Bash

```
sudo mkdir /mnt/temp
```

SCSI disks

Use `mount` to mount the file system. The following example mounts the `/dev/sdb1` partition to the `/mnt/temp` mount point:

Bash

```
sudo mount /dev/sdb1 /mnt/temp
```

You can also use the Azure device path:

Bash

```
sudo mount /dev/disk/azure/resource-part1 /mnt/temp
```

TRIM/UNMAP support for temporary disks

Local temporary disks support TRIM/UNMAP operations. For optimal performance:

Use the `discard` mount option in `/etc/fstab`:

```
UUID=33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e   /mnt/temp   xfs
defaults,discard,nobootwait   0   0
```

Alternatively, run `fstrim` periodically:

Ubuntu

Bash

```
sudo apt install util-linux
sudo fstrim /mnt/temp
```

Troubleshooting

When adding data disks to a Linux VM, you may encounter errors if a disk does not exist at LUN 0. If you are adding a disk manually using the `az vm disk attach -new` command and you specify a LUN (`--lun`) rather than allowing the Azure platform to determine the appropriate LUN, take care that a disk already exists / will exist at LUN 0.

Consider the following example showing a snippet of the output from `lsscsi`:

Bash

```
[5:0:0:0]    disk    Msft     Virtual Disk     1.0   /dev/sdc
[5:0:0:1]    disk    Msft     Virtual Disk     1.0   /dev/sdd
```

The two data disks exist at LUN 0 and LUN 1 (the first column in the `lsscsi` output details `[host:channel:target:lun]`). Both disks should be accessible from within the VM. If you had manually specified the first disk to be added at LUN 1 and the second disk at LUN 2, you may not see the disks correctly from within your VM.

> **Note**
>
> The Azure `host` value is 5 in these examples, but this may vary depending on the type of storage you select.

This disk behavior is not an Azure problem, but the way in which the Linux kernel follows the SCSI specifications. When the Linux kernel scans the SCSI bus for attached devices, a device

must be found at LUN 0 in order for the system to continue scanning for additional devices. As such:

- Review the output of `lsscsi` after adding a data disk to verify that you have a disk at LUN 0.
- If your disk does not show up correctly within your VM, verify a disk exists at LUN 0.

How to detach a data disk from a Linux virtual machine

Article • 12/09/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When you no longer need a data disk that's attached to a virtual machine, you can easily detach it. This removes the disk from the virtual machine (VM), but doesn't remove it from storage. In this article, we're working with an Ubuntu LTS 16.04 distribution. If you're using a different distribution, the instructions for unmounting the disk might be different.

> ⚠ **Warning**
>
> If you detach a disk it is not automatically deleted. If you have subscribed to Premium storage, you will continue to incur storage charges for the disk. For more information, see Pricing and Billing when using Premium Storage.

If you want to use the existing data on the disk again, you can reattach it to the same virtual machine, or another one.

Connect to the VM to unmount the disk

Before you can detach the disk using either CLI or the portal, you need to unmount the disk and removed references to if from your fstab file.

Connect to the VM. In this example, the public IP address of the VM is *10.0.1.4* with the username *azureuser*:

Bash

```
ssh azureuser@10.0.1.4
```

First, find the data disk that you want to detach. The following example uses dmesg to filter on SCSI disks:

Bash

```
dmesg | grep SCSI
```

The output is similar to the following example:

Output

```
[    0.294784] SCSI subsystem initialized
[    0.573458] Block layer SCSI generic (bsg) driver version 0.4 loaded
(major 252)
[    7.110271] sd 2:0:0:0: [sda] Attached SCSI disk
[    8.079653] sd 3:0:1:0: [sdb] Attached SCSI disk
[ 1828.162306] sd 5:0:0:0: [sdc] Attached SCSI disk
```

Here, *sdc* is the disk that we want to detach. You also should grab the UUID of the disk.

Bash

```
sudo -i blkid
```

The output looks similar to the following example:

Output

```
/dev/sda1: UUID="11111111-1b1b-1c1c-1d1d-1e1e1e1e1e1e" TYPE="ext4"
/dev/sdb1: UUID="22222222-2b2b-2c2c-2d2d-2e2e2e2e2e2e" TYPE="ext4"
/dev/sdc1: UUID="33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e" TYPE="ext4"
```

Edit the */etc/fstab* file to remove references to the disk.

> **Note**
>
> Improperly editing the **/etc/fstab** file could result in an unbootable system. If unsure, refer to the distribution's documentation for information on how to properly edit this file. It is also recommended that a backup of the /etc/fstab file is created before editing.

Open the **/etc/fstab** file in a text editor and remove the line containing the UUID of your disk. Using the example values in this article, the line would look like the following:

config

```
UUID=33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e   /datadrive   ext4
defaults,nofail   1   2
```

Save and close the file when you're done.

Next, use `umount` to unmount the disk. The following example unmounts the */dev/sdc1* partition from the */datadrive* mount point:

```
Bash

sudo umount /dev/sdc1 /datadrive
```

Detach a data disk using Azure CLI

This example detaches the *myDataDisk* disk from VM named *myVM* in *myResourceGroup*.

```
Azure CLI

az vm disk detach -g myResourceGroup --vm-name myVm -n myDataDisk
```

The disk stays in storage but is no longer attached to a virtual machine.

Lower latency

In select regions, the disk detach latency has been reduced, so you see an improvement of up to 15%. This is useful if you have planned/unplanned failovers between VMs, you're scaling your workload, or are running a high scale stateful workload such as Azure Kubernetes Service. However, this improvement is limited to the explicit disk detach command, `az vm disk detach`. You won't see the performance improvement if you call a command that may implicitly perform a detach, like `az vm update`. You don't need to take any action other than calling the explicit detach command to see this improvement.

Lower latency is currently available in every public region except for:

- Canada Central
- Central US
- East US
- East US 2
- South Central US
- West US 2
- Germany North
- India West
- North Europe
- West Europe

Detach a data disk using the portal

1. In the left menu, select **Virtual Machines**.
2. In the virtual machine blade, select **Disks**.
3. In the **Disks** blade, to the far right of the data disk that you would like to detach, select the detach button, to detach the disk.
4. After the disk has been removed, select **Save** on the top of the blade.

The disk stays in storage but is no longer attached to a virtual machine. The disk isn't deleted.

Next steps

If you want to reuse the data disk, you can just attach it to another VM.

If you want to delete the disk, so that you no longer incur storage costs, see Find and delete unattached Azure managed and unmanaged disks - Azure portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Expand virtual hard disks on a Linux VM

06/19/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article covers expanding operating system (OS) disks and data disks for a Linux virtual machine (VM). You can add data disks to provide more storage space, and you can also expand an existing data disk. The default virtual hard disk size for the OS is typically 30 GB on a Linux VM in Azure. This article covers expanding either OS disks or data disks. You can't expand the size of striped volumes.

An OS disk has a maximum capacity of 4,095 GiB. However, many operating systems are partitioned with master boot record (MBR) by default. MBR limits the usable size to 2 TiB. If you need more than 2 TiB, consider attaching data disks for data storage. If you do need to store data on the OS disk and require extra space, convert it to a GUID Partition Table (GPT).

> **⚠ Warning**
>
> Always make sure that your filesystem is in a healthy state and your disk partition table type (GPT or MBR) can support the new size. Back up your data before you perform disk expansion operations. For more information, see the Azure Backup quickstart.

Identify an Azure data disk object within the operating system

When you expand a data disk that has several data disks on the VM, it might be difficult to relate the Azure logical unit numbers (LUNs) to the Linux devices. If the OS disk needs expansion, it's clearly labeled in the Azure portal as the OS disk.

Start by identifying the relationship between disk utilization, mount point, and device, with the `df` command.

Bash

```
df -Th
```

Output

```
Filesystem              Type      Size  Used Avail Use% Mounted on
/dev/sda1               xfs        97G  1.8G   95G   2% /
<truncated>
```

```
/dev/sdd1                ext4       32G   30G  727M  98% /opt/db/data
/dev/sde1                ext4       32G   49M   30G   1% /opt/db/log
```

Here you can see, for example, that the `/opt/db/data` filesystem is nearly full and is located on the `/dev/sdd1` partition. The output of `df` shows the device path whether the disk is mounted by using the device path or the (preferred) UUID in the fstab. Note the Type column, which indicates filesystem format. The format is important later.

Now locate the LUN that correlates to `/dev/sdd` by examining the contents of `/dev/disk/azure/scsi1`. The output of the following `ls` command shows that the device known as `/dev/sdd` within the Linux OS is located at `LUN1` when you look in the Azure portal.

Bash

```
sudo ls -alF /dev/disk/azure/scsi1/
```

Output

```
total 0
drwxr-xr-x. 2 root root 140 Sep  9 21:54 ./
drwxr-xr-x. 4 root root  80 Sep  9 21:48 ../
lrwxrwxrwx. 1 root root  12 Sep  9 21:48 lun0 -> ../../../sdc
lrwxrwxrwx. 1 root root  12 Sep  9 21:48 lun1 -> ../../../sdd
lrwxrwxrwx. 1 root root  13 Sep  9 21:48 lun1-part1 -> ../../../sdd1
lrwxrwxrwx. 1 root root  12 Sep  9 21:54 lun2 -> ../../../sde
lrwxrwxrwx. 1 root root  13 Sep  9 21:54 lun2-part1 -> ../../../sde1
```

Expand an Azure managed disk

Expand without downtime

You can expand your managed disks without deallocating your VM. The host cache setting of your disk doesn't change whether or not you can expand a data disk without deallocating your VM.

This feature has the following limitations.

> **ⓘ Important**
>
> This limitation doesn't apply to Premium SSD v2 or Ultra Disks.
>
> If a Standard HDD, Standard SSD, or Premium SSD disk is 4 TiB or less, deallocate your VM and detach the disk before you expand it beyond 4 TiB. If one of those disk types is

already greater than 4 TiB, you can expand it without deallocating the VM and detaching the disk.

- Is supported only for data disks.
- Isn't supported for shared disks.
- Must be installed and use one of the following options:
  - The latest Azure CLI.
  - The latest Azure PowerShell module.
  - The Azure portal.
  - An Azure Resource Manager template with an API version that's `2021-04-01` or newer.
- Isn't available on some classic VMs. Use this script to get a list of classic VM products that support expanding without downtime.

Expand with Ultra Disks and Premium SSD v2

Expanding Ultra Disks and Premium SSD v2 disks without downtime has the following additional limitations:

- You can't expand a disk while a background copy of data is also occurring on that disk. An example is when a disk is being backfilled from snapshots.
- You can expand VMs by using NVMe controllers with Ultra Disks or Premium SSD v2 disks without downtime with this public preview. Because this release is a public preview, you should use it only to test the functionality of expanding without downtime. Don't expand VMs in production.

In the following regions, you can expand VMs that are using NVMe controllers with Ultra Disks or Premium SSD v2 disks without downtime. Use either the Azure portal, the Azure CLI, or the Azure PowerShell module:

- Southeast Asia
- Brazil South
- Canada Central
- Germany West Central
- Central India (not currently supported on V6 VMs)

In the following regions, you can expand VMs that are using NVMe controllers with Ultra Disks or Premium SSD v2 disks without downtime. Use only the Azure CLI or an Azure PowerShell module. You can't currently use the Azure portal:

- East Asia
- West Central US (not currently supported on V6 VMs)

Allow up to 10 minutes for the correct size to be reflected in Windows VMs and Linux VMs. For Linux VMs, you must perform a Linux rescan function. For a Windows VM that doesn't have a workload, you must perform a Windows rescan function. You can rescan immediately, but if the time is within 10 minutes, you might need to rescan again to display the correct size.

Expand Azure managed disk

Make sure that you have the latest Azure CLI installed and are signed in to an Azure account by using az login.

This article requires an existing VM in Azure with at least one data disk attached and prepared. If you don't already have a VM that you can use, see Create and prepare a VM with data disks.

In the following samples, replace the placeholder parameter names like *myResourceGroup* and *myVM* with your own values.

> **ⓘ Important**
>
> If your disk meets the requirements in **Expand without downtime**, you can skip steps 1 and 3.

Shrinking an existing disk isn't supported and might result in data loss.

After you expand the disks, expand the volume in the OS to take advantage of the larger disk.

1. Operations on virtual hard disks can't be performed with the VM running. Deallocate your VM with az vm deallocate. The following example deallocates the VM named *myVM* in the resource group named *myResourceGroup*:

   Azure CLI

   ```
   az vm deallocate --resource-group myResourceGroup --name myVM
   ```

   The VM must be deallocated to expand the virtual hard disk. Stopping the VM with `az vm stop` doesn't release the compute resources. To release compute resources, use `az vm deallocate`.

2. View a list of managed disks in a resource group with az disk list. The following example shows a list of managed disks in the resource group named *myResourceGroup*:

   Azure CLI

```
az disk list \
    --resource-group myResourceGroup  \
    --query '[*].{Name:name,size:diskSizeGB,Tier:sku.tier}' \
    --output table
```

Expand the required disk with az disk update. The following example expands the managed disk named *myDataDisk* to *200* GB:

Azure CLI

```
az disk update \
    --resource-group myResourceGroup \
    --name myDataDisk \
    --size-gb 200
```

When you expand a managed disk, the updated size is rounded up to the nearest managed disk size.

3. Start your VM with az vm start. The following example starts the VM named *myVM* in the resource group named *myResourceGroup*:

Azure CLI

```
az vm start --resource-group myResourceGroup --name myVM
```

Expand a disk partition and filesystem

You can use many tools to perform partition resizing. The tools detailed in the remainder of this article are the same ones that certain automated processes use, such as cloud-init. As described here, the `growpart` tool with the `gdisk` package provides universal compatibility with GPT disks because older versions of some tools such as `fdisk` didn't support GPT.

Detect a changed disk size

If you used the previously mentioned procedure to expand a data disk without downtime, the reported disk size doesn't change until the device is rescanned. Rescanning normally happens only during the boot process. To call this rescan on demand, use the following procedure. When you use the methods in this article, notice that in this example the data disk is currently `/dev/sda` and was resized from 256 GiB to 512 GiB.

1. Identify the currently recognized size on the first line of output from `fdisk -l /dev/sda`:

Bash

```
sudo fdisk -l /dev/sda
```

Output

```
Disk /dev/sda: 256 GiB, 274877906944 bytes, 536870912 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x43d10aad

Device     Boot Start       End   Sectors  Size Id Type
/dev/sda1        2048 536870878 536868831  256G 83 Linux
```

2. Insert a `1` character into the rescan file for this device. Note the reference to `sda` in the example. The disk identifier changes if a different disk device is resized.

Bash

```
echo 1 | sudo tee /sys/class/block/sda/device/rescan
```

3. Verify that the new disk size is now recognized.

Bash

```
sudo fdisk -l /dev/sda
```

Output

```
Disk /dev/sda: 512 GiB, 549755813888 bytes, 1073741824 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0x43d10aad

Device     Boot Start       End   Sectors  Size Id Type
/dev/sda1        2048 536870878 536868831  256G 83 Linux
```

The remainder of this article uses the OS disk for the examples of the procedure to increase the size of a volume at the OS level. If the expanded disk is a data disk, use the previous guidance to identify the data disk device. Follow these instructions as a guideline. Substitute the data

disk device (for example, `/dev/sda`), partition numbers, volume names, mount points, and filesystem formats, as necessary.

Consider all Linux OS guidance as generic and that it might apply on any distribution, but it generally matches the conventions of the named marketplace publisher. See the Red Hat documentation for the package requirements on any distribution based on Red Hat or that claims Red Hat compatibility.

Increase the size of the OS disk

The following instructions apply to endorsed Linux distributions.

Before you proceed, make a full backup copy of your VM or, at a minimum, take a snapshot of your OS disk.

Ubuntu

On Ubuntu 16.x and newer, the root partition of the OS disk and filesystems are automatically expanded to use all free contiguous space on the root disk by cloud-init. A small amount of free space must be available for the resize operation. In this case, the sequence is to:

1. Increase the size of the OS disk as previously described.
2. Restart the VM, and then access the VM by using the **root** user account.
3. Verify that the OS disk now displays an increased filesystem size.

As shown in the following example, the OS disk was resized from the portal to 100 GB. The `/dev/sda1` filesystem mounted on `/` now displays 97 GB.

Bash

```
df -Th
```

Output

```
Filesystem     Type      Size  Used Avail Use% Mounted on
udev           devtmpfs  314M     0  314M   0% /dev
tmpfs          tmpfs      65M  2.3M   63M   4% /run
/dev/sda1      ext4       97G  1.8G   95G   2% /
tmpfs          tmpfs     324M     0  324M   0% /dev/shm
tmpfs          tmpfs     5.0M     0  5.0M   0% /run/lock
tmpfs          tmpfs     324M     0  324M   0% /sys/fs/cgroup
/dev/sda15     vfat      105M  3.6M  101M   4% /boot/efi
/dev/sdb1      ext4       20G   44M   19G   1% /mnt
```

```
tmpfs           tmpfs      65M     0   65M   0% /run/user/1000
user@ubuntu:~#
```

Expand without downtime classic VM SKU support

If you're using a classic VM SKU, it might not support expanding disks without downtime.

Use the following PowerShell script to determine which VM SKUs it's available with:

Azure PowerShell

```
Connect-AzAccount
$subscriptionId="yourSubID"
$location="desiredRegion"
Set-AzContext -Subscription $subscriptionId
$vmSizes=Get-AzComputeResourceSku -Location $location | where{$_.ResourceType -eq
'virtualMachines'}

foreach($vmSize in $vmSizes){
    foreach($capability in $vmSize.Capabilities)
    {
       if(($capability.Name -eq "EphemeralOSDiskSupported" -and $capability.Value
-eq "True") -or ($capability.Name -eq "PremiumIO" -and $capability.Value -eq
"True") -or ($capability.Name -eq "HyperVGenerations" -and $capability.Value -
match "V2"))
        {
            $vmSize.Name
       }
   }
}
```

Expand virtual hard disks attached to a Windows virtual machine

07/02/2025

Applies to: ✔ Windows VMs ✔ Flexible scale sets

When you create a new virtual machine (VM) in a resource group by deploying an image from Azure Marketplace, the default operating system (OS) disk is usually 127 GiB. (Some images have smaller OS disk sizes by default.) You can add data disks to your VM. The amount depends on the VM version that you selected.

You should install applications and CPU-intensive workloads on data disks. You might need to expand the OS disk if you're supporting a legacy application that installs components on the OS disk or if you're migrating a physical PC or VM from on-premises that has a larger OS disk. This article covers expanding either OS disks or data disks.

An OS disk has a maximum capacity of 4,095 GiB. However, many operating systems are partitioned with master boot record (MBR) by default. MBR limits the usable size to 2 TiB. If you need more than 2 TiB, create and attach data disks and use them for data storage. If you need to store data on the OS disk and require the extra space, convert it to a GUID Partition Table (GPT). To learn about the differences between MBR and GPT on Windows deployments, see Windows and GPT FAQ.

Unless you use Expand without downtime, expanding a data disk requires the VM to be deallocated. Shrinking an existing disk isn't supported and might result in data loss.

After you expand the disks, expand the volume in the OS to take advantage of the larger disk. You can't expand the size of striped volumes.

Expand without downtime

You can expand data disks without deallocating your VM. The host cache setting of your disk doesn't change whether or not you can expand a data disk without deallocating your VM.

This feature has the following limitations.

> **Important**
>
> This limitation doesn't apply to Premium SSD v2 or Ultra Disks.
>
> If a Standard HDD, Standard SSD, or Premium SSD disk is 4 TiB or less, deallocate your VM and detach the disk before you expand it beyond 4 TiB. If one of those disk types is

already greater than 4 TiB, you can expand it without deallocating the VM and detaching the disk.

- Is supported only for data disks.
- Isn't supported for shared disks.
- Must be installed and use one of the following options:
  - The latest Azure CLI.
  - The latest Azure PowerShell module.
  - The Azure portal.
  - An Azure Resource Manager template with an API version that's `2021-04-01` or newer.
- Isn't available on some classic VMs. Use this script to get a list of classic VM products that support expanding without downtime.

Expand with Ultra Disks and Premium SSD v2

Expanding Ultra Disks and Premium SSD v2 disks without downtime has the following additional limitations:

- You can't expand a disk while a background copy of data is also occurring on that disk. An example is when a disk is being backfilled from snapshots.

You can expand Ultra Disks and Premium SSD v2 disks attached to VMs using NVMe controllers without downtime in all regions that support either of those disk types, except the following:

- US East 2
- US South Central

Use either the Azure CLI or the Azure PowerShell module to make the change.

Allow up to 10 minutes for the correct size to be reflected in Windows VMs and Linux VMs. For Linux VMs, you must perform a Linux rescan function. For a Windows VM that doesn't have a workload, you must perform a Windows rescan function. You can rescan immediately, but if the time is within 10 minutes, you might need to rescan again to display the correct size. If a rescan doesn't work properly, you can either repeat the rescan or restart the VM to display the correct size.

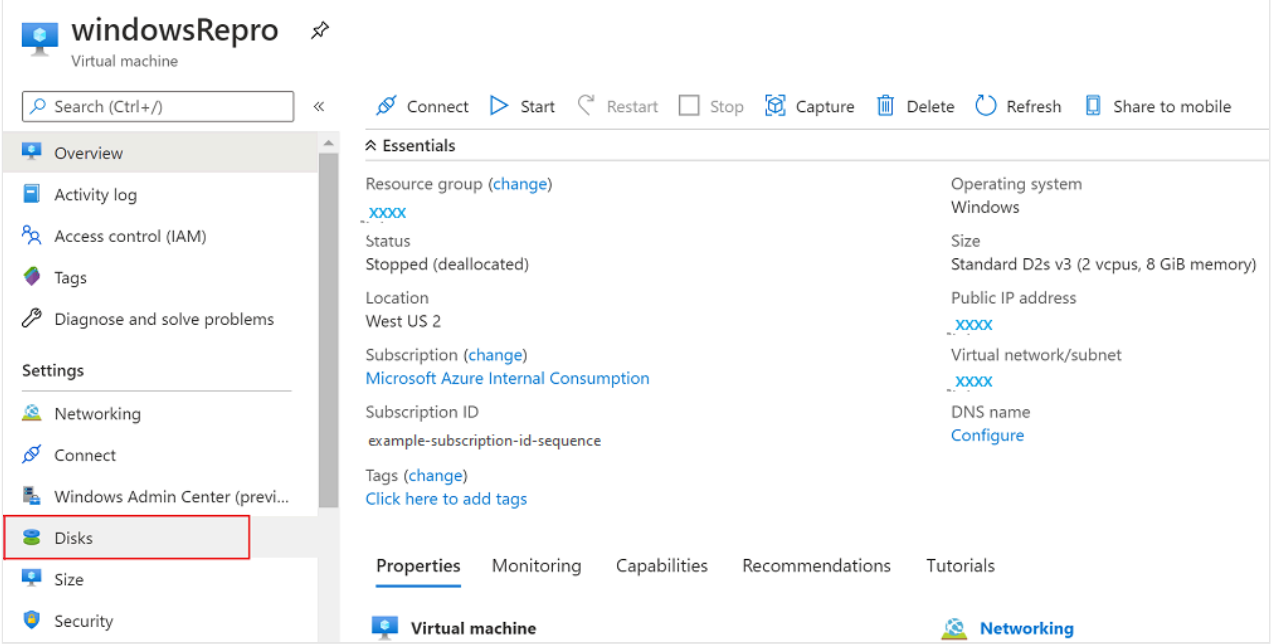
Resize a managed disk in the Azure portal

Important

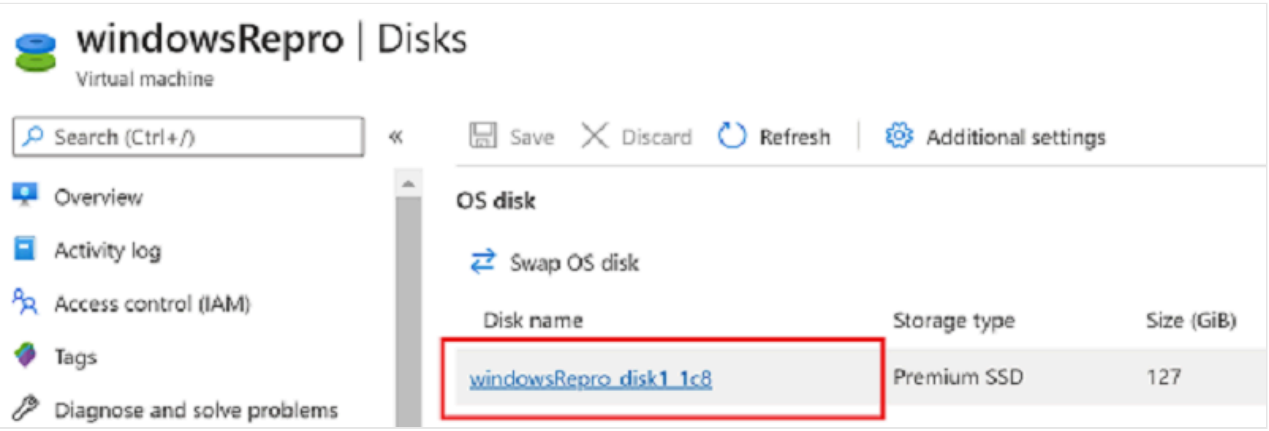
If your disk meets the requirements in Expand without downtime, you can skip step 1.

1. In the Azure portal, go to the VM in which you want to expand the disk. Select **Stop** to deallocate the VM.

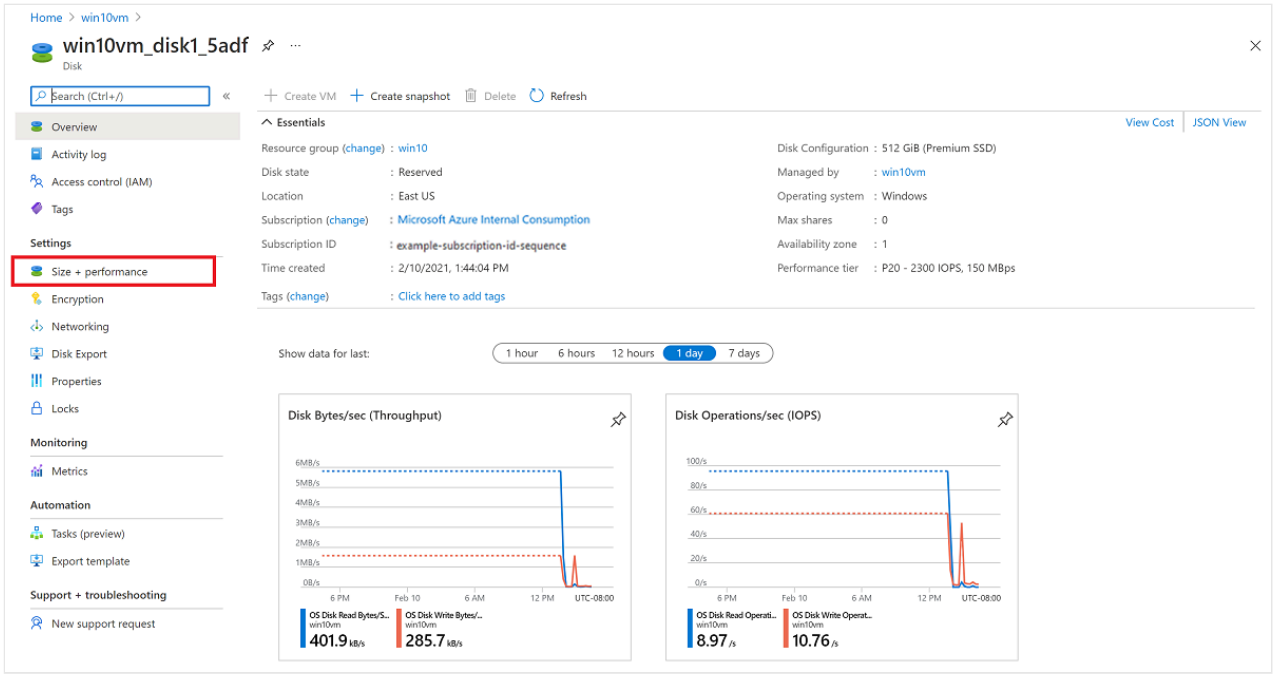
2. On the left menu, under **Settings**, select **Disks**.



3. Under **Disk name**, select the disk that you want to expand.

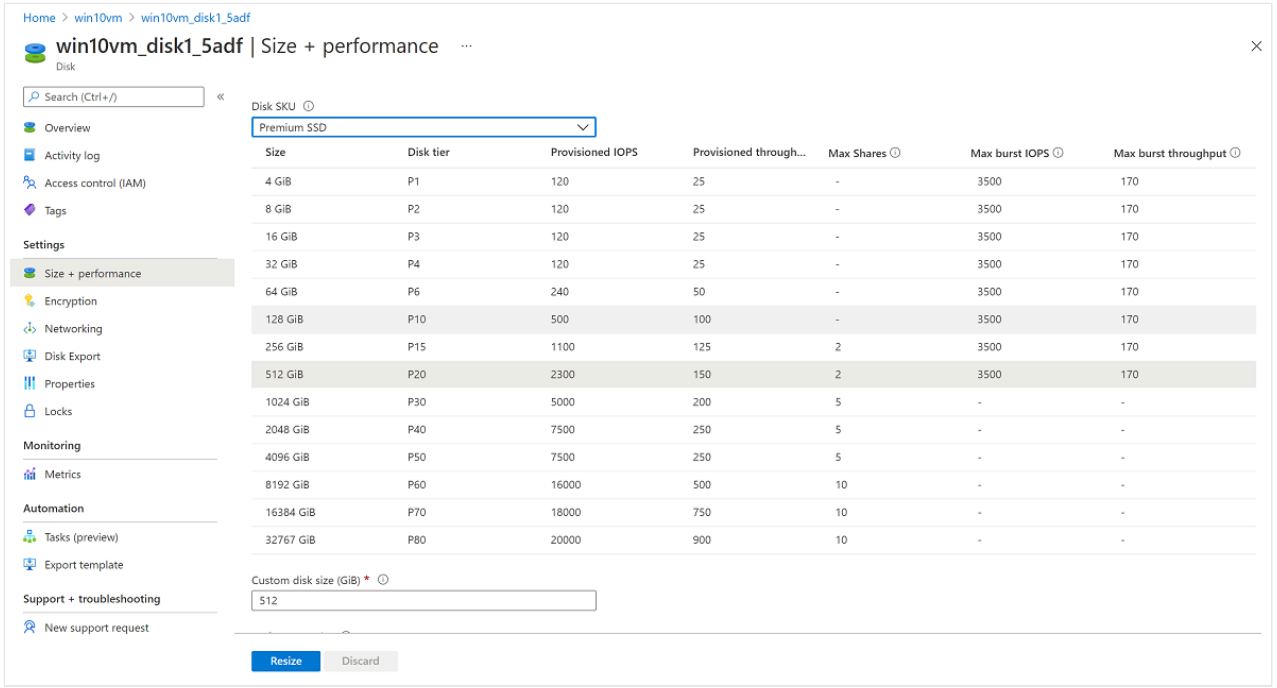


4. On the left menu, under **Settings**, select **Size + performance**.

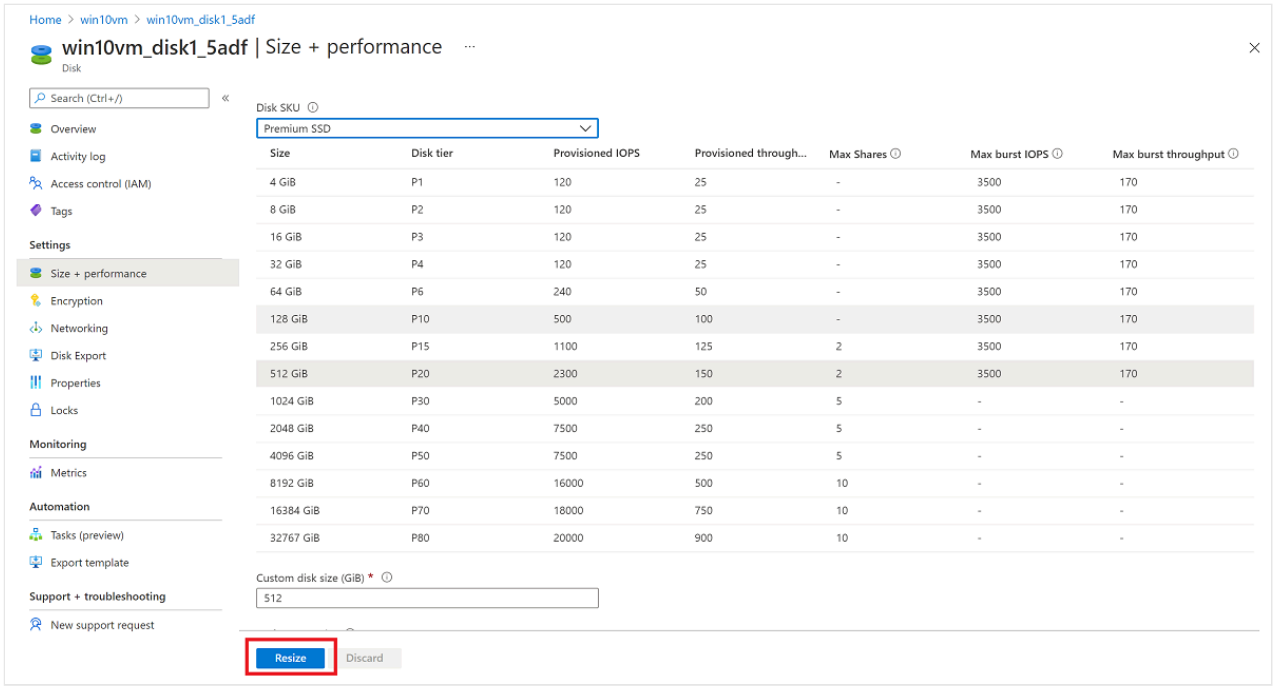


5. On **Size + performance**, select the disk size that you want.

   The new size should be greater than the existing disk size. The maximum allowed is 4,095 GB for OS disks. It's possible to expand the virtual hard disk (VHD) blob beyond that size, but the OS works only with the first 4,095 GB of space.



6. Select **Resize** at the bottom of the page.



Resize a managed disk by using PowerShell

Open a PowerShell window in administrative mode and follow these steps:

1. Sign in to your Azure account in resource management mode, and select your subscription:

   PowerShell

   ```powershell
   Connect-AzAccount
   Select-AzSubscription -SubscriptionName 'my-subscription-name'
   ```

2. Set your resource group name and VM name:

   PowerShell

   ```powershell
   $rgName = 'my-resource-group-name'
   $vmName = 'my-vm-name'
   $diskName = 'my-disk-name'
   ```

3. Obtain a reference to your VM:

   PowerShell

   ```powershell
   $vm = Get-AzVM -ResourceGroupName $rgName -Name $vmName
   ```

> **Important**

> If your disk meets the requirements in **Expand without downtime**, you can skip steps 4 and 6.

4. Stop the VM before you resize the disk:

```
PowerShell

Stop-AzVM -ResourceGroupName $rgName -Name $vmName
```

5. Obtain a reference to the managed OS disk. Set the size of the managed OS disk to the value that you want and update the disk:

```
PowerShell

$disk= Get-AzDisk -ResourceGroupName $rgName -DiskName $diskName
$disk.DiskSizeGB = 1023
Update-AzDisk -ResourceGroupName $rgName -Disk $disk -DiskName $disk.Name
```

   The new size should be greater than the existing disk size. The maximum allowed is 4,095 GB for OS disks. It's possible to expand the VHD blob beyond that size, but the OS works only with the first 4,095 GB of space.

6. Updating the VM might take a few seconds. After the command finishes running, restart the VM:

```
PowerShell

Start-AzVM -ResourceGroupName $rgName -Name $vmName
```

Remote into the VM, open **Computer Management** (or **Disk Management**), and expand the drive by using the newly allocated space.

Expand the volume in the operating system

After you expand the disk for the VM, go into the OS and expand the volume to encompass the new space. There are several methods for expanding a partition. This section covers connecting the VM by using a Remote Desktop Protocol (RDP) connection to expand the partition by using DiskPart or Disk Manager.

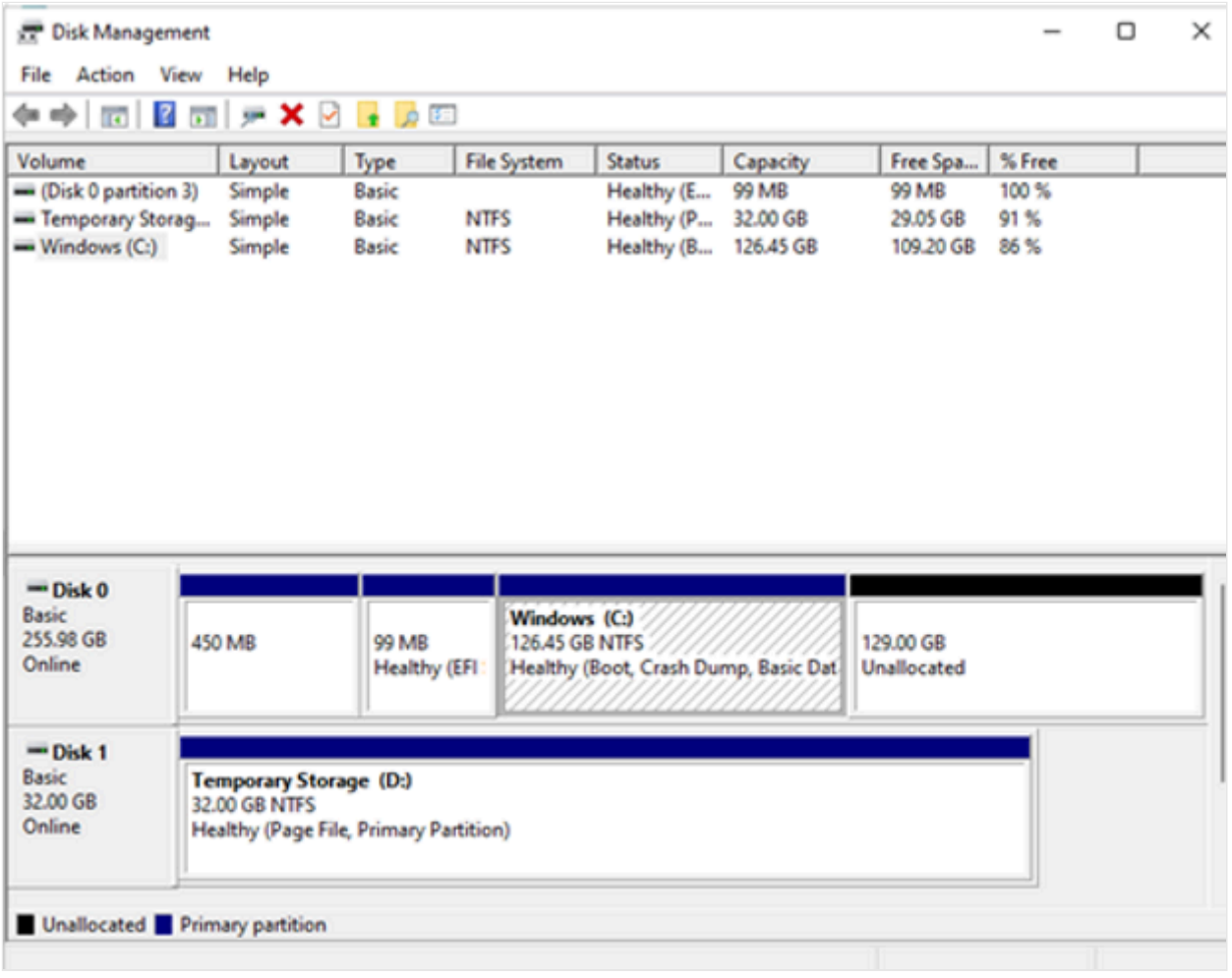
Use DiskPart

1. Open an RDP connection to your VM.

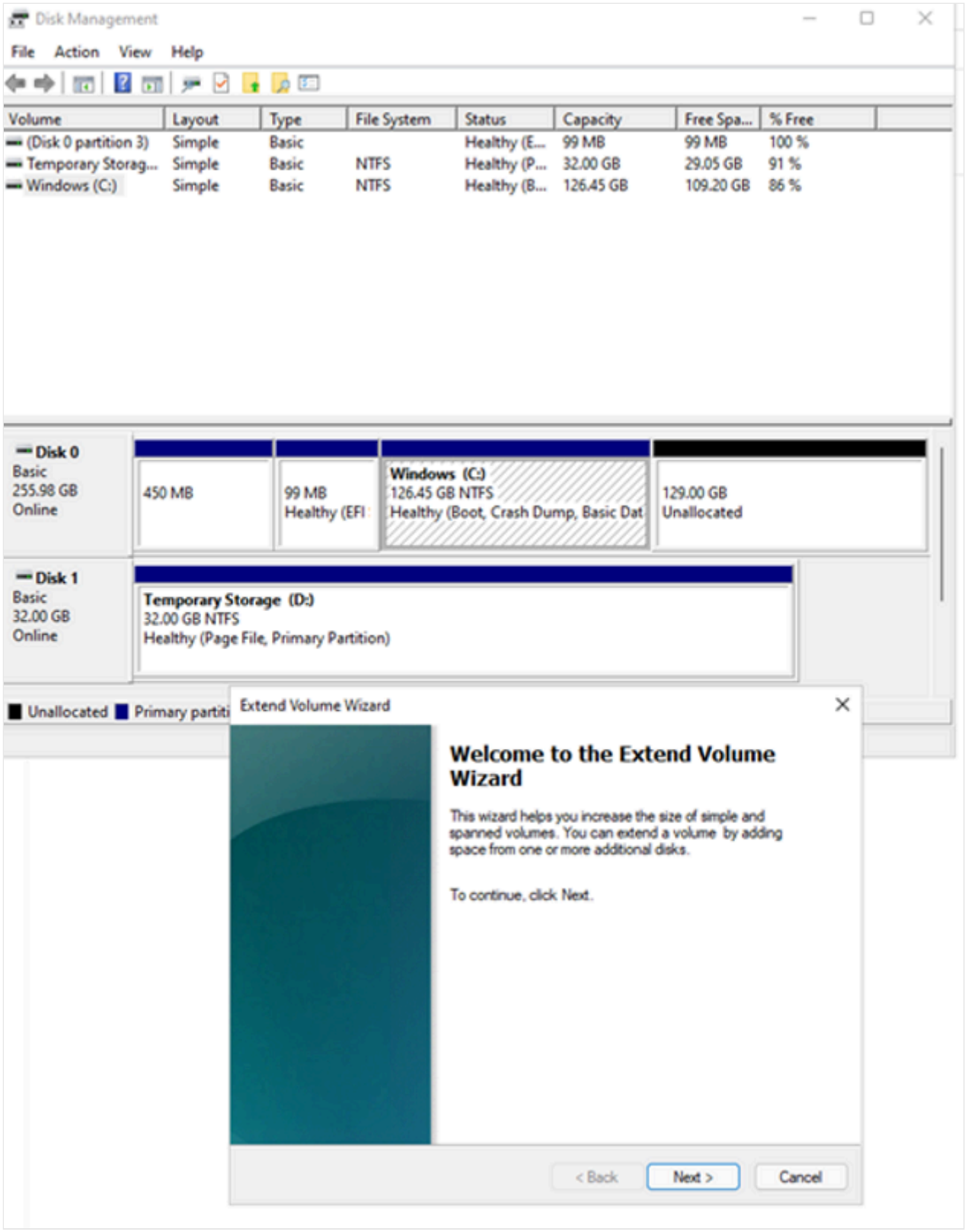
2. Open a command prompt and enter `diskpart`.
3. At the `DISKPART` prompt, enter `list volume`. Make a note of the volume that you want to extend.
4. At the `DISKPART` prompt, enter `select volume <volumenumber>`. This action selects the volume `volumenumber` that you want to extend into contiguous, empty space on the same disk.
5. At the `DISKPART` prompt, enter `extend [size=<size>]`. This action extends the selected volume by size in megabytes (MBs).

Use Disk Manager

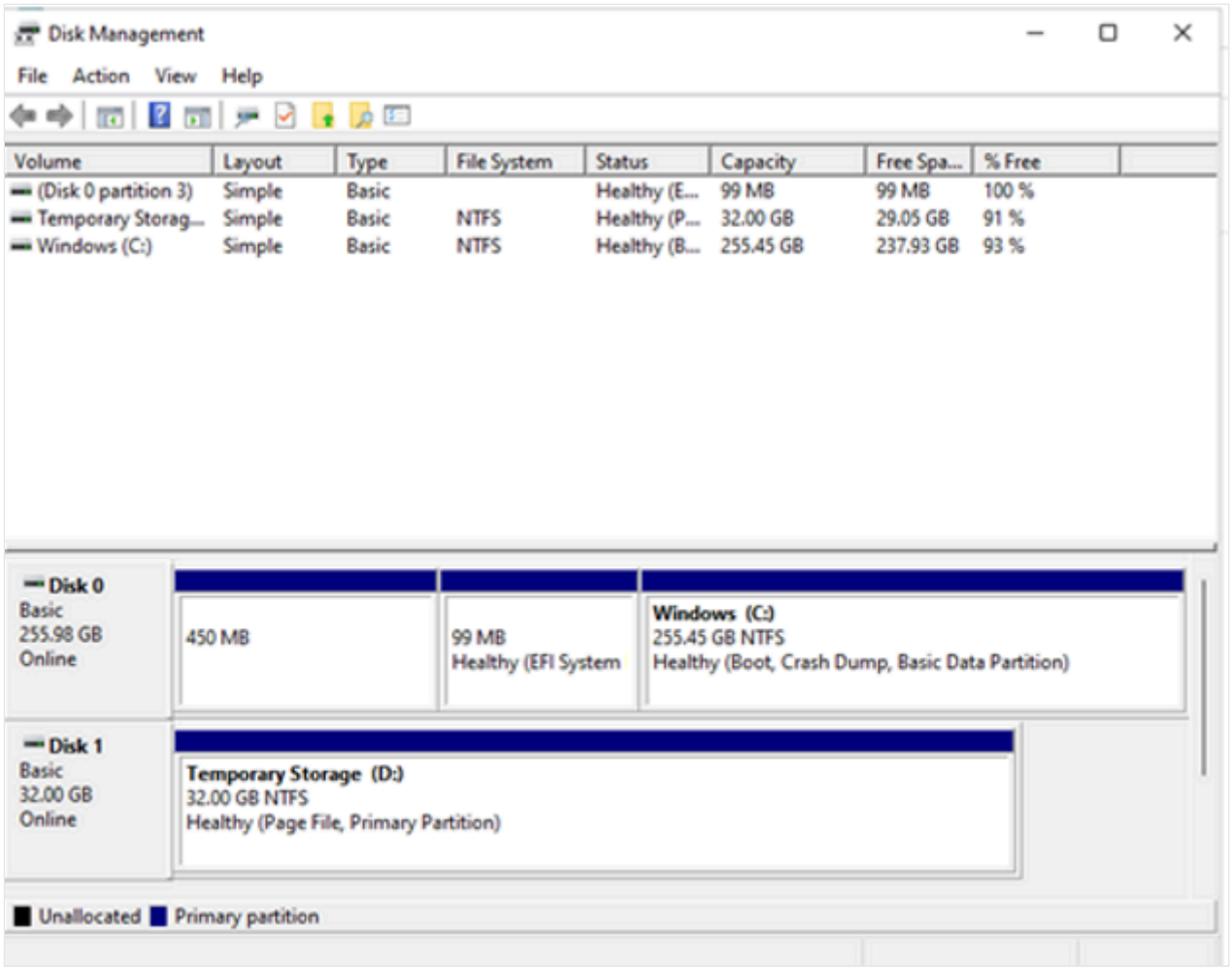
1. Start a remote desktop session with the VM.
2. Open **Disk Management**.



3. Right-click an existing **C:** drive partition and select **Extend Volume**.



4. Follow the steps in the wizard to see the disk with updated capacity.



Expand without downtime classic VM SKU support

If you're using a classic VM SKU, or version, it might not support expanding disks without downtime.

Use the following PowerShell script to determine which VM versions it's available with:

Azure PowerShell

```
Connect-AzAccount
$subscriptionId="yourSubID"
$location="desiredRegion"
Set-AzContext -Subscription $subscriptionId
$vmSizes=Get-AzComputeResourceSku -Location $location | where{$_.ResourceType -eq
'virtualMachines'}

foreach($vmSize in $vmSizes){
    foreach($capability in $vmSize.Capabilities)
    {
       if(($capability.Name -eq "EphemeralOSDiskSupported" -and $capability.Value
-eq "True") -or ($capability.Name -eq "PremiumIO" -and $capability.Value -eq
"True") -or ($capability.Name -eq "HyperVGenerations" -and $capability.Value -
match "V2"))
        {
            $vmSize.Name
```

```
        }
    }
}
```

Related content

You can also attach disks by using the Azure portal.

Expand unmanaged virtual hard disks attached to a virtual machine

Summarize this article for me

This article covers how to expand unmanaged disks. To learn how to expand a managed disk, use either the Windows or Linux articles.

Applies to: ✔ Windows VMs ✔ Linux VMs ✔ Flexible scale sets

On March 31st, 2026, unmanaged disks are being retired. If you're using unmanaged disks, migrate to managed disks before that date. For details on the retirement, see Migrate your Azure unmanaged disks by March 31st, 2026.

When you create a new virtual machine (VM) in a resource group by deploying an image from Azure Marketplace, the default operating system (OS) drive is often 127 GB. (Some images have smaller OS disk sizes by default.) It's possible to add data disks to the VM, and the number depends on the version you chose. We recommend that you install applications and CPU-intensive workloads on these addendum disks, but customers often need to expand the OS drive to support specific scenarios:

- To support legacy applications that install components on the OS drive.
- To migrate a physical PC or VM from on-premises with a larger OS drive.

Resizing an OS or data disk of an Azure VM requires the VM to be deallocated.

Shrinking an existing disk isn't supported and can potentially result in data loss.

After you expand the disks, expand the volume within the OS in either Windows or Linux to take advantage of the larger disk.

Resize an unmanaged disk by using PowerShell

Open your PowerShell Integrated Scripting Environment or PowerShell window in administrative mode and follow these steps:

- Sign in to your Azure account in resource management mode, and select your subscription:

  PowerShell

  ```
  Connect-AzAccount
  Select-AzSubscription -SubscriptionName 'my-subscription-name'
  ```

- Set your resource group name and VM names:

  PowerShell

  ```
  $rgName = 'my-resource-group-name'
  $vmName = 'my-vm-name'
  ```

- Obtain a reference to your VM:

  PowerShell

  ```
  $vm = Get-AzVM -ResourceGroupName $rgName -Name $vmName
  ```

- Stop the VM before you resize the disk:

  PowerShell

  ```
  Stop-AzVM -ResourceGroupName $rgName -Name $vmName
  ```

- Set the size of the unmanaged OS disk to the desired value and update the VM:

  PowerShell

  ```
  $vm.StorageProfile.OSDisk.DiskSizeGB = 1023
  Update-AzVM -ResourceGroupName $rgName -VM $vm
  ```

  The new size should be greater than the existing disk size. The maximum allowed is 2,048 GB for OS disks. It's possible to expand the virtual hard disk blob beyond that size, but the OS can work with only the first 2,048 GB of space.

- Update the size of any data disks that you want to resize. To expand the first data disk attached to the VM, use a numeric index to obtain a reference to the first attached data disk:

  PowerShell

  ```
  $vm.StorageProfile.DataDisks[0].DiskSizeGB = 1023
  ```

  Similarly, you can reference other data disks attached to the VM, either by using an index or the `Name` property of the disk:

  PowerShell

  ```
  ($vm.StorageProfile.DataDisks | Where ({$_.Name -eq 'my-second-data-
  ```

```
disk'})).DiskSizeGB = 1023
```

- Updating the VM might take a few seconds. When the command finishes running, restart the VM:

  PowerShell

  ```powershell
  Start-AzVM -ResourceGroupName $rgName -Name $vmName
  ```

Related content

You can also attach disks by using the Azure portal.

Last updated on 02/22/2026

Architecture best practices for Azure Disk Storage

08/13/2025

Azure managed disks are a type of Azure Disk Storage that simplify the management of storage for Azure Virtual Machines. Managed disks are block-level storage volumes that Azure manages. They're similar to physical disks in an on-premises server, but they operate in a virtual environment. When you use a managed disk, you must specify the disk size type and configure the disk. After you configure the disk, Azure manages subsequent operations and maintenance tasks.

This article assumes that as an architect, you've reviewed the storage options and chose Azure Disk Storage as the storage service for your workload. The guidance in this article provides architectural recommendations that are mapped to the principles of the Well-Architected Framework pillars.

This guide focuses on how to make decisions about Azure managed disks. But managed disks are a critical dependency of Azure Virtual Machines. As a prerequisite, read and implement the recommendations in Azure Well-Architected Framework perspective on Virtual Machines and scale sets.

Technology scope

This review focuses on the interrelated decisions for the following Azure resources:

- Azure Disk Storage

Reliability

The purpose of the Reliability pillar is to provide continued functionality by **building enough resilience and the ability to recover fast from failures**.

Reliability design principles provide a high-level design strategy applied for individual components, system flows, and the system as a whole.

Workload design checklist

Start your design strategy based on the design review checklist for Reliability. Determine its relevance to your business requirements while keeping in mind the features and capabilities of Azure Disk Storage. Extend the strategy to include more approaches as needed.

- ✓ **Review best practices to achieve high availability with managed disks.** Optimize your application for high availability by considering these recommendations and how they relate to the configuration of your managed disks and virtual machines (VMs).
- ✓ **Define reliability and recovery targets.** Review the Azure service-level agreements (SLAs). The disk types that you attach to your VM affect the VM SLA. For the highest SLA, only use Azure Ultra Disk Storage, Azure Premium SSD v2, or Premium SSD disks for OS and data disks. For guidance about calculating your reliability targets, see Recommendations for defining reliability targets.
- ✓ **Create a recovery plan.** Evaluate data-protection features, backup and restore operations, and failover procedures. Decide whether to use Azure Backup, Azure Site Recovery, or create your own backup solution by using incremental disk snapshots or restore points. A custom backup solution increases your costs.
- ✓ **Monitor potential availability problems.** Subscribe to the Azure Service Health dashboard. Use disk storage metrics in Azure Monitor to help prevent disk throttling. Manually check VMs to ensure that attached disks don't reach their storage capacity. For guidance about how to integrate these metrics into your overall workload health monitoring strategy, see Health modeling for workloads.
- ✓ **Use failure mode analysis.** Consider internal dependencies, such as the availability of virtual networks or Azure Key Vault, to help minimize points of failure.

Configuration recommendations

Expand table

Recommendation	Benefit
Distribute VMs and disks across multiple availability zones. Use a zone-redundant virtual machine scale set in flexible orchestration mode, or deploy VMs and disks across three availability zones. Use zone balancing to equally spread the instances across zones.	You provision VM and disk instances in physically separate locations within each Azure region. Each location is tolerant to local failures. Depending on resource availability, you might have an uneven number of instances across zones. Zone balancing supports availability by making sure that, if one zone is down, the other zones have sufficient instances. Two instances in each zone provide a buffer during upgrades.
Use Ultra Disk Storage, Premium SSD v2, and Premium SSD disks.	Single-instance VMs that use Premium SSD OS disks and Ultra Disk Storage, Premium SSD v2, or Premium SSD data disks have the highest uptime SLA.

Recommendation	Benefit
For maximum availability and durability, use a zone-redundant storage (ZRS) disk, especially when you share disks between VMs.	ZRS disks minimize the effect of a failure in an availability zone and increase recoverability from such zonal failures. If a zone fails and your VM remains active, workloads on ZRS disks continue to run. But if an outage does affect your VM, and you want to recover disks before the outage resolves, you can force-detach your ZRS disks from the failed VM. Then the ZRS disks can attach to a different VM. When you share a disk among multiple VMs, use a ZRS disk to prevent the shared disk from becoming a single point of failure.
Implement one of the backup options. For managed solutions, use Backup or Site Recovery. If you need to curate your own backup solution, use restore points or snapshots.	Identify the ideal backup option for your needs to help maximize your environment's recoverability.
If you manage your own snapshots, copy them across regions by using scripts.	Use scripts to simplify transferring data from one region to another. Use this option if you can't use Site Recovery. You can still create disaster recovery backups in other regions when you use this option.

Security

The purpose of the Security pillar is to provide **confidentiality, integrity, and availability** guarantees to the workload.

The Security design principles provide a high-level design strategy for achieving those goals by applying approaches to the technical design of Azure Disk Storage.

Workload design checklist

Start your design strategy based on the design review checklist for Security and identify vulnerabilities and controls to improve your security posture. Extend the strategy to include more approaches as needed.

- ✓ **Limit the ability to export or import managed disks.** Use this approach to increase the security of your data. To limit export or import capabilities, you can use one of these methods:

- Create a custom role-based access control (RBAC) role that has the permissions necessary to import and export.
- Use Microsoft Entra ID authentication.
- Set up private links.
- Configure an Azure policy.
- Configure a network access policy.

For more information, see Restrict managed disks from being imported or exported.

- ✓ **Take advantage of encryption options.** By default, managed disks are encrypted with server-side encryption (SSE), which helps you protect your data and meet organization and compliance commitments. You might need other configurations and options. You can:
  - Use SSE with encryption keys that you manage.
  - Enable encryption at host.
  - Enable double encryption at rest.

  For more information, see Server-side encryption of Azure Disk Storage.

- ✓ **Secure your shared access signature (SAS) with Microsoft Entra ID.** Microsoft Entra ID provides extra security compared to a shared key and SAS, and it's easier to use. Grant security principals only necessary permissions to perform their tasks.

- ✓ **Protect secrets.** Protect secrets, such as customer-managed keys and SAS tokens. We generally don't recommend these forms of authorization. But if you use them, make sure to rotate your keys, set key expirations as early as practical, and securely store these secrets.

- ✓ **Detect threats.** Enable Microsoft Defender for Cloud so that you can trigger security alerts when anomalies in activity occur. Defender for Cloud notifies subscription administrators by email. The email includes details about the suspicious activity and recommendations to investigate and remediate threats.

- ✓ **Use tags and labels.** Apply tags and labels to important disks to help ensure that you apply the appropriate levels of protection to the disks.

- ✓ **Harden all workload components.** Reduce extraneous surface area and tighten configurations to help reduce the likelihood of attacks. Properly secure any related resources that you use with your managed disks, such as backup recovery vaults or Azure key vaults.

Configuration recommendations

Expand table

Recommendation	Benefit
Use encryption at host for your managed disks whenever possible.	Encryption at host provides end-to-end encryption for environments that have it enabled. Encryption begins on your VM and flows through to its attached disks.
Apply an Azure Resource Manager lock on the disk.	Lock a disk to help prevent it from being deleted so that you don't lose data.
Disable traffic to the public endpoints of your disk. Create private endpoints for clients that run in Azure.	Disable traffic to the public endpoints so that traffic travels over the Microsoft backbone network, which helps eliminate exposure to the public internet.
If possible, use Azure RBAC to limit access to resources and functions.	Use RBAC to avoid tokens or keys that could be compromised. Microsoft Entra ID authenticates the security principal, such as a user, group, managed identity, or service principal. Microsoft Entra ID returns an OAuth 2.0 token. The token is used to authorize a request against the disk service.
Microsoft discourages the use of SAS tokens. If you must create one, review this list of SAS best practices before you create and distribute it. Set the expiration of SAS tokens to 60 days or less.	Best practices can help you prevent a SAS token from being leaked and quickly recover from a leak if one occurs.
Consider using your own encryption key, or a customer-managed key, to help protect the data in your managed disk.	A customer-managed key provides greater flexibility and control if you need it. For example, you can store encryption keys in Key Vault and automatically rotate them.

Cost Optimization

Cost Optimization focuses on **detecting spend patterns, prioritizing investments in critical areas, and optimizing in others** to meet the organization's budget while meeting business requirements.

The Cost Optimization design principles provide a high-level design strategy for achieving those goals and making tradeoffs as necessary in the technical design related to Azure Disk Storage.

Workload design checklist

Start your design strategy based on the design review checklist for Cost Optimization for investments. Fine-tune the design so that the workload is aligned with the budget that's

allocated for the workload. Your design should use the right Azure capabilities, monitor investments, and find opportunities to optimize over time.

- ✓ **Understand how Azure Disk Storage is billed.** Different disk types are billed in different ways and have different features that can affect billing. To design the most cost-optimized environment, see Understand Azure Disk Storage billing. For exact billing, find the specific pricing details and apply the appropriate settings. For more information, see Managed disks pricing.
- ✓ **Estimate the cost of capacity and operations.** Use the pricing calculator to model the costs that are associated with disk types, transactions, and capabilities. Compare the costs that are associated with various regions, account types, namespace types, and redundancy configurations.
- ✓ **Choose a billing model.** Evaluate whether a commitment-based model is more cost efficient than a consumption-based model. If you don't know how much capacity you need, start with a consumption-based model, monitor capacity metrics, and evaluate your choice later.
- ✓ **Decide which features you need.** Some features, such as snapshots or on-demand bursting, incur extra transaction costs, capacity costs, and other charges. For example, if you enable snapshots, you're billed for the amount of storage that each snapshot uses. When you decide which capabilities your disks need, review the pricing and billing details for those capabilities.
- ✓ **Create guardrails.** Create budgets based on subscriptions and resource groups. Use governance policies to restrict resource types, configurations, and locations. You can also use RBAC to block actions that can lead to overspending.
- ✓ **Monitor costs.** to ensure that you stay within budgets, compare costs against forecasts, and see where overspending might have occurred. Use the cost analysis feature in the Azure portal. You also can export cost data to a storage account and use Excel or Power BI to analyze that data.
- ✓ **Monitor disk resources.** Use sample scripts to search for unattached disks.

Configuration recommendations

Expand table

Recommendation	Benefit
Carefully select the appropriate disk types for your workloads. Understand the available disk	One of the best ways to reduce costs is to plan for your requirements, and use the pricing calculator to model

Recommendation	Benefit
types and their features before you deploy an environment. Then use the pricing calculator to estimate costs.	the environment.
Use reserved capacity for Premium SSD disks.	Reserved capacity for Premium SSDs reduces the total cost of your environment because you prepay for your capacity at a discount.
Assess whether the features that existing disks offer can improve performance without switching to another disk size or type. Features like disk bursting or changing performance tiers could improve performance to levels that meet your needs.	Depending on your environment and needs, enabling features to improve your disk performance can be more cost effective than switching to a different disk type. These features incur costs but might incur less costs than different disk types.
Directly adjust the performance of your Ultra Disk Storage and Premium SSD v2 disks to fit your performance needs.	These two disk types support a set number of adjustments to the disk's performance within a 24 hour period. This setting allows your workloads to be cost efficient while meeting your performance needs. You can increase performance (increase cost) to meet higher demand and then lower performance (decrease cost) when the increase is no longer needed. For example, a transaction-intensive database might need a large amount of input/output operations per second (IOPS) at a small size. Or a gaming application might need a large amount of IOPS but only during peak hours.

Operational Excellence

Operational Excellence primarily focuses on procedures for **development practices, observability, and release management**.

The Operational Excellence design principles provide a high-level design strategy for achieving those goals towards the operational requirements of the workload.

Workload design checklist

Start your design strategy based on the design review checklist for Operational Excellence for defining processes for observability, testing, and deployment related to Azure Disk Storage.

- ✓ **Create maintenance and emergency recovery plans.** Evaluate data-protection features, backup operations, and restore operations. Select backup solutions that you can use to

recover from regional disasters.

- ✓ **Create internal documentation.** Document your organization's standard practices. Incorporate existing Azure documentation to streamline your processes. Include documentation about attaching a disk to Windows or Linux VMs or expanding a disk on Windows or Linux VMs.

- ✓ **Detect threats.** Enable Defender for Cloud so that you can trigger security alerts when anomalies in activity occur. Defender for Cloud notifies subscription administrators by email. The email includes details about the suspicious activity and recommendations to investigate and remediate threats.

Configuration recommendations

Expand table

Recommendation	Benefit
Use Azure Monitor to analyze metrics and create alerts.	Azure Monitor provides insight about how your disks and VMs perform. Use these metrics to ensure that your performance remains optimal.
Review the available backup options for managed disks.	Understand the available options so that you can select the configuration that best suits your needs.

Performance Efficiency

Performance Efficiency is about **maintaining user experience even when there's an increase in load** by managing capacity. The strategy includes scaling resources, identifying and optimizing potential bottlenecks, and optimizing for peak performance.

The Performance Efficiency design principles provide a high-level design strategy for achieving those capacity goals against the expected usage.

Workload design checklist

Start your design strategy based on the design review checklist for Performance Efficiency. Define a baseline that's based on key performance indicators for Azure Disk Storage.

- ✓ **Choose optimal disk types.** Identify the disk types that you need before you deploy your resources. This approach helps you maximize performance and cost efficiency. The five disk types include Ultra Disk Storage, Premium SSD v2, Premium SSD, Azure Standard SSD,

and Azure Standard HDD. For the highest performance, use Premium SSD for your VM's OS disk, and use Ultra Disk Storage or Premium SSD v2 for your data disks.

- ✓ **Reduce the travel distance between the client and server.** Place data in regions that are closest to connecting clients, ideally in the same region. Default network configurations provide the best performance. Modify network settings only to improve security. In general, network settings don't decrease travel distance and don't improve performance.
- ✓ **Collect performance data.** Monitor your disks and VMs to identify performance bottlenecks that occur from throttling. For more information, see Storage IO metrics.
- ✓ **Benchmark your disks.** Create a test environment and determine whether it meets your needs and expectations. For more information, see Benchmark a disk.

Configuration recommendations

Expand table

Recommendation	Benefit
Create disks in the same region as the VM that you attach them to. If clients from a different region don't require the same data, create a separate disk in each region.	Reduce the physical distance between VMs and their disks, services, and on-premises clients to help improve performance and reduce network latency. This approach also reduces cost for applications that you host in Azure because bandwidth usage within a single region is free.
For workloads and solutions that require the lowest latency, such as e-commerce workloads or databases, use a Premium SSD OS disk and Ultra Disk Storage or Premium SSD v2 data disks.	This configuration offers the best reliability and highest SLA and performance.
Use Azure metrics to monitor your environment and help prevent disk throttling.	Use Azure metrics to identify disks that are being throttled and address them. Throttling leads to suboptimal performance and problems like increased latency.
For disks that are being throttled, evaluate whether changing to a larger disk size or changing to a more performant disk is better for your needs. For Premium SSD disks that are being throttled, if you have short-term bursts of demand, enable on-demand bursting.	Place applications on disks that aren't being throttled to help ensure optimal performance without increased latency.

Recommendation	Benefit
For longer-term extended demand, change the tier of the disk or evaluate whether Premium SSD v2 or Ultra Disk Storage disks better fit your needs.	
Review the expand without downtime limitations, to ensure your data disks qualify. Expanding without downtime allows you to dynamically increase your storage capacity without deallocating attached VMs. Start with smaller disk sizes and expand based on actual demand to implement just-in-time capacity provisioning strategies.	Dynamic disk expansion eliminates downtime traditionally associated with storage expansion while enabling elastic storage scaling. You can reduce upfront costs by avoiding overprovisioning and increasing the size of your disk if you need the additional capacity or performance.
When you upload a virtual hard disk (VHD), use the Add-AzVHD Azure PowerShell command.	The *Add-AzVHD* Azure PowerShell command automates most of the upload process to help streamline the process.
For existing deployments that are on-premises or in another public cloud provider, use Azure Migrate and Modernize.	Azure Migrate and Modernize can evaluate your deployment and provide curated suggestions for the best sizing of disks and VMs in a prospective Azure deployment.

Tradeoffs

You might have to make design tradeoffs if you use the approaches in the pillar checklists. Here are some examples of advantages and drawbacks.



Disk type selection and performance versus cost

- **Higher performance disk types:** Premium SSD v2 and Ultra Disk Storage provide the highest performance with customizable IOPS and throughput. These disk types offer higher availability with lower latency, making them ideal for mission-critical workloads. Premium SSDs provide built-in redundancy and zone-redundant storage options.

  However, Premium disk types can cost significantly more than Standard SSDs or HDDs. Ultra Disk Storage and Premium SSD v2 have limited regional availability and specific VM type requirements. For workloads with predictable, moderate performance needs, the additional cost might not justify the performance gains. Evaluate whether Standard SSD or even Standard HDD can meet your performance requirements at a lower cost.

- **Lower cost disk types:** Standard SSD and Standard HDD provide cost-effective storage for development, testing, and workloads with less demanding performance requirements.

These options help reduce overall infrastructure costs and are suitable for backup storage, archival data, or applications that don't require high IOPS.

These disk types have lower performance ceilings, higher latency, and reduced SLAs compared to premium options. Standard HDDs are particularly unsuitable for performance-sensitive workloads and can become bottlenecks in production environments.

Azure policies

Azure provides an extensive set of built-in policies related to Azure Disk Storage and its dependencies. Some of the preceding recommendations can be audited through Azure Policy. For example, you can check whether:

- Public network access to your managed disks is disabled.
- Backup is enabled.
- Double encryption is enabled.
- Specific disk encryption sets are used with your disks.
- Customer-managed keys are used.
- Managed disks are zone resilient.
- Notification policies for key expiration are configured.
- Autorotate for customer-managed keys is enabled.

For comprehensive governance, review the Azure Policy built-in definitions for Azure compute and other policies that might impact the security of the storage infrastructure.

Azure Advisor recommendations

Azure Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. Here are some recommendations that can help you improve the reliability, security, cost effectiveness, performance, and operational excellence of Azure Disk Storage.

- Reliability
- Security
- Cost Optimization
- Performance
- Operational Excellence

Example architecture

Foundational architecture that demonstrates the key recommendations: Azure Virtual Machines baseline architecture.

Next step

Azure Virtual Machines baseline architecture

Using disks in Azure Resource Manager Templates

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Unmanaged disks are no longer supported. If you're using unmanaged disks, migrate to managed disks. For details, see Migrate unmanaged disks to managed disks.

This document walks through the differences between managed and unmanaged disks when using Azure Resource Manager templates to provision virtual machines. The examples help you to update existing templates that are using unmanaged Disks to managed disks. For reference, we are using the vm-simple-windows template as a guide. You can see the template using both managed disks and a prior version using unmanaged disks if you'd like to directly compare them.

Unmanaged Disks template formatting

To begin, let's take a look at how unmanaged disks are deployed. When creating unmanaged disks, you need a storage account to hold the VHD files. You can create a new storage account or use one that already exists. This article shows you how to create a new storage account. Create a storage account resource in the resources block as shown below.

JSON

```
{
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2018-07-01",
    "name": "[variables('storageAccountName')]",
    "location": "[resourceGroup().location]",
    "sku": {
        "name": "Standard_LRS"
    },
    "kind": "Storage",
    "properties": {}
}
```

Within the virtual machine object, add a dependency on the storage account to ensure that it's created before the virtual machine. Within the `storageProfile` section, specify the full URI of the VHD location, which references the storage account and is needed for the OS disk and any data disks.

JSON

```
{
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2018-10-01",
    "name": "[variables('vmName')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
    "[resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))]",
    "[resourceId('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
    ],
    "properties": {
        "hardwareProfile": {...},
        "osProfile": {...},
        "storageProfile": {
            "imageReference": {
                "publisher": "MicrosoftWindowsServer",
                "offer": "WindowsServer",
                "sku": "[parameters('windowsOSVersion')]",
                "version": "latest"
            },
            "osDisk": {
                "name": "osdisk",
                "vhd": {
                    "uri": "
[concat(reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob, 'vhds/osdisk.vhd')]"
                },
                "caching": "ReadWrite",
                "createOption": "FromImage"
            },
            "dataDisks": [
                {
                    "name": "datadisk1",
                    "diskSizeGB": 1023,
                    "lun": 0,
                    "vhd": {
                        "uri": "
[concat(reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob, 'vhds/datadisk1.vhd')]"
                    },
                    "createOption": "Empty"
                }
            ]
        },
        "networkProfile": {...},
        "diagnosticsProfile": {...}
    }
}
```

Managed disks template formatting

With Azure managed disks, the disk becomes a top-level resource and no longer requires a storage account to be created by the user. Managed disks were first exposed in the `2016-04-30-preview` API version, they are available in all subsequent API versions and are now the default disk type. The following sections walk through the default settings and detail how to further customize your disks.

> ⓘ **Note**
>
> It is recommended to use an API version later than `2016-04-30-preview` as there were breaking changes between `2016-04-30-preview` and `2017-03-30`.

Default managed disk settings

To create a VM with managed disks, you no longer need to create the storage account resource. Referencing the template example below, there are some differences from the previous unmanaged disk examples to note:

- The `apiVersion` is a version that supports managed disks.
- `osDisk` and `dataDisks` no longer refer to a specific URI for the VHD.
- When deploying without specifying additional properties, the disk will use a storage type based on the size of the VM. For example, if you are using a VM size that supports premium storage (sizes with "s" in their name such as Standard_D2s_v3) then premium disks will be configured by default. You can change this by using the sku setting of the disk to specify a storage type.
- If no name for the disk is specified, it takes the format of `<VMName>_OsDisk_1_<randomstring>` for the OS disk and `<VMName>_disk<#>_<randomstring>` for each data disk.
  - If a VM is being created from a custom image then the default settings for storage account type and disk name are retrieved from the disk properties defined in the custom image resource. These can be overridden by specifying values for these in the template.
- By default, Azure disk encryption is disabled.
- By default, disk caching is Read/Write for the OS disk and None for data disks.
- In the example below there is still a storage account dependency, though this is only for storage of diagnostics and is not needed for disk storage.

JSON

```
{
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2018-10-01",
    "name": "[variables('vmName')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))]",
        "[resourceId('Microsoft.Network/networkInterfaces/',
variables('nicName'))]"
    ],
    "properties": {
        "hardwareProfile": {...},
        "osProfile": {...},
        "storageProfile": {
            "imageReference": {
                "publisher": "MicrosoftWindowsServer",
                "offer": "WindowsServer",
                "sku": "[parameters('windowsOSVersion')]",
                "version": "latest"
            },
            "osDisk": {
                "createOption": "FromImage"
            },
            "dataDisks": [
                {
                    "diskSizeGB": 1023,
                    "lun": 0,
                    "createOption": "Empty"
                }
            ]
        },
        "networkProfile": {...},
        "diagnosticsProfile": {...}
    }
}
```

Using a top-level managed disk resource

As an alternative to specifying the disk configuration in the virtual machine object, you can create a top-level disk resource and attach it as part of the virtual machine creation. For example, you can create a disk resource as follows to use as a data disk.

JSON

```
{
    "type": "Microsoft.Compute/disks",
    "apiVersion": "2018-06-01",
    "name": "[concat(variables('vmName'),'-datadisk1')]",
    "location": "[resourceGroup().location]",
    "sku": {
```

```
        "name": "Standard_LRS"
    },
    "properties": {
        "creationData": {
            "createOption": "Empty"
        },
        "diskSizeGB": 1023
    }
}
```

Within the VM object, reference the disk object to be attached. Specifying the resource ID of the managed disk created in the `managedDisk` property allows the attachment of the disk as the VM is created. The `apiVersion` for the VM resource is set to `2017-03-30`. A dependency on the disk resource is added to ensure it's successfully created before VM creation.

JSON

```
{
    "type": "Microsoft.Compute/virtualMachines",
    "apiVersion": "2018-10-01",
    "name": "[variables('vmName')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))]",
        "[resourceId('Microsoft.Network/networkInterfaces/',
variables('nicName'))]",
        "[resourceId('Microsoft.Compute/disks/', concat(variables('vmName'),'-
datadisk1'))]"
    ],
    "properties": {
        "hardwareProfile": {...},
        "osProfile": {...},
        "storageProfile": {
            "imageReference": {
                "publisher": "MicrosoftWindowsServer",
                "offer": "WindowsServer",
                "sku": "[parameters('windowsOSVersion')]",
                "version": "latest"
            },
            "osDisk": {
                "createOption": "FromImage"
            },
            "dataDisks": [
                {
                    "lun": 0,
                    "name": "[concat(variables('vmName'),'-datadisk1')]",
                    "createOption": "attach",
                    "managedDisk": {
                        "id": "[resourceId('Microsoft.Compute/disks/',
concat(variables('vmName'),'-datadisk1'))]"
                    }
```

```
                }
            ]
        },
        "networkProfile": {...},
        "diagnosticsProfile": {...}
    }
}
```

Create managed availability sets with VMs using managed disks

To create managed availability sets with VMs using managed disks, add the `sku` object to the availability set resource and set the `name` property to `Aligned`. This property ensures that the disks for each VM are sufficiently isolated from each other to avoid single points of failure. Also note that the `apiVersion` for the availability set resource is set to `2018-10-01`.

JSON

```
{
    "type": "Microsoft.Compute/availabilitySets",
    "apiVersion": "2018-10-01",
    "location": "[resourceGroup().location]",
    "name": "[variables('avSetName')]",
    "properties": {
        "PlatformUpdateDomainCount": 3,
        "PlatformFaultDomainCount": 2
    },
    "sku": {
        "name": "Aligned"
    }
}
```

Standard SSDs

Below are the parameters needed in the Resource Manager template to create Standard SSDs:

- *apiVersion* for Microsoft.Compute must be set as `2018-04-01` (or later)
- Specify *managedDisk.storageAccountType* as `StandardSSD_LRS`

The following example shows the *properties.storageProfile.osDisk* section for a VM that uses Standard SSDs:

JSON

```
"osDisk": {
    "osType": "Windows",
```

```
    "name": "myOsDisk",
    "caching": "ReadWrite",
    "createOption": "FromImage",
    "managedDisk": {
        "storageAccountType": "StandardSSD_LRS"
    }
}
```

For a complete template example of how to create a Standard SSD with a template, see Create a VM from a Windows Image with Standard SSD Data Disks.

Additional scenarios and customizations

To find full information on the REST API specifications, please review the create a managed disk REST API documentation. You will find additional scenarios, as well as default and acceptable values that can be submitted to the API through template deployments.

Next steps

- For full templates that use managed disks visit the following Azure Quickstart Repo links.
  - Windows VM with managed disk
  - Linux VM with managed disk
- Visit the Azure managed disks Overview document to learn more about managed disks.
- Review the template reference documentation for virtual machine resources by visiting the Microsoft.Compute/virtualMachines template reference document.
- Review the template reference documentation for disk resources by visiting the Microsoft.Compute/disks template reference document.
- For information on how to use managed disks in Azure virtual machine scale sets, visit the Use data disks with scale sets document.

Last updated on 04/02/2026

Use Azure Storage Explorer to manage Azure managed disks

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure Storage Explorer contains a rich set of features that allows you to:

- Upload, download, and copy managed disks.
- Create snapshots from operating system or data disk virtual hard disk.
- Migrate data from on-premises to Azure.
- Migrate data across Azure regions.

Prerequisites

To complete this article, you'll need:

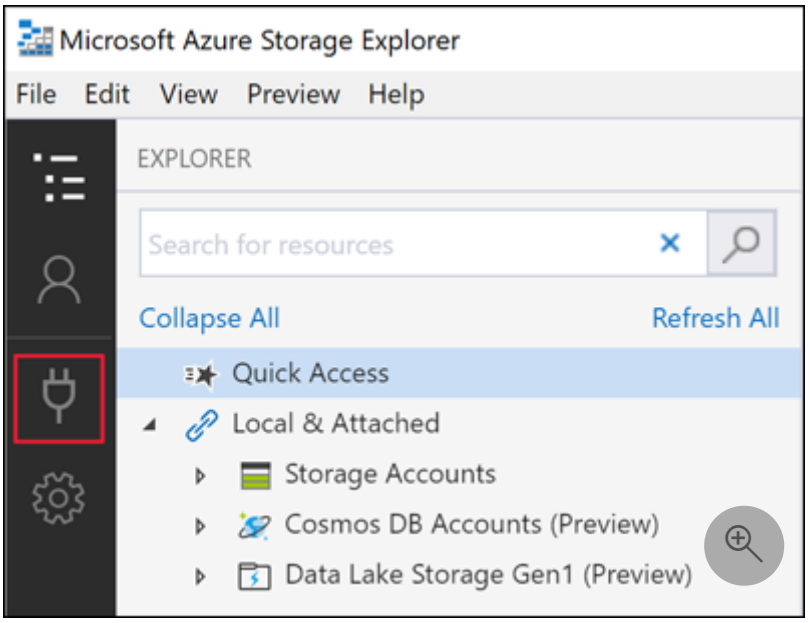
- An Azure subscription.
- At least one Azure managed disk.
- The latest version of Azure Storage Explorer.

If you don't have an Azure subscription, create a free account before you begin.

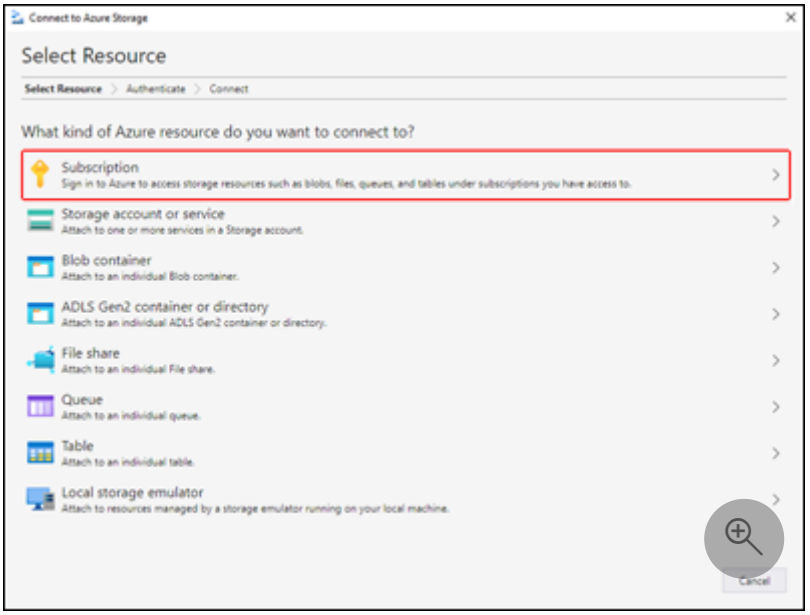
Connect to an Azure subscription

If your Storage Explorer isn't connected to Azure, you can't use it to manage resources. Follow the steps in this section to connect Storage Explorer to your Azure account. Afterward, you can use it to manage your disks.

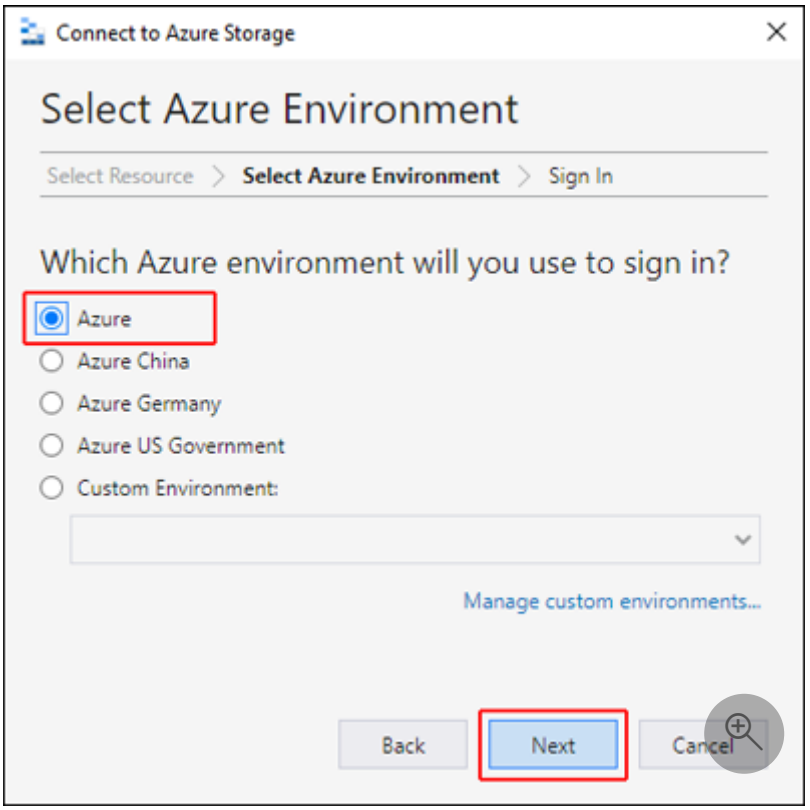
1. Open Azure Storage Explorer and select the **Connect** icon in the toolbar.



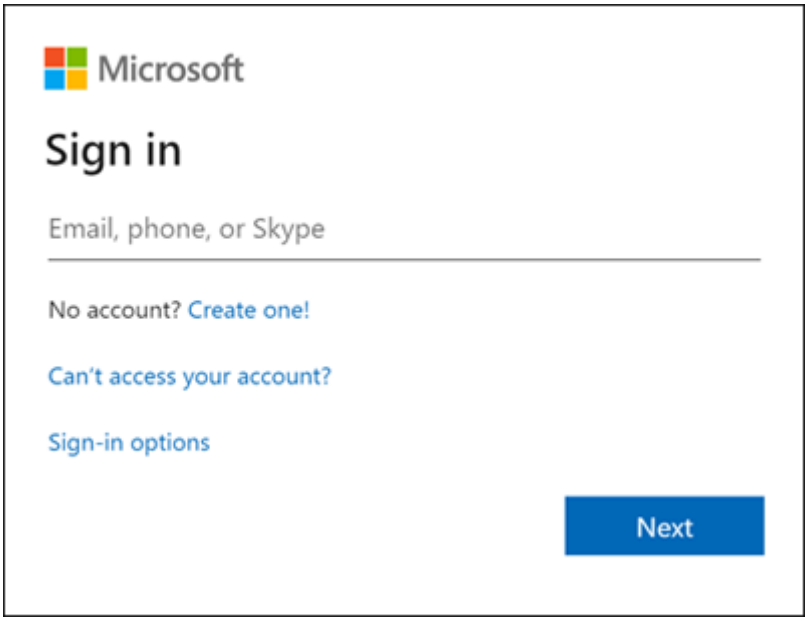
2. In the **Connect to Azure Storage** dialog box, select **Subscription**.



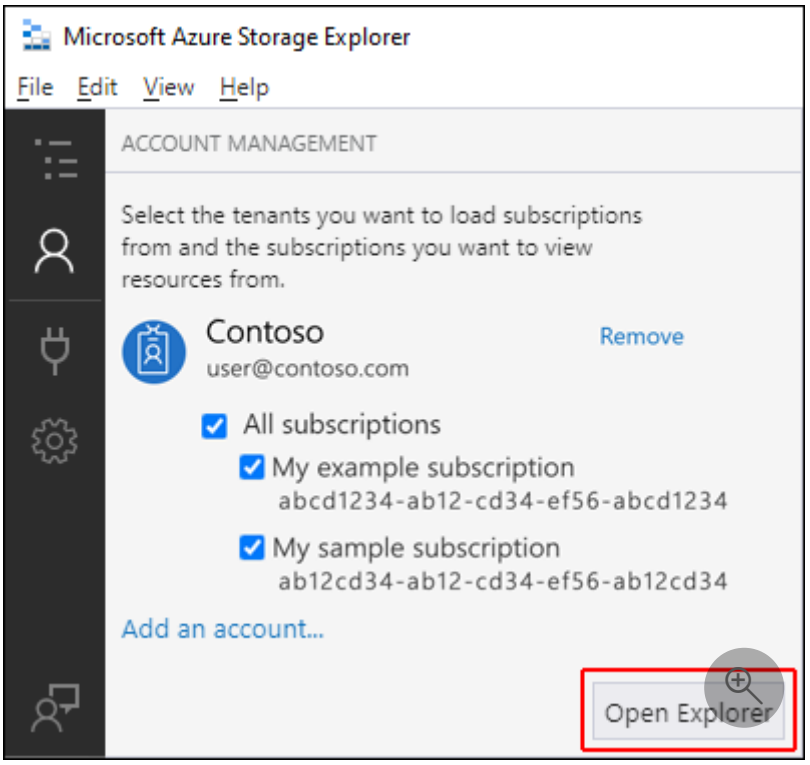
3. Select the appropriate environment and select **Next**. You can also select **Manage custom environments** to configure and add a custom environment.



4. In the **Sign in** dialog box, enter your Azure credentials.



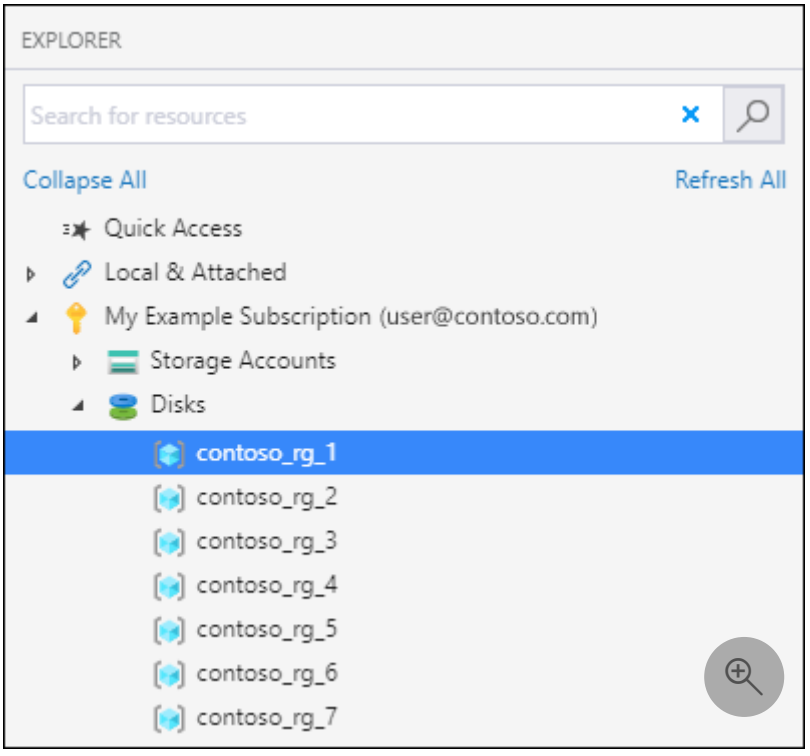
5. Select your subscription from the list and then select **Open Explorer**.



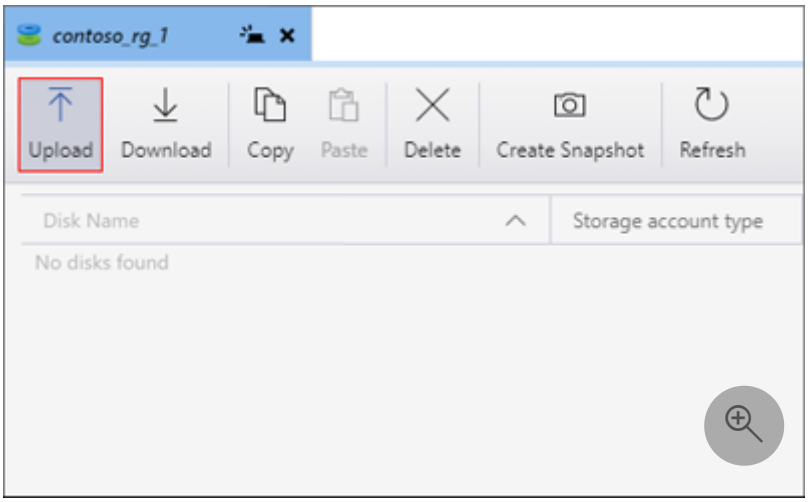
Upload an on-premises VHD

You can upload an on-premises virtual hard disk (VHD) file to Azure and use it to create an image. Follow the steps in this section to upload your source file.

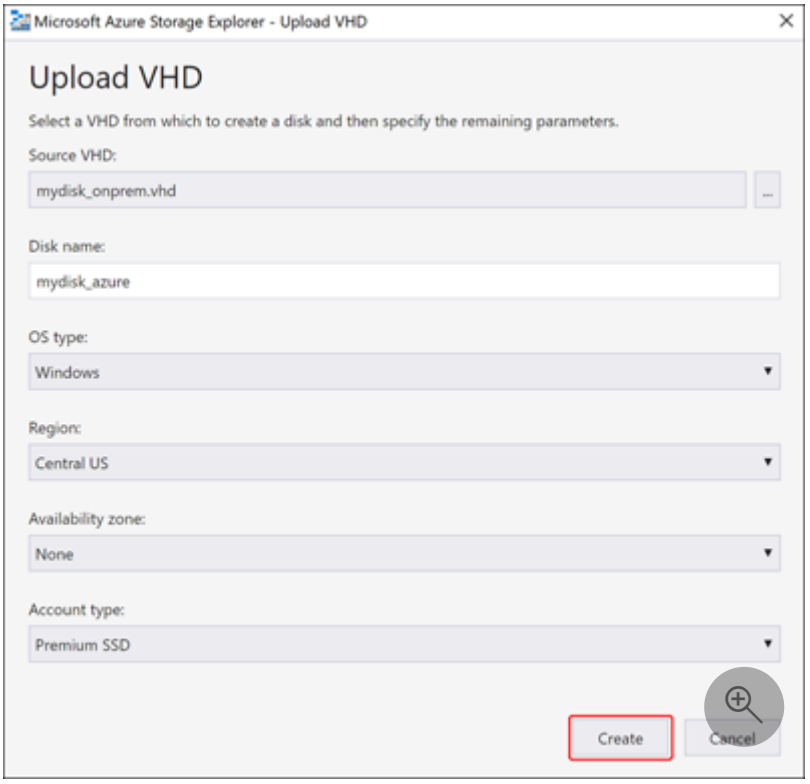
1. In the **Explorer** pane, expand **Disks** and select the resource group to which you'll upload your disk.



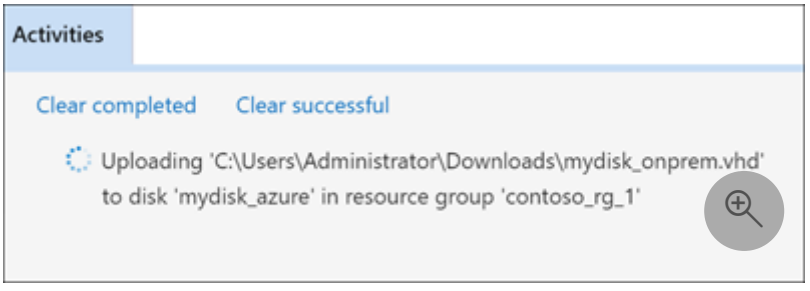
2. In the resource group details pane, select **Upload**.



3. In the **Upload VHD** dialog box, specify your VHD source file, the name of the disk, the operating system type, the region to which you want to upload the disk, and the account type. If the region supports availability zones, you can select a zone of your choice. Select **Create** to begin uploading your disk.



4. The status of the upload will now display in **Activities**.

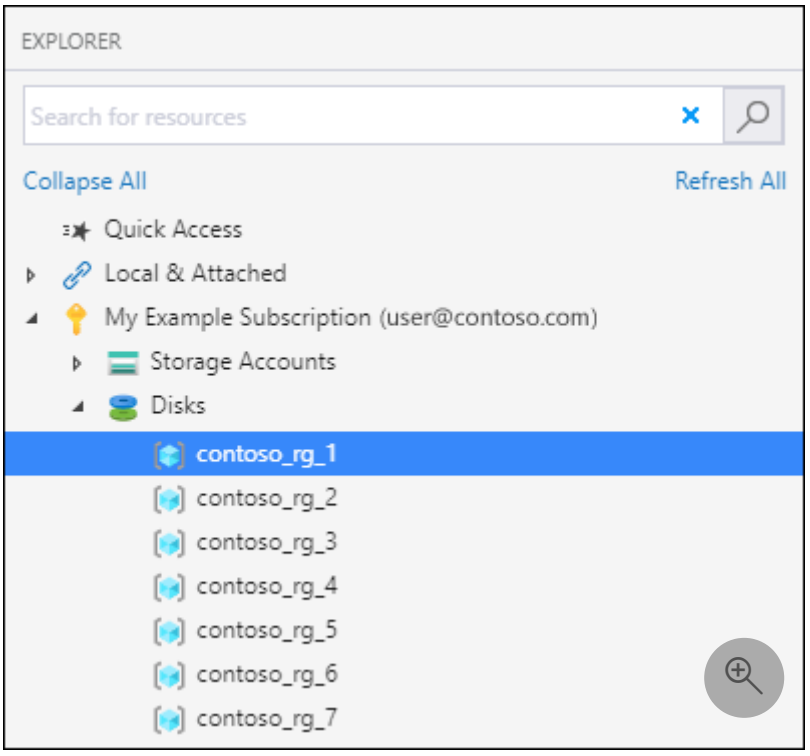


If the upload has finished and you don't see the disk in the **Activities** pane, select **Refresh**.

Download a managed disk

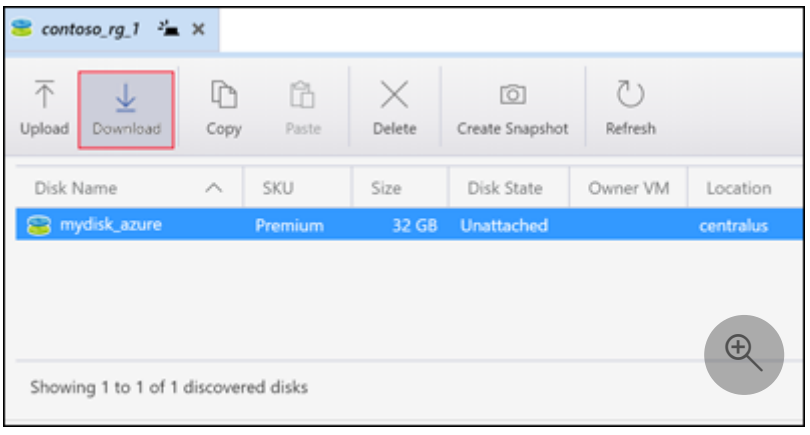
Follow the steps in this section to download a managed disk to an on-premises VHD. A disk's state must be **Unattached** before it can be downloaded.

1. In the **Explorer** pane, expand **Disks** and select the resource group from which you'll download your disk.

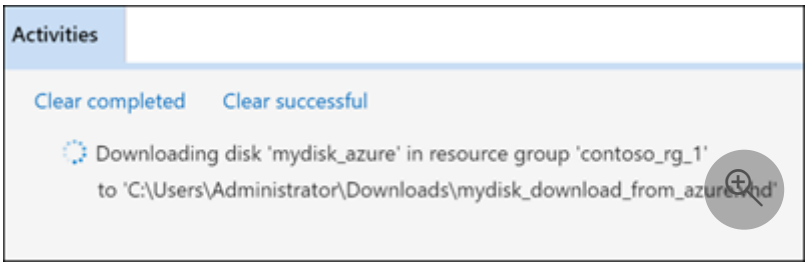


2. In the resource group details pane, select the disk you want to download.

3. Select **Download** and then choose where you would like to save the disk.



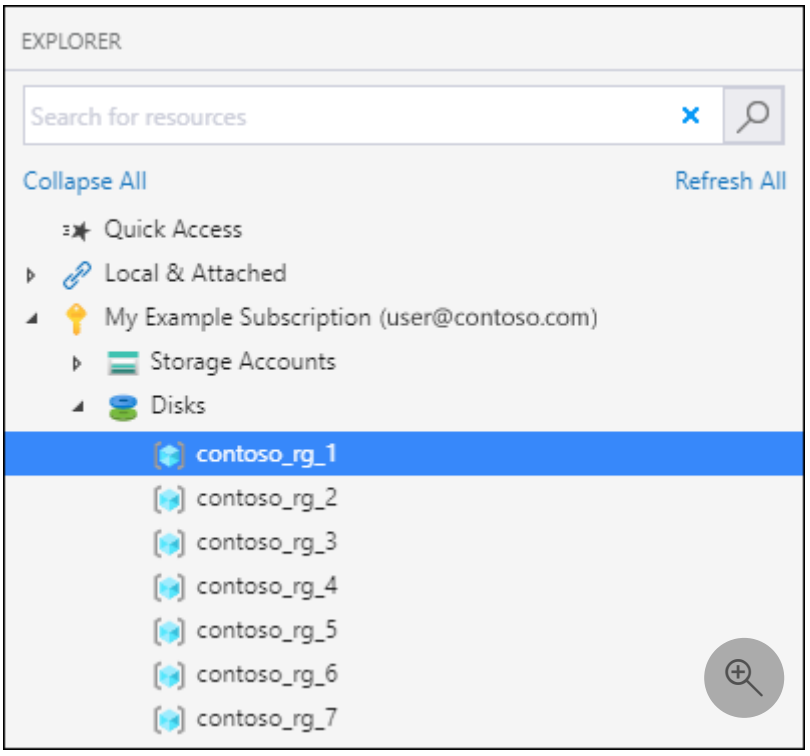
4. Select **Save** the begin the download. The download status will display in **Activities**.



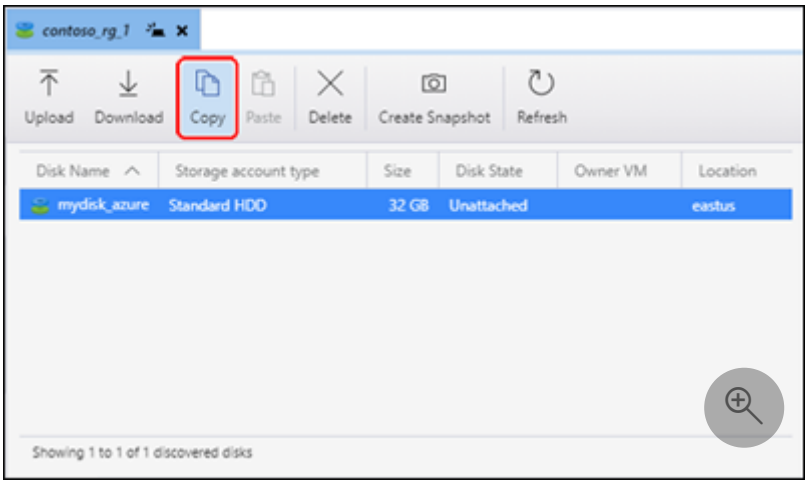
Copy a managed disk

With Storage Explorer, you can copy a manged disk within or across regions. To copy a disk:

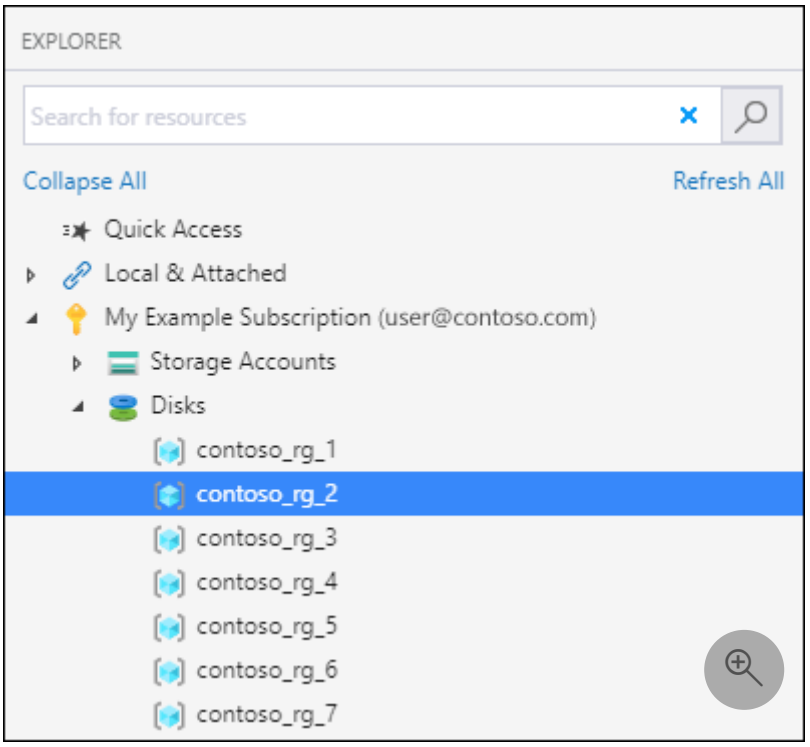
1. In the **Explorer** pane, expand the **Disks** dropdown and select the resource group that contains the disk you want to copy.



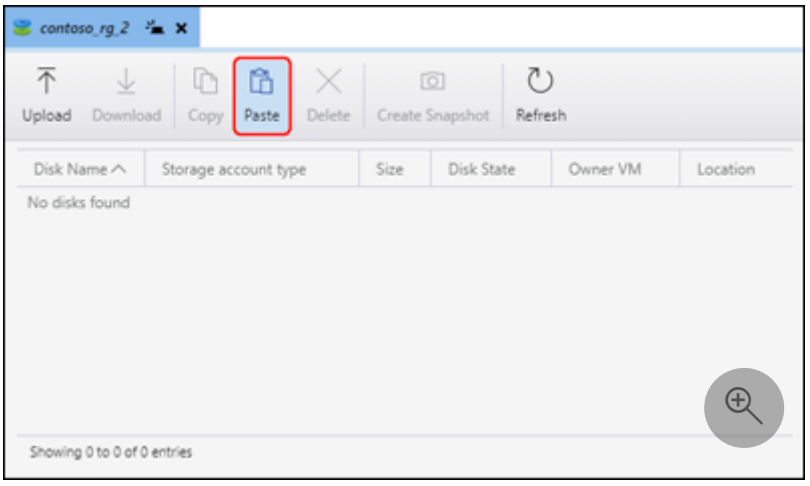
2. In the resource group details pane, select the disk you'd like to copy and select **Copy**.



3. In the **Explorer** pane, expand **Disks** and select the resource group in which you'd like to paste the disk.



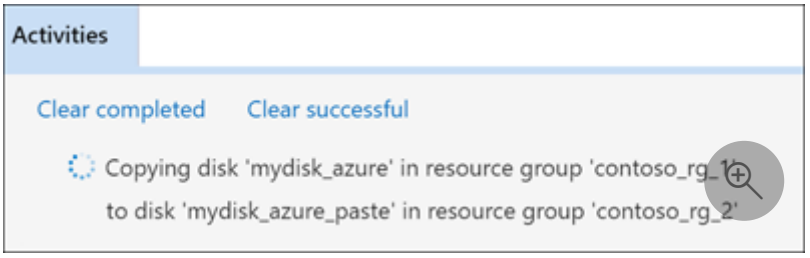
4. Select **Paste** in the resource group details pane.



5. In the **Paste Disk** dialog box, fill in the values. You can also specify an availability zone in supported regions.

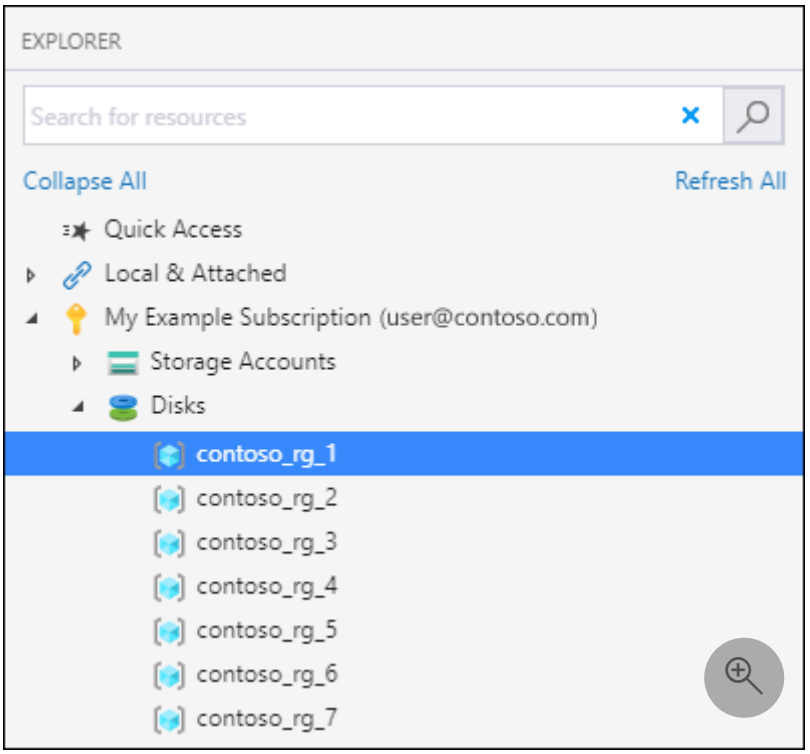


6. Select **Paste** to begin the disk copy. The status is displayed in **Activities**.

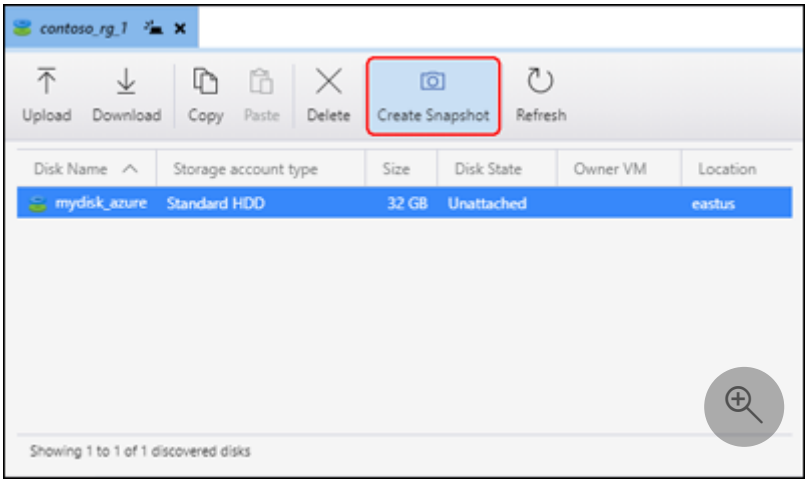


Create a snapshot

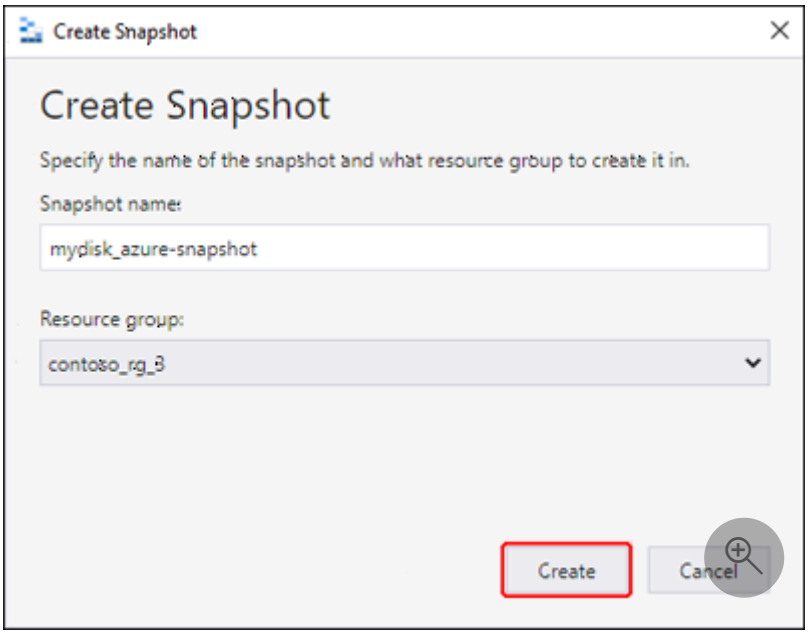
1. In the **Explorer** pane, expand **Disks** and select the resource group that contains the disk you want to snapshot.



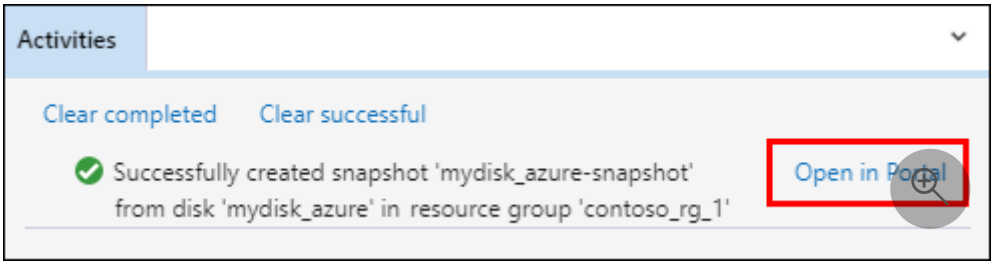
2. In the resource group details pane, select the disk you'd like to snapshot and select **Create Snapshot**.



3. In **Create Snapshot**, specify the name of the snapshot and the resource group in which you'll create it. Select **Create**.



4. After the snapshot has been created, you can select **Open in Portal** in **Activities** to view the snapshot in the Azure portal.



Next steps

- Create a virtual machine from a VHD by using the Azure portal
- Attach a managed data disk to a Windows virtual machine by using the Azure portal

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Change the OS disk used by an Azure VM using the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

If you have an existing VM, but you want to swap the disk for a backup disk or another OS disk, you can use the Azure CLI to swap the OS disks. You don't have to delete and recreate the VM. You can even use a managed disk in another resource group, as long as it isn't already in use.

The VM does not need to be stopped\deallocated. The resource ID of the managed disk can be replaced with the resource ID of a different managed disk.

Make sure that the VM size and storage type are compatible with the disk you want to attach. For example, if the disk you want to use is in Premium Storage, then the VM needs to be capable of Premium Storage (like a DS-series size). Both disks must also be the same size. And ensure that you're not mixing an un-encrypted VM with an encrypted OS disk, this is not supported. If the VM doesn't use Azure Disk Encryption, then the OS disk being swapped in shouldn't be using Azure Disk Encryption. If disks are using Disk Encryption Sets, both disks should belong to same Disk Encryption set.

This article requires Azure CLI version 2.0.25 or greater. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Use az disk list to get a list of the disks in your resource group.

Azure CLI

```
az disk list \
   -g myResourceGroupDisk \
   --query '[*].{diskId:id}' \
   --output table
```

(Optional) Use az vm stop to stop\deallocate the VM before swapping the disks.

Azure CLI

```
az vm stop \
   -n myVM \
   -g myResourceGroup
```

Use az vm update with the full resource ID of the new disk for the `--osdisk` parameter

Azure CLI

```
az vm update \
   -g myResourceGroup \
   -n myVM \
   --os-disk /subscriptions/<subscription ID>/resourceGroups/<resource
group>/providers/Microsoft.Compute/disks/myDisk
```

Restart the VM using az vm start.

Azure CLI

```
az vm start \
   -n myVM \
   -g myResourceGroup
```

Next steps

To create a copy of a disk, see Snapshot a disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Change the OS disk used by an Azure VM using PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

If you have an existing VM, but you want to swap the disk for a backup disk or another OS disk, you can use Azure PowerShell to swap the OS disks. You don't have to delete and recreate the VM. You can even use a managed disk in another resource group, as long as it isn't already in use.

The VM does not need to be stopped\deallocated. The resource ID of the managed disk can be replaced with the resource ID of a different managed disk.

Make sure that the VM size and storage type are compatible with the disk you want to attach. For example, if the disk you want to use is in Premium Storage, then the VM needs to be capable of Premium Storage (like a DS-series size). Both disks must also be the same size. And ensure that you're not mixing an un-encrypted VM with an encrypted OS disk, this is not supported. If the VM doesn't use Azure Disk Encryption, then the OS disk being swapped in shouldn't be using Azure Disk Encryption. If disks are using Disk Encryption Sets, both disks should belong to same Disk Encryption set.

Get a list of disks in a resource group using Get-AzDisk

Azure PowerShell

```
Get-AzDisk -ResourceGroupName myResourceGroup | Format-Table -Property Name
```

When you have the name of the disk that you would like to use, set that as the OS disk for the VM. This example stop\deallocates the VM named *myVM* and assigns the disk named *newDisk* as the new OS disk.

Azure PowerShell

```
# Get the VM
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM

# (Optional) Stop/ deallocate the VM
Stop-AzVM -ResourceGroupName myResourceGroup -Name $vm.Name -Force

# Get the new disk that you want to swap in
$disk = Get-AzDisk -ResourceGroupName myResourceGroup -Name newDisk

# Set the VM configuration to point to the new disk
```

```
Set-AzVMOSDisk -VM $vm -ManagedDiskId $disk.Id -Name $disk.Name

# Update the VM with the new OS disk
Update-AzVM -ResourceGroupName myResourceGroup -VM $vm

# Start the VM
Start-AzVM -Name $vm.Name -ResourceGroupName myResourceGroup
```

Next steps

To create a copy of a disk, see Snapshot a disk.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to map Azure Disks to Linux VM guest disks

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

You may need to determine the Azure Disks that back a VM's guest disks. In some scenarios, you can compare the disk or volume size to the size of the attached Azure Disks. In scenarios where there are multiple Azure Disks of the same size attached to the VM you need to use the Logical Unit Number (LUN) of the data disks.

What is a LUN?

A Logical Unit Number (LUN) is a number that is used to identify a specific storage device. Each storage device is assigned a unique numeric identifier, starting at zero. The full path to a device is represented by the bus number, target ID number, and Logical Unit Number (LUN).

For example: ***Bus Number 0, Target ID 0, LUN 3***

For our exercise, you only need to use the LUN.

Finding the LUN

Below we have listed two methods for finding the LUN of a disk in Linux.

lsscsi

1. Connect to the VM
2. `sudo lsscsi`

The first column listed will contain the LUN, the format is [Host:Channel:Target:**LUN**].

Listing block devices

1. Connect to the VM
2. `sudo ls -l /sys/block/*/device`

The last column listed will contain the LUN, the format is [Host:Channel:Target:**LUN**]

Finding the LUN for the Azure Disks

You can locate the LUN for an Azure Disk using the Azure portal, Azure CLI.

Finding an Azure Disk's LUN in the Azure portal

1. In the Azure portal, select "Virtual Machines" to display a list of your Virtual Machines
2. Select the Virtual Machine
3. Select "Disks"
4. Select a data disk from the list of attached disks.
5. The LUN of the disk will be displayed in the disk detail pane. The LUN displayed here correlate to the LUNs that you looked up in the Guest using lsscsi, or listing the block devices.

Finding an Azure Disk's LUN using Azure CLI

Azure CLI

```
az vm show -g myResourceGroup -n myVM --query "storageProfile.dataDisks"
```

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to map Azure Disks to Windows VM guest disks

Article • 08/22/2024

Applies to: ✔ Windows VMs

You may need to determine the Azure Disks that back a VM's guest disks. In some scenarios, you can compare the disk or volume size to the size of the attached Azure Disks. In scenarios where there are multiple Azure Disks of the same size attached to the VM you need to use the Logical Unit Number (LUN) of the data disks.

What is a LUN?

A Logical Unit Number (LUN) is a number that is used to identify a specific storage device. Each storage device is assigned a unique numeric identifier, starting at zero. The full path to a device is represented by the bus number, target ID number, and Logical Unit Number (LUN).

For example: ***Bus Number 0, Target ID 0, LUN 3***

For our exercise, you only need to use the LUN.

Finding the LUN

There are two methods to finding the LUN, which one you choose will depend on if you are using Storage Spaces or not.

Disk Management

If you are not using Storage Pools, you can use Disk Management to find the LUN.

1. Connect to the VM and open Disk Management a. Right-click on the Start button and choose "Disk Management" a. You can also type `diskmgmt.msc` into the Start Search box
2. In the lower pane, right-click any of the Disks and choose "Properties"
3. The LUN will be listed in the "Location" property on the "General" tab

Storage Pools

1. Connect to the VM and open Server Manager
2. Select "File and Storage Services", "Volumes", "Storage Pools"
3. In the bottom-right corner of Server Manager, there will be a "Physical Disks" section. The disks that make up the Storage Pool are listed here as well as the LUN for each disk.

Finding the LUN for the Azure Disks

You can locate the LUN for an Azure Disk using the Azure portal, Azure CLI, or Azure PowerShell

Finding an Azure Disk's LUN in the Azure portal

1. In the Azure portal, select "Virtual Machines" to display a list of your Virtual Machines
2. Select the Virtual Machine
3. Select "Disks"
4. Select a data disk from the list of attached disks.
5. The LUN of the disk will be displayed in the disk detail pane. The LUN displayed here correlates to the LUNs that were looked up in the Guest using Device Manager or Server Manager.

Finding an Azure Disk's LUN using Azure CLI or Azure PowerShell

Azure CLI

```
az vm show -g myResourceGroup -n myVM --query "storageProfile.dataDisks"
```

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Learn about Azure-VM-Utils

09/30/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets

The azure-vm-utils package provides essential utilities and udev rules to optimize the Linux experience on Azure Virtual Machines. This package consolidates device management tools for Small Computer System Interface (SCSI), Non-Volatile Memory Express (NVMe), Microsoft Azure Network Adapter (MANA), and Mellanox devices, making disk identification and management more reliable and consistent across different VM configurations.

NVMe udev rules

Newer virtual machines (VMs) SKUs on Azure use the NVMe interface for disk management. VMs with NVMe interfaces interpret and present disks differently from VMs that use SCSI interfaces. For details, see SCSI to NVMe conversion.

NVMe udev rules in this package consolidate critical tools and udev rules to create stable, predictable symlinks for Azure disks. This package provides an easy and reliable way to identify disks, making automation, troubleshooting, and management simpler.

Symlinks

WALinuxAgent currently includes udev rules to provide several symlinks for SCSI disks:

- `/dev/disk/azure/resource`
- `/dev/disk/azure/root`
- `/dev/disk/azure/scsi0/lun<lun>`
- `/dev/disk/azure/scsi1/lun<lun>`

The rules found in WALinuxAgent are being extended with azure-vm-utils to add identification support for NVMe devices.

The following symlinks are provided for all instances with NVMe disks:

- `/dev/disk/azure/data/by-lun/<lun>`
- `/dev/disk/azure/local/by-serial/<serial>`
- `/dev/disk/azure/os`

For v6 and newer VM sizes with local NVMe disks that support namespace identifiers, the following links are also available:

- `/dev/disk/azure/local/by-index/<index>`
- `/dev/disk/azure/local/by-name/<name>`

SCSI compatibility

To ensure backward compatibility for disks using SCSI controllers, azure-vm-utils supports the following links:

- `/dev/disk/azure/os`
- `/dev/disk/azure/resource`

> **Note**
>
> Some VM sizes come with both a NVMe temporary disk in addition to a SCSI temporary disk.

Linux distribution support

The following distros and versions currently include az-vm-utils in their official Azure marketplace images:

Expand table

Distribution	Version
Fedora	42
Flatcar	4152.2.3
Azure Linux	2.0

Installation

If the package isn't present in the default platform image, install it via package managers or from the GitHub repository.

Manual installation

For distributions where azure-vm-utils isn't preinstalled, build and install it manually:

```
Bash
```

```
# Clone the repository
git clone https://github.com/Azure/azure-vm-utils.git
cd azure-vm-utils

# Build the package
cmake .
make

# Install (requires root privileges)
sudo make install
```

Utilities

azure-nvme-id

The `azure-nvme-id` utility helps identify Azure NVMe devices and their properties. This utility is useful for troubleshooting and scripting.

To run the utility:

Bash

```
sudo azure-nvme-id
```

To run in udev mode (typically used by udev rules):

Bash

```
DEVNAME=/dev/nvme0n1 azure-nvme-id --udev
```

Using the symlinks

After azure-vm-utils is installed, you can use the predictable symlinks for disk operations instead of relying on device names that might change between reboots.

Examples

List all Azure disk symlinks:

Bash

```
find /dev/disk/azure/ -type l
```

Access the OS disk:

```Bash
ls -la /dev/disk/azure/os
```

Access data disks by LUN:

```Bash
ls -la /dev/disk/azure/data/by-lun/
```

Access local NVMe disks by serial number:

```Bash
ls -la /dev/disk/azure/local/by-serial/
```

Verification

To verify that azure-vm-utils is working correctly on your VM:

1. Check if the package is installed:

   ```Bash
   # For RPM-based systems
   rpm -qa azure-vm-utils

   # For DEB-based systems
   dpkg -l azure-vm-utils
   ```

2. Verify udev rules are in place:

   ```Bash
   ls -al /usr/lib/udev/rules.d/10-azure-unmanaged-sriov.rules
   ls -al /usr/lib/udev/rules.d/80-azure-disk.rules
   ```

3. Check for Azure disk symlinks:

Bash

```
find /dev/disk/azure -type l
```

Use SCP to move files to and from a VM

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This article shows how to move files from your workstation up to an Azure VM, or from an Azure VM down to your workstation, using Secure Copy (SCP). Moving files between your workstation and a VM, quickly and securely, is critical for managing your Azure infrastructure.

For this article, you need a VM deployed in Azure with SSH enabled. You also need an SCP client for your local computer. It's built on top of SSH and included in the default shell of most Linux and Windows (10 and newer) installations.

Quick commands

Upload a file to the VM

```Bash
scp file azureuser@azurehost:directory/targetfile
```

Download a file from the VM

```Bash
scp azureuser@azurehost:directory/file targetfile
```

Detailed walkthrough

As examples, we move an Azure configuration file up to a VM and pull down a log file directory, both using SCP.

SSH key pair authentication

SCP uses SSH for the transport layer. SSH handles the authentication on the destination host, and it moves the file in an encrypted tunnel provided by default with SSH. For SSH authentication, usernames and passwords can be used. However, SSH public and private key authentication are recommended as a security best practice. Once SSH has authenticated the connection, SCP then begins copying the file. When you use a

properly configured `~/.ssh/config` and SSH public and private keys, the SCP connection can be established by just using a server name (or IP address). If you only have one SSH key, SCP looks for it in the `~/.ssh/` directory, and uses it by default to log in to the VM.

For more information on configuring your `~/.ssh/config` and SSH public and private keys, see Create SSH keys.

Upload a file to a VM

For the first example, we copy an Azure configuration file up to a VM that is used to deploy automation. Because this file contains Azure API credentials, which include secrets, security is important. The encrypted tunnel provided by SSH protects the contents of the file.

The following command copies the local *.azure/config* file to an Azure VM with FQDN *myserver.eastus.cloudapp.azure.com*. If you don't have an FQDN set, you can also use the IP address of the VM. The admin user name on the Azure VM is *azureuser*. The file is targeted to the */home/azureuser/* directory. Substitute your own values in this command.

Bash

```
scp ~/.azure/config
azureuser@myserver.eastus.cloudapp.com:/home/azureuser/config
```

Download a directory from a VM

For this example, we copy a directory of log files from the VM down to your workstation. A log file may or may not contain sensitive or secret data. However, using SCP ensures the contents of the log files are encrypted. A log directory may contain too many relevant files to copy one at a time, so downloading the whole directory is preferred in this situation. Using SCP to transfer the files is the easiest way to get the log directory and files down to your workstation while also being secure.

The following command copies files in the */home/azureuser/logs/* directory on the Azure VM to the local /tmp directory:

Bash

```
scp -r azureuser@myserver.eastus.cloudapp.com:/home/azureuser/logs/. /tmp/
```

The `-r` flag instructs SCP to recursively copy the files and directories from the point of the directory listed in the command. Also notice that the command-line syntax is similar to a `cp` copy command.

Next steps

- Manage users, SSH, and check or repair disks on Azure Linux VMs using the 'VMAccess' Extension

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Find and delete unattached Azure managed and unmanaged disks by using the Azure CLI

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Flexible scale sets

When you delete a virtual machine (VM) in Azure, by default, any disks that are attached to the VM aren't deleted. This feature helps to prevent data loss due to the unintentional deletion of VMs. After a VM is deleted, you will continue to pay for unattached disks. This article shows you how to find and delete any unattached disks and reduce unnecessary costs.

> **Note**
>
> Use the az disk show command to get the `LastOwnershipUpdateTime` for any disk. This property shows when the disk's state was last updated. For an unattached disk, this property shows the time when the disk was unattached. This property is blank for newly created disks, until their state changes.

Managed disks: Find and delete unattached disks

The following script looks for unattached managed disks by examining the value of the **ManagedBy** property. When you attach a managed disk to a VM, the **ManagedBy** property contains the resource ID of the VM. When a managed disk is unattached, the **ManagedBy** property is null. The script examines all the managed disks in an Azure subscription. When the script locates a managed disk with the **ManagedBy** property set to null, the script determines that the disk is unattached.

> **Important**
>
> First, run the script by setting the **deleteUnattachedDisks** variable to 0. This action lets you find and view all the unattached managed disks.
>
> After you review all the unattached disks, run the script again and set the **deleteUnattachedDisks** variable to 1. This action lets you delete all the unattached managed disks.

Azure CLI

```
# Set deleteUnattachedDisks=1 if you want to delete unattached Managed Disks
# Set deleteUnattachedDisks=0 if you want to see the Id of the unattached Managed
Disks
deleteUnattachedDisks=0
unattachedDiskIds=$(az disk list --query '[?managedBy==`null`].[id]' -o tsv)
for id in ${unattachedDiskIds[@]}
do
    if (( $deleteUnattachedDisks == 1 ))
    then

        echo "Deleting unattached Managed Disk with Id: "$id
        az disk delete --ids $id --yes
        echo "Deleted unattached Managed Disk with Id: "$id

    else
        echo $id
    fi
done
```

Unmanaged disks: Find and delete unattached disks

Unmanaged disks are VHD files that are stored as page blobs in Azure storage accounts. The following script looks for unattached unmanaged disks (page blobs) by checking the value of the **LeaseStatus** property. When you attach an unmanaged disk to a VM, the **LeaseStatus** property is **Locked**. When an unmanaged disk is unattached, the **LeaseStatus** property is **Unlocked**. The script checks all the unmanaged disks in all the Azure storage accounts in an Azure subscription. When the script finds an unmanaged disk with a **LeaseStatus** property set to **Unlocked**, the script determines that the disk is unattached.

> **ⓘ Important**
>
> First, run the script by setting the **deleteUnattachedVHDs** variable to 0. This lets you find and view all the unattached unmanaged VHDs.
>
> After you review all the unattached disks, run the script again and set the **deleteUnattachedVHDs** variable to 1. This deletes all your unattached unmanaged VHDs.

Azure CLI

```
# Set deleteUnattachedVHDs=1 if you want to delete unattached VHDs
# Set deleteUnattachedVHDs=0 if you want to see the details of the unattached VHDs
deleteUnattachedVHDs=0
storageAccountIds=$(az storage account list --query [].[id] -o tsv)
for id in ${storageAccountIds[@]}
```

```
do
    connectionString=$(az storage account show-connection-string --ids $id --query
connectionString -o tsv)
    containers=$(az storage container list --connection-string $connectionString --
query [].[name] -o tsv)

    for container in ${containers[@]}
    do

        blobs=$(az storage blob list --show-next-marker -c $container --connection-
string $connectionString --query "[?properties.blobType=='PageBlob' &&
ends_with(name,'.vhd')].[name]" -o tsv)

        for blob in ${blobs[@]}
        do
            leaseStatus=$(az storage blob show -n $blob -c $container --connection-
string $connectionString --query "properties.lease.status" -o tsv)

            if [ "$leaseStatus" == "unlocked" ]
            then

                if (( $deleteUnattachedVHDs == 1 ))
                then

                    echo "Deleting VHD: "$blob" in container: "$container" in
storage account: "$id

                    az storage blob delete --delete-snapshots include  -n $blob -c
$container --connection-string $connectionString

                    echo "Deleted VHD: "$blob" in container: "$container" in
storage account: "$id
                else
                    echo "StorageAccountId: "$id" container: "$container" VHD:
"$blob
                fi

            fi
        done
    done
done
```

Next steps

For more information, see Delete a storage account.

Last updated on 02/18/2026

Find and delete unattached Azure managed and unmanaged disks by using Azure PowerShell

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

When you delete a virtual machine (VM) in Azure, by default, any disks that are attached to the VM aren't deleted. This feature helps to prevent data loss due to the unintentional deletion of VMs. After a VM is deleted, you will continue to pay for unattached disks. This article shows you how to find and delete any unattached disks and reduce unnecessary costs.

> **Note**
>
> You can use the `Get-AzureDisk` command to get the `LastOwnershipUpdateTime` for any disk. This property represents when the disk's state was last updated. For an unattached disk, this value shows the time when the disk was unattached. This property is blank for newly created disks, until their state changes.

Managed disks: Find and delete unattached disks

The following script looks for unattached managed disks by examining the value of the **ManagedBy** property. When you attach a managed disk to a VM, the **ManagedBy** property contains the resource ID of the VM. When a managed disk is unattached, the **ManagedBy** property is null. The script examines all the managed disks in an Azure subscription. When the script locates a managed disk with the **ManagedBy** property set to null, the script determines that the disk is unattached.

> **Important**
>
> First, run the script by setting the **deleteUnattachedDisks** variable to 0. This action lets you find and view all the unattached managed disks.
>
> After you review all the unattached disks, run the script again and set the **deleteUnattachedDisks** variable to 1. This action deletes all the unattached managed disks.

Azure PowerShell

```
# Set deleteUnattachedDisks=1 if you want to delete unattached Managed Disks
# Set deleteUnattachedDisks=0 if you want to see the Id of the unattached Managed
Disks
$deleteUnattachedDisks=0
$managedDisks = Get-AzDisk
foreach ($md in $managedDisks) {
    # ManagedBy property stores the Id of the VM to which Managed Disk is attached
to
    # If ManagedBy property is $null then it means that the Managed Disk is not
attached to a VM
    if($md.ManagedBy -eq $null){
        if($deleteUnattachedDisks -eq 1){
            Write-Host "Deleting unattached Managed Disk with Id: $($md.Id)"
            $md | Remove-AzDisk -Force
            Write-Host "Deleted unattached Managed Disk with Id: $($md.Id) "
        }else{
            $md.Id
        }
    }
 }
```

Unmanaged disks: Find and delete unattached disks

On March 31st, 2026, unmanaged disks are being retired. If you're using unmanaged disks, migrate to managed disks before that date. For details on the retirement, see Migrate your Azure unmanaged disks by March 31st, 2026.

Unmanaged disks are VHD files that are stored as page blobs in Azure storage accounts. The following script looks for unattached unmanaged disks (page blobs) by examining the value of the **LeaseStatus** property. When an unmanaged disk is attached to a VM, the **LeaseStatus** property is set to **Locked**. When an unmanaged disk is unattached, the **LeaseStatus** property is set to **Unlocked**. The script examines all the unmanaged disks in all the Azure storage accounts in an Azure subscription. When the script locates an unmanaged disk with a **LeaseStatus** property set to **Unlocked**, the script determines that the disk is unattached.

> **ⓘ Important**
>
> The script has **deleteUnattachedVHDs** variable set to `$false` by default. This setting lets you find and view all the unattached unmanaged VHDs.
>
> After you review all the unattached disks, if you want to delete all those VHDs, set the **deleteUnattachedVHDs** variable to `$true` and run the script again. This action deletes all unattached VHDs.

Azure PowerShell

```
# Set deleteUnattachedVHDs=$true if you want to delete unattached VHDs
# Set deleteUnattachedVHDs=$false if you want to see the Uri of the unattached VHDs
$deleteUnattachedVHDs = $false
$storageAccounts = Get-AzStorageAccount
foreach ($storageAccount in $storageAccounts) {
    $storageKey = (Get-AzStorageAccountKey -ResourceGroupName
$storageAccount.ResourceGroupName -Name $storageAccount.StorageAccountName)
[0].Value

    $context = New-AzStorageContext -StorageAccountName
$storageAccount.StorageAccountName -StorageAccountKey $storageKey

    $containers = Get-AzStorageContainer -Context $context

    foreach ($container in $containers) {
        $blobs = Get-AzStorageBlob -Container $container.Name -Context $context
        #Fetch all the Page blobs with extension .vhd as only Page blobs can be
attached as disk to Azure VMs
        $blobs | Where-Object {$_.BlobType -eq 'PageBlob' -and
$_.Name.EndsWith('.vhd')} | ForEach-Object {
            if ($_.BlobProperties.LeaseStatus -eq 'Unlocked') {
                if ($deleteUnattachedVHDs) {
                    Write-Host "Deleting unattached VHD with Uri:
$($_.ICloudBlob.Uri.AbsoluteUri)"
                    Remove-AzStorageBlob -Blob $_.Name -Container $container.Name -
Context $context -Force
                    Write-Host "Deleted unattached VHD with Uri:
$($_.ICloudBlob.Uri.AbsoluteUri)"
                } else {
                    Write-Host "Unattached VHD Uri:
$($_.ICloudBlob.Uri.AbsoluteUri)"
                }
            }
        }
    }
}
```

Next steps

For more information, see Delete a storage account and Identify Orphaned Disks Using PowerShell.

Last updated on 02/22/2026

Frequently asked questions about Azure IaaS VM disks and managed and unmanaged premium disks

This article answers some frequently asked questions about Azure managed disks and Azure Premium SSDs.

Managed disks

If I create a standard managed disk from an existing VHD that's 80 GiB, how much will that cost me?

A standard managed disk created from an 80-GiB VHD is treated as the next available standard disk size, which is an S10 disk. You're charged according to the S10 disk pricing. For more information, see the pricing page.

For a standard managed disk, will I be charged for the actual size of the data on the disk or for the provisioned capacity of the disk?

You're charged based on the provisioned capacity of the disk. For more information, see the pricing page.

Can I use a VHD file in an Azure storage account to create a managed disk with a different subscription?

Yes.

Can I use a VHD file in an Azure storage account to create a managed disk in a different region?

No.

Are there limits on how many managed disks I can have?

Yes. The maximum limit is 50,000 managed disks per region and per disk type for a subscription.

Can VMs in an availability set consist of a combination of managed and unmanaged disks?

No. The VMs in an availability set must use either all managed disks or all unmanaged disks. When you create an availability set, you can choose which type of disks you want to use.

Can I create an empty managed disk?

Yes. You can create an empty disk. A managed disk can be created independently of a VM, for example, without attaching it to a VM.

What is the supported fault domain count for an availability set that uses managed disks?

Depending on the region where the availability set that uses managed disks is located, the supported fault domain count is 2 or 3.

What kind of Azure role-based access control support is available for managed disks?

Managed disks supports three key default roles:

- Owner: Can manage everything, including access
- Contributor: Can manage everything except access
- Reader: Can view everything, but can't make changes

Can I copy or export a managed disk to a private storage account?

You can generate a read-only shared access signature (SAS) URI for the managed disk and use it to copy the contents to a private storage account or on-premises storage. You can use the

SAS URI using the Azure portal, Azure PowerShell, the Azure CLI, or AzCopy.

Can I create a copy of my managed disk?

You can take a snapshot of a managed disk and then use the snapshot to create another managed disk. You can also create a new managed disk from an existing managed disk.

Are unmanaged disks still supported?

Yes, but they'll be retired on March 31, 2026 (extended from September 30, 2025). As of November 18, 2022, new customer subscriptions aren't eligible to create unmanaged disks. Page blobs aren't affected by this change. Plan and complete migration of any VMs using unmanaged disks to managed disks before March 31, 2026. See pricing and migration guidance for Windows and Linux VMs, or refer to the migration overview. You should use managed disks for new workloads and migrate your current workloads to managed disks.

Can I colocate unmanaged and managed disks on the same VM?

No.

Can I shrink or downsize my managed disks?

No.

Can I change the computer name property when a specialized (not created by using the System Preparation tool or generalized) operating system disk is used to provision a VM?

No. You can't update the computer name property. The new VM inherits it from the parent VM, which was used to create the operating system disk.

Where can I find sample Azure Resource Manager templates to create VMs with managed disks?

- List of templates using managed disks

- https://github.com/chagarw/MDPP

When creating a disk from a blob, is there any continually existing relationship with that source blob?

No, when the new disk is created it's a full standalone copy of that blob at that time and there's no connection between the two. If you like, once you've created the disk, the source blob may be deleted without affecting the newly created disk in any way.

Can I rename a managed or unmanaged disk after it has been created?

Managed disks can't be renamed. However, you may rename an unmanaged disk as long as it isn't currently attached to a VHD or VM.

Can I use GPT partitioning on an Azure Disk?

Generation 1 images can only use GPT partitioning on data disks, not OS disks. OS disks must use the MBR partition style.

Generation 2 images can use GPT partitioning on the OS disk and the data disks.

What options does Azure disk reservation offer?

Azure disk reservation provides the option to purchase Premium SSDs in the specified SKUs from P30 (1 TiB) up to P80 (32 TiB) for a one-year term. There's no limitation on the minimum number of disks necessary to purchase a disk reservation. Additionally, you can choose to pay with a single, upfront payment or monthly payments. There's no additional transactional cost applied for Premium SSD managed disks.

Reservations are made in the form of disks, not capacity. In other words, when you reserve a P80 (32 TiB) disk, you get a single P80 disk, you can't then divide that specific reservation up into two smaller P70 (16 TiB) disks. You can reserve as many or as few disks as you like, including two separate P70 (16 TiB) disks.

How is Azure disk reservation applied?

Disks reservation follows a model similar to reserved virtual machine (VM) instances. The difference is that a disk reservation can't be applied to different SKUs, while a VM instance can. See Save costs with Azure Reserved VM Instances for more information on VM instances.

Can I use my data storage purchased through Azure disks reservation across multiple regions?

Azure disks reservations are purchased for a specific region and SKU (like P30 in East US 2), and can't be used outside these constructs. You can always purchase an additional Azure Disks Reservation for your disk storage needs in other regions or SKUs.

Do managed disks support "single instance VM SLA"?

Yes, all disk types support single instance VM SLA.

Can I attach a disk to a VM in another region?

No. All managed disks, even shared disks, must be in the same region as the VM they're attaching to.

For Premium SSD v2, the Azure pricing calculator gives me different pricing?

Premium SSD v2 provides a baseline performance of 3,000 IOPS and 125 MB/s for any size that is offered at no extra cost. Currently, for the following regions, the pricing calculator has an incorrect price. Until the pricing calculator is corrected, deduct 3000IOPS from target IOPS and 125MB/s for target bandwidth for these regions to correctly estimate the cost: North Central US, West Central US, West US, UK West, UAE Central, UAE North, Switzerland North, Switzerland West, Sweden South, Korea Central, Korea South, Japan East, Japan West, Italy North, Central India, South India, Germany North, Germany West Central, France Central, Canada Central, Canada East, Brazil South, US GOV Arizona, US GOV Virginia, Australia Central, Australia Central 2, Australia Southeast, South Africa North, South Arica West.

Why should I use the data disk to store applications and data instead of the OS disk?

- You can have more efficient backup and recovery operations, reducing downtime in the event of data loss or system failure, by frequently backing up your data disks. If the data disk experiences an issue, it's easier to recover since it's separate from the OS disk.
- You can expand your storage capacity by adding or expanding data disks without affecting the OS disk. Only data disks support live resize, you can't increase the size of OS disks without stopping the VM.
- Separating the OS from your applications and data onto different disks helps ensure that disks IOPs from the operating system and the application data don't interfere with each other. This can enhance overall system performance, and prevent contention issues that might happen when both the OS and applications are competing for disk resources.
- If there's a system failure or corruption occurs on the OS disk, you can reimage or reinstall the operating system without affecting the data disk. This reduces the risk of data loss and makes it easier when troubleshooting and for maintenance procedures.
- You can apply separate access controls and permissions to the OS disk and the data disk. This separation limits access to critical system files on the OS disk, mitigating potential security risks.

Snapshots

Can an encrypted incremental snapshot be copied across regions?

Yes.

Can I copy snapshots in an order other than their order of creation to another region?

No. Snapshots must be copied to other regions in creation order.

If a source incremental snapshot is deleted before a copy across regions completes, what happens?

The copy fails.

Are managed snapshots and images encrypted?

Yes. All managed snapshots and images are automatically encrypted.

What disk types support snapshots?

All disk types support some form of snapshot. For Ultra Disks and Premium SSD v2 disks, they only support incremental snapshots and have some limitations. For details, see Create an incremental snapshot for managed disks. The other disk types support both types of snapshots for all their disk sizes.

What happens if I have multiple incremental snapshots and delete one of them?

Deleting one of your incremental snapshots doesn't affect subsequent incremental snapshots. The system merges the data occupied by the first snapshot with the next snapshot under the hood to ensure that the subsequent snapshots aren't impacted due to the deletion of the first snapshot.

Azure shared disks

If I have an existing disk, can I enable shared disks on it?

Compatible managed disks created with API version 2019-07-01 or newer can enable shared disks. To do this, you need to unmount the disk from all VMs that it's attached to. Then, edit the **maxShares** property on the disk.

If I no longer want to use a disk in shared mode, how do I disable it?

Unmount the disk from all VMs that it's attached to. Then edit the maxShare property on the disk to 1.

Can you increase the size of a shared disk?

Yes.

Ultra Disks

What should I set my Ultra Disk throughput to?

If you're unsure what to set your disk throughput to, we recommend you start by assuming an IO size of 16 KiB and adjust the performance from there as you monitor your application. The formula is: Throughput in MB/s = # of IOPS * 16 / 1000.

I configured my disk to 40000 IOPS but I'm only seeing 12800 IOPS, why am I not seeing the performance of the disk?

In addition to the disk throttle, there's an IO throttle that gets imposed at the VM level. Ensure that the VM size you're using can support the levels that are configured on your disks. For details regarding IO limits imposed by your VM, see Sizes for virtual machines in Azure.

Can I use caching levels with an Ultra Disk?

No, Ultra Disks don't support the different caching methods that are supported on other disk types. Set the disk caching to **None**.

Can I attach an Ultra Disk to my existing VM?

Maybe, your VM has to be in a region and availability zone pair that supports Ultra Disks. See getting started with Ultra Disks for details.

Can I use an Ultra Disk as the OS disk for my VM?

No, Ultra Disks are only supported as data disks. You can migrate data from an existing data disk to an Ultra Disk. Attach both disks to the same VM and directly copy the data to the Ultra Disk, or utilize a third party solution for data migration.

Can I convert an existing disk to an Ultra Disk?

No, but you can migrate the data from an existing disk to an Ultra Disk. To migrate an existing disk to an Ultra Disk, attach both disks to the same VM, and copy the disk's data from one disk to the other or use a third party solution for data migration.

Can I attach an Ultra Disk to a VM running in an availability set?

No, this isn't currently supported.

Uploading to a managed disk

Can I upload data to an existing managed disk?

No, upload can only be used during the creation of a new empty disk with the **ReadyToUpload** state.

Can I attach a disk to a VM while it is in an upload state?

No.

Migrate to managed disks

Is there any impact of migration on the managed disks performance?

Migration involves movement of the disk from one storage location to another. This is orchestrated via background copy of data, which can take several hours to complete, typically less than 24 Hrs depending on the amount of data in the disks. During that time your application can experience higher than usual read latency as some reads can get redirected to the original location, and can take longer to complete. There's no impact on write latency during this period.

What changes are required in a pre-existing Azure Backup service configuration prior/after migration to managed disks?

No changes are required.

Will my VM backups created via Azure Backup service before the migration continue to work?

Yes, backups work seamlessly.

What changes are required in a pre-existing Azure Disks Encryption configuration prior/after migration to managed disks?

No changes are required.

Is automated migration of an existing virtual machine scale set from unmanaged disks to managed disks supported?

No. You can create a new scale set with managed disks using the image from your old scale set with unmanaged disks.

Can I create a managed disk from a page blob snapshot taken before migrating to managed disks?

No. You can export a page blob snapshot as a page blob and then create a managed disk from the exported page blob.

Can I fail over my on-premises machines protected by Azure Site Recovery to a VM with managed disks?

Yes, you can choose to fail over to a VM with managed disks.

Is there any impact of migration on Azure VMs protected by Azure Site Recovery via Azure to Azure replication?

No. Azure Site Recovery Azure to Azure protection for VMs with managed disks is available.

Can I migrate VMs with unmanaged disks that are located on storage accounts that are or were previously encrypted to managed disks?

Yes

managed disks and Storage Service Encryption

Is Server-side Encryption enabled by default when I create a managed disk?

Yes. Managed disks are encrypted with server-side encryption with platform managed keys.

Is the boot volume encrypted by default on a managed disk?

Yes. By default, all managed disks are encrypted, including the OS disk.

Can I disable Server-side Encryption for my managed disks?

No.

Does Azure Site Recovery support server-side encryption with customer-managed key for on-premises to Azure and Azure to Azure disaster recovery scenarios?

Yes.

Can I backup managed disks encrypted with server-side encryption with customer-managed key using Azure Backup service?

Yes.

Can I convert VMs with unmanaged disks that are located on storage accounts that are or were previously encrypted to managed disks?

Yes

Will an exported VHD from a managed disk or a snapshot also be encrypted?

No. But if you export a VHD to an encrypted storage account from an encrypted managed disk or snapshot, then it's encrypted.

Can I switch from Azure Disk Encryption to server-side encryption with either customer-managed keys or encryption at host?

Yes, it's possible to migrate from Azure Disk Encryption to encryption at host, though it requires creating new disks and VMs rather than in-place conversion. For detailed step-by-step instructions, see Migrate from Azure Disk Encryption to encryption at host.

Can I switch from server-side encryption with customer-managed keys to Azure Disk Encryption?

Yes. First, switch your disk to use platform-managed keys with one of the following steps:

Portal

1. Sign in to the Azure portal.
2. Select the disk you'd like to change the encryption type of.
3. Select **Encryption**.
4. For **Key management** select **Platform-managed key** and select save.

Your managed disk has successfully switched from being secured with your own customer-managed key to a platform-managed key.

Then, encrypt your current disk with Azure Disk Encryption.

Premium disks: Managed and unmanaged

If a VM uses a size series that supports Premium SSDs, such as a DSv2, can I attach both premium and standard data disks?

Yes.

Can I deploy a VM with an unmanaged disk in the Azure portal?

Not all OS images support deploying with an unmanaged disk in the Azure portal. If your chosen image doesn't support this method of deployment, we recommend that you use managed disks for new workloads and convert your existing disk to a managed disk. If you're unable to use managed disks, you can deploy a VM using unmanaged disks with either the Azure PowerShell module or the Azure CLI.

Can I attach both premium and standard data disks to a size series that doesn't support Premium SSDs, such as D, Dv2, G, or F series?

No. You can attach only standard data disks to VMs that don't use a size series that supports Premium SSDs.

Are there transaction costs to use Premium SSDs?

There's a fixed cost for each disk size, which comes provisioned with specific limits on IOPS and throughput. The other costs are outbound bandwidth, snapshot capacity, and costs associated with on-demand bursting, if applicable. For more information, see the pricing page.

What are the limits for IOPS and throughput that I can get from the disk cache?

The combined limits for cache and local SSD for a DS series are 4,000 IOPS per core and 33 MB/s per core. The GS series offers 5,000 IOPS per core and 50 MB/s per core.

Is the local SSD supported for a VM using managed disks?

The local SSD is temporary storage that is included with a VM using managed disks. There's no extra cost for this temporary storage. You shouldn't use this local SSD to store application data because it isn't persisted in Azure Storage.

Are there any repercussions for the use of TRIM on premium disks?

There's no downside to the use of TRIM on Azure disks on either premium or standard disks.

New disk sizes

What is the largest managed disk size supported for operating system and data disks on Gen1 VMs?

The partition type that Azure supports for Gen1 operating system disks is the master boot record (MBR). Although Gen1 OS disks only support MBR the data disks support GPT. While you can allocate up to a 4-TiB OS disk, the MBR partition type can only use up to 2 TiB of this disk space for the operating system. Azure supports up to 32 TiB for managed data disks.

What is the largest managed disk size supported for operating system and data disks on Gen2 VMs?

The partition type that Azure supports for Gen2 operating system disks is GUID Partition Table (GPT). Gen2 VMs support up to a 4-TiB OS disk. Azure supports up to 32 TiB for managed data disks.

What is the largest unmanaged disk size supported for operating system and data disks?

The partition type that Azure supports for an operating system disk using unmanaged disks is the master boot record (MBR). While you can allocate up to 4 TiB for an OS disk, the MBR

partition type can only use up to 2 TiB of this disk space for the operating system. Azure supports up to 4 TiB for unmanaged data disks.

What is the largest supported page blob size?

The largest page blob size that Azure supports is 8 TiB (8,191 GiB). The maximum page blob size when attached to a VM as data or operating system disks is 4 TiB (4,095 GiB).

Are P4 and P6 disk sizes supported for unmanaged disks or page blobs?

P4 (32 GiB) and P6 (64 GiB) disk sizes aren't supported as the default disk tiers for unmanaged disks and page blobs. You need to explicitly set the Blob Tier to P4 and P6 to have your disk mapped to these tiers. If you deploy an unmanaged disk or page blob with the disk size or content length less than 32 GiB or between 32 GiB to 64 GiB without setting the Blob Tier, you'll continue to land on P10 with 500 IOPS and 100 MB/s and the mapped pricing tier.

If my existing premium managed disk less than 64 GiB was created before the small disk was enabled (around June 15, 2017), how is it billed?

Existing small premium disks less than 64 GiB continue to be billed according to the P10 pricing tier.

How can I switch the disk tier of small premium disks less than 64 GiB from P10 to P4 or P6?

You can take a snapshot of your small disks and then create a disk to automatically switch the pricing tier to P4 or P6 based on the provisioned size. You can also use performance tiers, see the articles for changing performance tiers with either the Azure CLI/PowerShell module or the Azure portal.

Can I resize existing managed disks from sizes fewer than 4 tebibytes (TiB) to 32 TiB?

Yes.

What are the largest disk sizes supported by Azure Backup and Azure Site Recovery service?

The largest disk size supported by Azure Backup is 32 TiB (4 TiB for encrypted disks). The largest disk size supported by Azure Site Recovery is 8 TiB. Support for the larger disks up to 32 TiB isn't yet available in Azure Site Recovery.

What are the recommended VM sizes for larger disk sizes (>4 TiB) for Standard SSD and Standard HDDs to achieve optimized disk IOPS and Bandwidth?

To achieve the disk throughput of Standard SSD and Standard HDD large disk sizes (>4 TiB) beyond 500 IOPS and 60 MB/s, we recommend you deploy a new VM from one of the following VM sizes to optimize your performance: B-series, DSv2-series, Dsv3-Series, ESv3-Series, Fs-series, Fsv2-series, M-series, GS-series, NCv2-series, NCv3-series, or Ls-series VMs. Attaching large disks to existing VMs or VMs that aren't using the recommended sizes above may experience lower performance.

How can I upgrade my disks (>4 TiB) which were deployed during the larger disk sizes preview in order to get the higher IOPS & bandwidth at GA?

You can either stop and start the VM that the disk is attached to or, detach and reattach your disk. The performance targets of larger disk sizes have been increased for both Premium SSDs and Standard SSDs at GA.

Do we support enabling Host Caching on all disk sizes?

Host Caching (**ReadOnly** and **Read/Write**) is supported on disk sizes less than 4 TiB. This means any disk that is provisioned up to 4,095 GiB can take advantage of Host Caching. Host caching isn't supported for disk sizes more than or equal to 4,096 GiB. For example, a P50 premium disk provisioned at 4,095 GiB can take advantage of Host caching and a P50 disk provisioned at 4,096 GiB can't take advantage of Host Caching. We recommend using caching for smaller disk sizes where you can expect to observe better performance boost with data cached to the VM.

Private Links for managed disks

How can I ensure that a disk can be exported or imported only via Private Links?

Set the `DiskAccessId` property to an instance of a disk access object and set the NetworkAccessPolicy property to `AllowPrivate`.

Can I use the SAS URI of a disk or snapshot to download the underlying VHD of a VM in the same subnet as the subnet of the private endpoint associated with the disk?

Yes.

Can I use a SAS URI of a disk/snapshot to download the underlying VHD of a VM not in the same subnet as the subnet of the private endpoint not associated with the disk?

No.

What if my question isn't answered here?

If your question isn't listed here, let us know and we'll help you find an answer. You can post a question at the end of this article in the comments. To engage with the Azure Storage team and other community members about this article, use the Microsoft Q&A question page for Azure Storage.

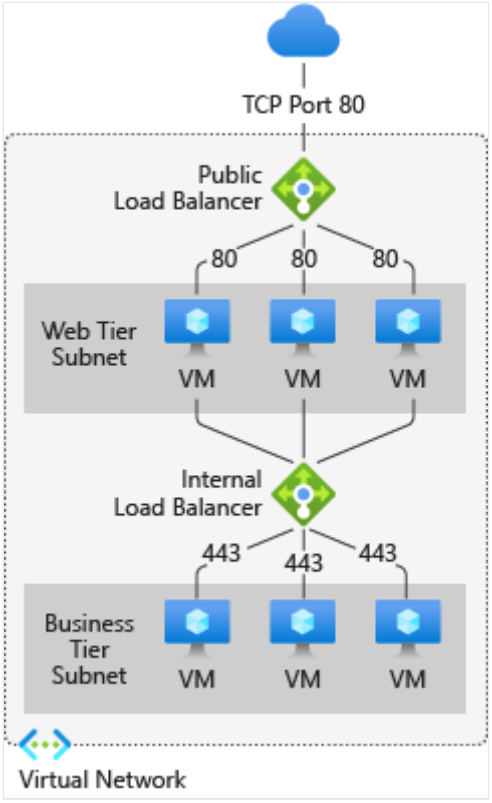
To request features, submit your requests and ideas to the Azure Storage feedback forum.

Virtual networks and virtual machines in Azure

Article • 04/15/2025

When you create a virtual machine (VM), you create a virtual network or use an existing one. Decide how your virtual machines are intended to be accessed on the virtual network. It's important to plan before creating resources and make sure you understand the limits of networking resources.

In the following figure, virtual machines are represented as web servers and application servers. Each set of virtual machines is assigned to separate subnets in the virtual network.



You can create a virtual network before you create a virtual machine or you can create the virtual network as you create a virtual machine.

You create these resources to support communication with a virtual machine:

- Network interfaces
- IP addresses
- Virtual network and subnets

Additionally, consider these optional resources:

- Network security groups

- Load balancers

Network interfaces

A network interface (NIC) is the interconnection between a virtual machine and a virtual network. A virtual machine must have at least one NIC. A virtual machine can have more than one NIC, depending on the size of the VM you create. To learn about the number of NICs each virtual machine size supports, see VM sizes.

You can create a VM with multiple NICs, and add or remove NICs through the lifecycle of a VM. Multiple NICs allow a VM to connect to different subnets.

Each NIC attached to a VM must exist in the same location and subscription as the VM. Each NIC must be connected to a virtual network that exists in the same Azure location and subscription as the NIC. The subnet a VM is connected to can be changed after the VM is created. You can't change the virtual network. Each NIC attached to a VM is assigned a MAC address that doesn't change until the VM is deleted.

This table lists the methods that you can use to create a network interface.

Expand table

Method	Description
Azure portal	When you create a VM in the Azure portal, a network interface is automatically created for you. The portal creates a VM with only one NIC. If you want to create a VM with more than one NIC, you must create it with a different method.
Azure PowerShell	Use New-AzNetworkInterface with the `-PublicIpAddressId` parameter to provide the identifier of the public IP address that you previously created.
Azure CLI	To provide the identifier of the public IP address that you previously created, use az network nic create with the `--public-ip-address` parameter.
Template	For information on deploying a networking interface using a template, see Network Interface in a Virtual Network with Public IP Address.

IP addresses

You can assign these types of IP addresses to a network interface in Azure:

- **Public IP addresses** - Used to communicate inbound and outbound (without network address translation (NAT)) with the Internet and other Azure resources not connected to a virtual network. Assigning a public IP address to a NIC is optional. Public IP addresses

have a nominal charge, and there's a maximum number that can be used per subscription.

- **Private IP addresses** - Used for communication within a virtual network, your on-premises network, and the Internet (with NAT). At least one private IP address must be assigned to a VM. To learn more about NAT in Azure, read Understanding outbound connections in Azure.

You can assign public IP addresses to:

- Virtual machines
- Public load balancers

You can assign private IP address to:

- Virtual machines
- Internal load balancers

You assign IP addresses to a VM using a network interface.

Azure assigns IP addresses dynamically by default. An IP address isn't assigned at creation but is assigned when you create or start a stopped VM. The IP address is released when you stop or delete the VM.

To keep the IP address for the VM the same, set the allocation method to static. The IP address assigns immediately and releases only if you delete the VM or change its allocation method to dynamic.

This table lists the methods that you can use to create an IP address.

Expand table

Method	Description
Azure portal	By default, public IP addresses are dynamic. The IP address might change when the VM is stopped or deleted. To guarantee that the VM always uses the same public IP address, create a static public IP address. By default, the portal assigns a dynamic private IP address to a NIC when creating a VM. You can change this IP address to static after the VM is created.
Azure PowerShell	You use New-AzPublicIpAddress with the `-AllocationMethod` parameter as Dynamic or Static.
Azure CLI	You use az network public-ip create with the `--allocation-method` parameter as Dynamic or Static.

Method	Description
Template	For more information on deploying a public IP address using a template, see Network Interface in a Virtual Network with Public IP Address.

After you create a public IP address, you can associate it with a VM by assigning it to a NIC.

> **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Virtual network and subnets

A subnet is a range of IP addresses in the virtual network. You can divide a virtual network into multiple subnets for organization and security. Each NIC in a VM is connected to one subnet in one virtual network. NICs connected to subnets (same or different) within a virtual network can communicate with each other without any extra configuration.

When you set up a virtual network, you specify the topology, including the available address spaces and subnets. Select address ranges that don't overlap if the virtual network is connected to other virtual networks or on-premises networks. The IP addresses are private and can't be accessed from the Internet. Azure treats any address range as part of the private virtual network IP address space. The address range is only reachable within the virtual network, within interconnected virtual networks, and from your on-premises location.

If you work within an organization in which someone else is responsible for the internal networks, talk to that person before selecting your address space. Ensure there's no overlap in the address space. Communicate to them the space you want to use so they don't try to use the same range of IP addresses.

There aren't security boundaries by default between subnets. Virtual machines in each of these subnets can communicate. If your deployment requires security boundaries, use **Network Security Groups (NSGs)**, which control the traffic flow to and from subnets and to and from VMs.

This table lists the methods that you can use to create a virtual network and subnets.

Expand table

Method	Description
Azure portal	If you let Azure create a virtual network when you create a VM, the name is a combination of the resource group name that contains the virtual network and `-vnet`. The address space is 10.0.0.0/24, the required subnet name is **default**, and the subnet address range is 10.0.0.0/24.
Azure PowerShell	You use New-AzVirtualNetworkSubnetConfig and New-AzVirtualNetwork to create a subnet and a virtual network. You can also use Add-AzVirtualNetworkSubnetConfig to add a subnet to an existing virtual network.
Azure CLI	The subnet and the virtual network are created at the same time. Provide a `--subnet-name` parameter to az network vnet create with the subnet name.
Template	For more information on using a template to create a virtual network and subnets, see Virtual Network with two subnets.

Multiple address prefixes on a subnet

Multiple address prefixes on a subnet are a feature that offers the ability to modify IP address spaces on a subnet. With this solution, customers using virtual machines and virtual machine scale sets can add and remove IP address prefixes to meet their scaling requirements.

Limitations:

- Customers can only use a single customer address (CA) configuration per NIC.
- Customers can only use on VM and Virtual Machine Scale Sets subnets. Delegated subnets aren't supported.

Pricing: Multiple address prefixes on a subnet are offered free of charge.

We welcome you to share your feedback about this feature in this quick survey.

> **Important**
>
> Multiple prefix support for Azure Virtual Network subnets is currently in public preview. This preview version has no service level agreement and isn't recommended for production workloads. Certain features might not be supported or might have limited capabilities. For more information, see Supplemental Terms of Use for Microsoft Azure Previews.

For more information about how-to configure multiple address prefixes on a subnet, see Create multiple prefixes for a subnet.

> **Important**
>
> There are two subnet properties for address space, **AddressPrefix** (string), and **AddressPrefixes** (list). The distinction and usage is explained as follows.
>
> - The array property was introduced for dual stack. The property is also used for scenarios with more than one subnet prefixes as discussed previously.
> - As part of the Azure portal customer experience update, the **AddressPrefixes** is the default property for subnet address space when a subnet is created via the portal.
>   - Any new subnets created via portal default to the **AddressPrefixes** list parameter.
> - If customers are using dual-stack in their virtual network or have more than one subnet prefixes, they're updated to use the list property.
>   - For existing deployments using the string, the current behavior is retained unless there are explicit changes in your virtual network to use the list property for subnet address prefixes. An example is adding IPv6 address space or another prefix to the subnet.
> - We recommend that customers should look for both the properties in subnet wherever applicable.

Network security groups

A network security group (NSG) contains a list of Access Control List (ACL) rules that allow or deny network traffic to subnets, NICs, or both. NSGs can be associated with either subnets or individual NICs connected to a subnet. When an NSG is associated with a subnet, the ACL rules

apply to all the VMs in that subnet. Traffic to an individual NIC can be restricted by associating an NSG directly to a NIC.

NSGs contain two sets of rules, inbound and outbound. The priority for a rule must be unique within each set.

Each rule has properties of:

- Protocol
- Source and destination port ranges
- Address prefixes
- Direction of traffic
- Priority
- Access type

All NSGs contain a set of default rules. You can't delete or override these default rules, as they have the lowest priority and any rules you create can't supersede them.

When you associate an NSG to a NIC, the network access rules in the NSG are applied only to that NIC. If an NSG is applied to a single NIC on a multi-NIC VM, it doesn't affect traffic to the other NICs. You can associate different NSGs to a NIC (or VM, depending on the deployment model) and the subnet that a NIC or VM is bound to. Priority is given based on the direction of traffic.

Be sure to plan your NSGs when you plan your virtual machines and virtual network.

This table lists the methods that you can use to create a network security group.

Expand table

Method	Description
Azure portal	When you create a VM in the Azure portal, an NSG is automatically created and associated to the NIC the portal creates. The name of the NSG is a combination of the name of the VM and `-nsg`. This NSG contains one inbound rule: With a priority of 1000. The service set to RDP. The protocol set to TCP. The port set to 3389. The action set to **Allow**. If you want to allow any other inbound traffic to the VM, create another rule or rules.

Method	Description
Azure PowerShell	Use New-AzNetworkSecurityRuleConfig and provide the required rule information. Use New-AzNetworkSecurityGroup to create the NSG. Use Set-AzVirtualNetworkSubnetConfig to configure the NSG for the subnet. Use Set-AzVirtualNetwork to add the NSG to the virtual network.
Azure CLI	Use az network nsg create to initially create the NSG. Use az network nsg rule create to add rules to the NSG. Use az network vnet subnet update to add the NSG to the subnet.
Template	Use Create a Network Security Group as a guide for deploying a network security group using a template.

Load balancers

Azure Load Balancer delivers high availability and network performance to your applications. A load balancer can be configured to balance incoming Internet traffic to VMs or balance traffic between VMs in a virtual network. A load balancer can also balance traffic between on-premises computers and VMs in a cross-premises network, or forward external traffic to a specific VM.

The load balancer maps incoming and outgoing traffic between:

- The public IP address and port on the load balancer.
- The private IP address and port of the VM.

When you create a load balancer, you must also consider these configuration elements:

- **Front-end IP configuration** – A load balancer can include one or more front-end IP addresses. These IP addresses serve as ingress for the traffic.
- **Back-end address pool** – IP addresses that are associated with the NIC to which load is distributed.
- **Port Forwarding** - Defines how inbound traffic flows through the front-end IP and distributed to the back-end IP using inbound NAT rules.
- **Load balancer rules** - Maps a given front-end IP and port combination to a set of back-end IP addresses and port combination. A single load balancer can have multiple load-balancing rules. Each rule is a combination of a front-end IP and port and back-end IP and port associated with VMs.
- **Probes** - Monitors the health of VMs. When a probe fails to respond, the load balancer stops sending new connections to the unhealthy VM. The existing connections aren't

affected, and new connections are sent to healthy VMs.

- **Outbound rules** - An outbound rule configures outbound Network Address Translation (NAT) for all virtual machines or instances identified by the backend pool of your Standard Load Balancer to be translated to the frontend.

This table lists the methods that you can use to create an internet-facing load balancer.

Expand table

Method	Description
Azure portal	You can load balance internet traffic to VMs using the Azure portal.
Azure PowerShell	To provide the identifier of the public IP address that you previously created, use New-AzLoadBalancerFrontendIpConfig with the `-PublicIpAddress` parameter. Use New-AzLoadBalancerBackendAddressPoolConfig to create the configuration of the back-end address pool. Use New-AzLoadBalancerInboundNatRuleConfig to create inbound NAT rules associated with the front-end IP configuration that you created. Use New-AzLoadBalancerProbeConfig to create the probes that you need. Use New-AzLoadBalancerRuleConfig to create the load balancer configuration. Use New-AzLoadBalancer to create the load balancer.
Azure CLI	Use az network lb create to create the initial load balancer configuration. Use az network lb frontend-ip create to add the public IP address that you previously created. Use az network lb address-pool create to add the configuration of the back-end address pool. Use az network lb inbound-nat-rule create to add NAT rules. Use az network lb rule create to add the load balancer rules. Use az network lb probe create to add the probes.
Template	Use 3 VMs in a Load Balancer as a guide for deploying a load balancer using a template.

This table lists the methods that you can use to create an internal load balancer.

Expand table

Method	Description
Azure portal	You can balance internal traffic load with a load balancer in the Azure portal.
Azure PowerShell	To provide a private IP address in the network subnet, use New-AzLoadBalancerFrontendIpConfig with the `-PrivateIpAddress` parameter. Use New-AzLoadBalancerBackendAddressPoolConfig to create the configuration of the back-end address pool. Use New-AzLoadBalancerInboundNatRuleConfig to create inbound NAT rules associated with the front-end IP configuration that you created. Use New-AzLoadBalancerProbeConfig to create the probes that you need. Use New-

Method	Description
	AzLoadBalancerRuleConfig to create the load balancer configuration. Use New-AzLoadBalancer to create the load balancer.
Azure CLI	Use the az network lb create command to create the initial load balancer configuration. To define the private IP address, use az network lb frontend-ip create with the `--private-ip-address` parameter. Use az network lb address-pool create to add the configuration of the back-end address pool. Use az network lb inbound-nat-rule create to add NAT rules. Use az network lb rule create to add the load balancer rules. Use az network lb probe create to add the probes.
Template	Use 2 VMs in a Load Balancer as a guide for deploying a load balancer using a template.

Virtual machines

Virtual machines can be created in the same virtual network and they can connect to each other using private IP addresses. Virtual machines can connect if they're in different subnets. They connect without the need to configure a gateway or use public IP addresses. To put VMs into a virtual network, you create the virtual network. As you create each VM, you assign it to the virtual network and subnet. Virtual machines acquire their network settings during deployment or startup.

Virtual machines are assigned an IP address when they're deployed. When you deploy multiple VMs into a virtual network or subnet, they're assigned IP addresses as they boot up. You can also assign a static IP to a VM. If you assign a static IP, you should consider using a specific subnet to avoid accidentally reusing a static IP for another VM.

If you create a VM and later want to migrate it into a virtual network, it isn't a simple configuration change. Redeploy the VM into the virtual network. The easiest way to redeploy is to delete the VM, but not any disks attached to it, and then re-create the VM using the original disks in the virtual network.

This table lists the methods that you can use to create a VM in a virtual network.

Expand table

Method	Description
Azure portal	Uses the default network settings that were previously mentioned to create a VM with a single NIC. To create a VM with multiple NICs, you must use a different method.
Azure PowerShell	Includes the use of Add-AzVMNetworkInterface to add the NIC that you previously created to the VM configuration.

Method	Description
Azure CLI	Create and connect a VM to a virtual network, subnet, and NIC that builds as individual steps.
Template	Use Very simple deployment of a Windows VM as a guide for deploying a VM using a template.

NAT Gateway

Azure NAT Gateway simplifies outbound-only Internet connectivity for virtual networks. When configured on a subnet, all outbound connectivity uses your specified static public IP addresses. Outbound connectivity is possible without load balancer or public IP addresses directly attached to virtual machines. NAT is fully managed and highly resilient.

Outbound connectivity can be defined for each subnet with NAT. Multiple subnets within the same virtual network can have different NATs. A subnet is configured by specifying which NAT gateway resource to use. All UDP and TCP outbound flows from any virtual machine instance use a NAT gateway. NAT is compatible with standard public IP address resources or public IP prefix resources or a combination of both. You can use a public IP prefix directly or distribute the public IP addresses of the prefix across multiple NAT gateway resources. NAT grooms all traffic to the range of IP addresses of the prefix. Any IP filtering of your deployments is easier.

NAT Gateway automatically processes all outbound traffic without any customer configuration. User-defined routes aren't necessary. NAT takes precedence over other outbound scenarios and replaces the default Internet destination of a subnet.

Virtual machine scale sets that create virtual machines with Flexible Orchestration mode don't have default outbound access. Azure NAT Gateway is the recommended outbound access method for Virtual machine scale sets Flexible Orchestration Mode.

For more information about Azure NAT Gateway, see What is Azure NAT Gateway?.

This table lists the methods that you can use to create a NAT gateway resource.

Expand table

Method	Description
Azure portal	Creates a virtual network, subnet, public IP, NAT gateway, and a virtual machine to test the NAT gateway resource.
Azure PowerShell	Includes the use of New-AzNatGateway to create a NAT gateway resource. Creates a virtual network, subnet, public IP, NAT gateway, and a virtual machine to test the NAT gateway resource.

Method	Description
Azure CLI	Includes the use of az network nat gateway create to create a NAT gateway resource. Creates a virtual network, subnet, public IP, NAT gateway, and a virtual machine to test the NAT gateway resource.
Template	Creates a virtual network, subnet, public IP, and NAT gateway resource.

Azure Bastion

Azure Bastion is deployed to provide secure management connectivity to virtual machines in a virtual network. Azure Bastion Service enables you to securely and seamlessly RDP & SSH to the VMs in your virtual network. Azure bastion enables connections without exposing a public IP on the VM. Connections are made directly from the Azure portal, without the need of an extra client/agent or piece of software. Azure Bastion supports standard public IP addresses.

Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

For more information about Azure Bastion, see What is Azure Bastion?.

This table lists the methods you can use to create an Azure Bastion deployment.

Expand table

Method	Description
Azure portal	Creates a virtual network, subnets, public IP, bastion host, and virtual machines.
Azure PowerShell	Creates a virtual network, subnets, public IP, and bastion host. Includes the use of New-AzBastion to create the bastion host.
Azure CLI	Creates a virtual network, subnets, public IP, and bastion host. Includes the use of az network bastion create to create the bastion host.
Template	For an example of a template deployment that integrates an Azure Bastion host with a sample deployment, see Quickstart: Create a public load balancer to load balance VMs by using an ARM template.

Next steps

For VM-specific steps on how to manage Azure virtual networks for VMs, see the Windows or Linux tutorials.

There are also quickstarts on how to load balance VMs and create highly available applications using the CLI or PowerShell

- Learn how to configure VNet to VNet connections.
- Learn how to Troubleshoot routes.
- Learn more about Virtual machine network bandwidth.

Optimize network throughput for Azure virtual machines

06/12/2025

Azure virtual machines (VMs) have default network settings that can be further optimized for network throughput. This article describes how to optimize network throughput for Windows and Linux VMs, including major distributions such as Ubuntu and Red Hat.

Windows virtual machines

If your Windows VM supports *accelerated networking*, enable that feature for optimal throughput. For more information, see Create a Windows VM with accelerated networking.

For all other Windows VMs, using Receive Side Scaling (RSS) can reach higher maximal throughput than a VM without RSS. RSS might be disabled by default in a Windows VM. To check if RSS is enabled and enable it, follow these steps:

1. See if RSS is enabled for a network adapter with the Get-NetAdapterRss PowerShell command. In the following example, output returned from the `Get-NetAdapterRss` RSS isn't enabled.

   PowerShell

   ```
   Name                    : Ethernet
   InterfaceDescription    : Microsoft Hyper-V Network Adapter
   Enabled                 : False
   ```

2. To enable RSS, enter the following command:

   PowerShell

   ```
   Get-NetAdapter | % {Enable-NetAdapterRss -Name $_.Name}
   ```

   This command doesn't have an output. The command changes network interface card (NIC) settings. It causes temporary connectivity loss for about one minute. A **Reconnecting** dialog appears during the connectivity loss. Connectivity is typically restored after the third attempt.

3. Confirm that RSS is enabled in the VM by entering the `Get-NetAdapterRss` command again. If successful, the following example output is returned:

PowerShell

```
Name                    : Ethernet
InterfaceDescription    : Microsoft Hyper-V Network Adapter
Enabled                 : True
```

Linux virtual machines

RSS is always enabled by default in an Azure Linux Virtual Machine (VM). Linux kernels released since October 2017 include new network optimizations options that enable a Linux VM to achieve higher network throughput.

Enable Azure Accelerated Networking for optimal throughput

Azure provides accelerated networking which can really improve network performance, latency, jitter. There are currently two different technologies that are used depending on the virtual machine size, Mellanox which is wide available and MANA which is developed by Microsoft.

Azure Linux Tuned Kernels

Some distributions such as Ubuntu (Canonical) and SUSE have Azure tuned kernels.

Use the following command to make sure that you're using the Azure kernel, which has usually the `azure` string in the naming.

Bash

```
uname -r

#sample output on Azure kernel on a Ubuntu Linux VM
6.8.0-1017-azure
```

Other Linux distributions

Most modern distributions have significant improvements with newer kernels. Check the current kernel version to make sure that you're running a kernel that is newer than 4.19, which includes some great improvements in networking, for example support for the *BBR Congestion-Based Congestion Control*.

Achieving consistent transfer speeds in Azure Linux VMs

Azure Linux VMs often experience network performance issues, particularly when transferring large files (1 GB to 50 GB) between regions, such as West Europe and West US. These issues are caused by older kernel versions as well as, default kernel configurations, default network buffer settings and default congestion control algorithms, which result in delayed packets, limited throughput, and inefficient resource usage.

To get consistent network performance, consider implementing the following optimizations that are proven effective in many situations on Azure:

- **Network buffer settings**: Adjust kernel parameters to maximize read and write memory buffers. Add these configurations to `/etc/sysctl.d/99-azure-network-buffers.conf`:

plaintext

```
net.ipv4.tcp_mem = 4096 87380 67108864
net.ipv4.udp_mem = 4096 87380 33554432
net.ipv4.tcp_rmem = 4096 87380 67108864
net.ipv4.tcp_wmem = 4096 65536 67108864
net.core.rmem_default = 33554432
net.core.wmem_default = 33554432
net.ipv4.udp_wmem_min = 16384
net.ipv4.udp_rmem_min = 16384
net.core.wmem_max = 134217728
net.core.rmem_max = 134217728
net.core.busy_poll = 50
net.core.busy_read = 50
```

- **Congestion-Based Congestion control for kernels 4.19 and above**: Enabling Bottleneck Bandwidth and Round-trip propagation time (BBR) congestion control can often result in better throughput. Add this configuration to `/etc/sysctl.d/99-azure-congestion-control.conf`:

plaintext

```
net.ipv4.tcp_congestion_control = bbr
```

- **Extra TCP parameters that will usually help with better consistency, throughput**: Add these configurations to `/etc/sysctl.d/99-azure-network-extras.conf`:

plaintext

```
# For deployments where the Linux VM is BEHIND an Azure Load Balancer, timestamps
MUST be set to 0
net.ipv4.tcp_timestamps = 1

# Reuse does require tcp_timestamps to be enabled. If tcp_timestamps are disabled
because of load balancers, you should set reuse to 2.
net.ipv4.tcp_tw_reuse = 1

# Allowed local port range. This will increase the number of locally available
ports (source ports)
net.ipv4.ip_local_port_range = 1024 65535

# Maximum number of packets taken from all interfaces in one polling cycle (NAPI
poll). In one polling cycle interfaces which are # registered to polling are
probed in a round-robin manner.
net.core.netdev_budget = 1000

# For high-performance environments, it's recommended to increase from the default
20KB to 65KB, in some extreme cases, for environments that support 100G+
networking, you can
# increase it to 1048576
net.core.optmem_max = 65535

# F-RTO is not recommended on wired networks.
net.ipv4.tcp_frto = 0

# Increase the number of incoming connections / number of connections backlog
net.core.somaxconn = 32768
net.core.netdev_max_backlog = 32768
net.core.dev_weight = 64
```

- **Queue discipline (qdisc)**: Packet processing in Azure is improved by setting the default qdisc to `fq`. Add this configuration to `/etc/sysctl.d/99-azure-qdisc.conf`:

plaintext

```
net.core.default_qdisc = fq
```

- **Optimize NIC ring buffers for TX/RX**: Create an udev rule in `/etc/udev/rules.d/99-azure-ring-buffer.rules` to ensure they're applied to network interfaces:

plaintext

```
# Setup Accelerated Interface ring buffers (Mellanox / Mana)
SUBSYSTEM=="net", DRIVERS=="hv_pci", ACTION=="add",  RUN+="/usr/sbin/ethtool -G
$env{INTERFACE} rx 1024 tx 1024"

# Setup Synthetic interface ring buffers (hv_netvsc)
```

```
SUBSYSTEM=="net", DRIVERS=="hv_netvsc*", ACTION=="add",  RUN+="/usr/sbin/ethtool -
G $env{INTERFACE} rx 1024 tx 1024"
```

- Create an udev rule in `/etc/udev/rules.d/99-azure-qdisc.rules` to ensure the qdisc is applied to network interfaces:

```
plaintext

ACTION=="add|change", SUBSYSTEM=="net", KERNEL=="enP*", PROGRAM="/sbin/tc qdisc
replace dev \$env{INTERFACE} root noqueue"
ACTION=="add|change", SUBSYSTEM=="net", KERNEL=="eth*", PROGRAM="/sbin/tc qdisc
replace dev \$env{INTERFACE} root fq“
```

- **Interrupt Request (IRQ) scheduling**: Depending on your workload, you may wish to restrict the irqbalance service from scheduling IRQs on certain nodes. When using IRQBalance, you can update `/etc/default/irqbalance` to specify which CPUs shouldn't have IRQs scheduled you will need to determine the mask that will exclude the CPUs that need exclusion.

More information about how to calculate the mask available here.

The example below would assume that you want to exclude CPUs 8-15

```
Bash

IRQBALANCE_BANNED_CPULIST=0000ff00
```

- **UDEV rules**: Add rules to optimize queue length and manage device flags efficiently. Create the following rule in `/etc/udev/rules.d/99-azure-txqueue-len.rules`:

```
plaintext

SUBSYSTEM=="net", ACTION=="add|change", KERNEL=="eth*", ATTR{tx_queue_len}="10000“
```

For Packets delayed twice

When it comes to Linux performance networking we use SR-IOV with Mellanox drivers (mlx4 or mlx5), something specific to Azure is that this creates two interfaces a synthetic and a virtual interface. Learn More.

Additional Notes

System administrators can implement these solutions by editing configuration files such as `/etc/sysctl.d/`, `/etc/modules-load.d/`, and `/etc/udev/rules.d/`. Ensure that kernel driver updates and systemd configurations are reviewed for potential regressions.

For more information on specific configurations and troubleshooting, refer to Azure documentation on networking performance.

Related content

- Deploy VMs close to each other for low latency with proximity placement groups.
- See the optimized result with Bandwidth/Throughput testing for your scenario.
- Read about how bandwidth is allocated to virtual machines.
- Learn more with Azure Virtual Network frequently asked questions.

Azure Virtual Network concepts and best practices

Azure Virtual Network is a fundamental building block for your private network in Azure. It enables Azure resources to securely communicate with each other, the internet, and on-premises networks.

A virtual network is similar to a traditional network that you'd operate in your own data center. However, it brings extra benefits of Azure's infrastructure such as scale, availability, and isolation.

Understanding virtual network concepts and following best practices helps you design robust, secure, and scalable network architectures. Whether you're migrating existing workloads to Azure or building new cloud-native applications, proper network design is essential for performance, security, and cost optimization.

This article describes key Azure Virtual Network concepts and best practices to help you design efficient, secure, and scalable network infrastructure for your cloud-based applications. Learn about address spaces, subnets, security groups, and architectural guidelines that optimize your Azure networking strategy.

Virtual Network concepts

- **Address space:** When creating a virtual network, you must specify a custom private IP address space using public and private (RFC 1918) addresses. Azure assigns resources in a virtual network a private IP address from the address space that you assign. For example, if you deploy a VM in a virtual network with address space, 10.0.0.0/16, a private IP similar to 10.0.0.4 is assigned to the virtual machine.

- **Subnets:** Subnets enable you to segment the virtual network into one or more sub networks and allocate a portion of the virtual network's address space to each subnet. You can then deploy Azure resources in a specific subnet. Just like in a traditional network, subnets allow you to segment your virtual network address space into segments that are appropriate for the organization's internal network. Segmentation improves address allocation efficiency. You can secure resources within subnets using Network Security Groups. For more information, see Network security groups.

- **Regions**: A virtual network is scoped to a single region/location; however, multiple virtual networks from different regions can be connected together using Virtual Network Peering.

- **Subscription:** A virtual network is scoped to a subscription. You can implement multiple virtual networks within each Azure subscription and Azure region.

Best practices

As you build your network in Azure, it's important to keep in mind the following universal design principles:

- Ensure address spaces don't overlap. Make sure your virtual network address spaces (CIDR blocks) don't overlap with your organization's other in-use network ranges.
  - See the recommended address ranges in the FAQ
- Your subnets shouldn't cover the entire address space of the virtual network. Plan ahead and reserve some address space for the future.
- Use a few large virtual networks instead of multiple small ones to reduce management overhead within a single Azure subscription.
  - You should also follow the Azure landing zone design principle of Subscription Democratization meaning it is recommended to have many subscriptions, each with their own virtual network (possibly multiple), to help you achieve and implement Zero Trust in your Azure networking.
  - Commonly a hub and spoke architecture is used to implement this at scale with a traditional virtual network approach or Virtual WAN
- Secure your virtual networks by assigning Network Security Groups (NSGs) to the subnets beneath them. For more information about network security concepts, see Azure network security overview.

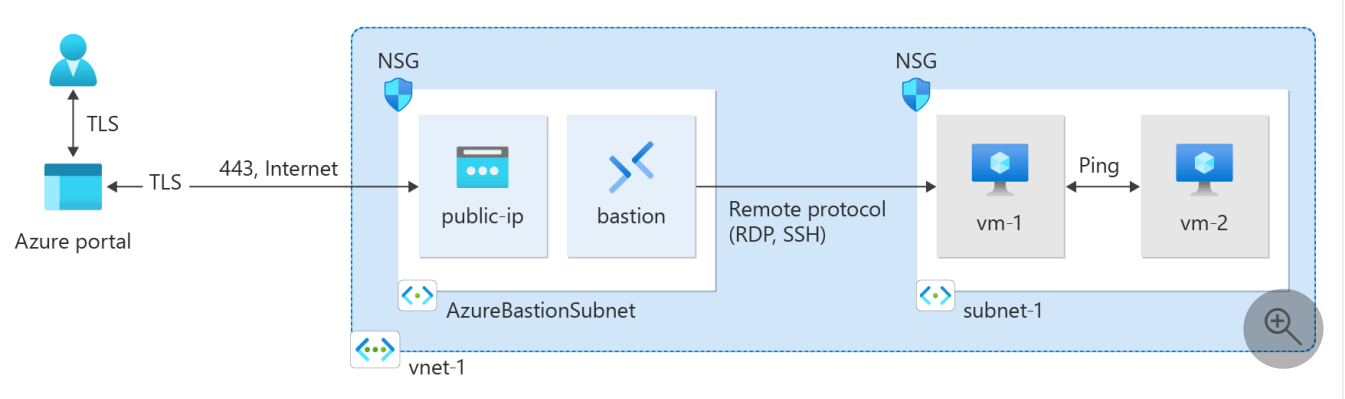
Next steps

To get started using a virtual network, create one, deploy a few VMs to it, and communicate between the VMs. To learn how, see the Create a virtual network quickstart.

Last updated on 12/11/2025

Quickstart: Create an Azure Virtual Network

Learn how to create an Azure Virtual Network using the Azure portal, Azure CLI, Azure PowerShell, Azure Resource Manager (ARM) template, Bicep template, and Terraform. Two virtual machines and an Azure Bastion host are deployed to test connectivity between the virtual machines in the same virtual network. The Azure Bastion host facilitates secure and seamless RDP and SSH connectivity to the virtual machines directly in the Azure portal over SSL.



A virtual network is the fundamental building block for private networks in Azure. Azure Virtual Network enables Azure resources such as virtual machines to securely communicate with each other and the internet.

https://learn-video.azurefd.net/vod/player?id=6b5b138e-8406-406e-8b34-40bdadf9fc6d&locale=en-us&embedUrl=%2Fazure%2Fvirtual-network%2Fquickstart-create-virtual-network

If you don't have an Azure account with an active subscription, create one for free.

Prerequisites

Portal

- An Azure account with an active subscription. You can create an account for free.

Resource values

Use the following values to replace the placeholders of resources in this article:

Expand table

Setting	Placeholder	Value
Resource group	`<resource-group>`	**test-rg**
Region	`<region>`	**East US 2**
Virtual network	`<virtual-network>`	**vnet-1**
Subnet	`<subnet>`	**subnet-1**
Network security group	`<network-security-group>`	**nsg-1**
Bastion	`<bastion>`	**bastion**
Virtual machine 1	`<virtual-machine-1>`	**vm-1**
Virtual machine 2	`<virtual-machine-2>`	**vm-2**

Portal

Create a resource group

1. Sign in to the Azure portal with your Azure account.
2. In the search box at the top of the portal, enter **Resource group**. Select **Resource groups** in the search results.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter, or select the following information:

   Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Enter **<resource-group>**.
Region	Select **<region>**.

5. Select **Review + create**.
6. Select **Create**.

Create a virtual network

1. In the search box at the top of the portal, enter **Virtual network**. Select **Virtual networks** in the search results.
2. Select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter, or select the following information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **<resource-group>**.
Instance details	
Name	Enter **<virtual-network>**.
Region	Select **<region>**.

4. Select **Next** to proceed to the **Security** tab.
5. Select **Next** to proceed to the **IP Addresses** tab.
6. In the address space box in **Subnets**, select the **default** subnet.
7. In **Edit subnet**, enter, or select the following information:

Expand table

Setting	Value
Subnet details	
Subnet template	Leave the default **Default**.
Name	Enter **<subnet>**.
Starting address	Leave the default of **10.0.0.0**.
Subnet size	Leave the default of **/24 (256 addresses)**.

8. Select **Save**.

9. Select **Review + create** at the bottom of the screen, and when validation passes, select **Create**.

Deploy Azure Bastion

Azure Bastion uses your browser to connect to virtual machines (VMs) in your virtual network over secure shell (SSH) or remote desktop protocol (RDP) by using their private IP addresses. The virtual machines don't need public IP addresses, client software, or special configuration. For more information about Azure Bastion, see Azure Bastion.

> **Note**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

1. In the search box at the top of the portal, enter **Bastion**. Select **Bastions** in the search results.
2. Select **+ Create**.
3. In the **Basics** tab of **Create a Bastion**, enter, or select the following information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **<resource-group>**.
Instance details	
Name	Enter **<bastion>**.
Region	Select **<region>**.
Tier	Select **Developer**.
Configure virtual networks	
Virtual network	Select **<virtual-network>**.

4. Select **Review + create**.

5. Select **Create**.

Create virtual machines

The following procedure creates two VMs named **<virtual-machine-1>** and **<virtual-machine-2>** in the virtual network:

1. In the portal, search for and select **Virtual machines**.

2. In **Virtual machines**, select **+ Create**, and then select **Azure virtual machine**.

3. On the **Basics** tab of **Create a virtual machine**, enter, or select the following information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **<resource-group>**.
Instance details	
Virtual machine name	Enter **<virtual-machine-1>**.
Region	Select **<region>**.
Availability options	Select **No infrastructure redundancy required**.
Security type	Leave the default of **Standard**.
Image	Select **Ubuntu Server 22.04 LTS - x64 Gen2**.
VM architecture	Leave the default of **x64**.
Size	Select a size.
Administrator account	
Authentication type	Select **SSH public key**.
Username	Enter **azureuser**.
SSH public key source	Select **Generate new key pair**.

Setting	Value
Key pair name	Enter **<virtual-machine-1>-key**.
Inbound port rules	
Public inbound ports	Select **None**.

4. Select the **Networking** tab. Enter or select the following information:

Expand table

Setting	Value
Network interface	
Virtual network	Select **<virtual-network>**.
Subnet	Select **<subnet> (10.0.0.0/24)**.
Public IP	Select **None**.
NIC network security group	Select **Advanced**.
Configure network security group	Select **Create new**. Enter **<network-security-group>** for the name. Leave the rest at the defaults and select **OK**.

5. Leave the rest of the settings at the defaults and select **Review + create**.

6. Review the settings and select **Create**.

7. Wait for the first virtual machine to deploy then repeat the previous steps to create a second virtual machine with the following settings:

Expand table

Setting	Value
Virtual machine name	Enter **<virtual-machine-2>**.
SSH public key source	Select **Generate new key pair**.
Key pair name	Enter **<virtual-machine-2>-key**.
Virtual network	Select **<virtual-network>**.
Subnet	Select **<subnet> (10.0.0.0/24)**.
Public IP	Select **None**.

Setting	Value
NIC network security group	Select **Advanced**.
Configure network security group	Select **<network-security-group>**.

> **Note**
>
> Virtual machines in a virtual network with an Azure Bastion host don't need public IP addresses. Bastion provides the public IP, and the VMs use private IPs to communicate within the network. You can remove the public IPs from any VMs in Bastion-hosted virtual networks. For more information, see Dissociate a public IP address from an Azure VM.

> **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Connect to a virtual machine

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.

2. In **Virtual machines**, select **vm-1**.

3. Select **Connect** then **Connect via Bastion** in the **Overview** section.

4. In the **Bastion** connection page, enter or select the following information:

Expand table

Setting	Value
Authentication Type	Select **SSH Private Key from Local File**.
Username	Enter **azureuser**.
Local File	Select the private key file you downloaded or created.

5. Select **Connect**.

Start communication between virtual machines

1. At the bash prompt for **vm-1**, enter `ping -c 4 vm-2`.

   You get a reply similar to the following message:

   Output

   ```
   azureuser@vm-1:~$ ping -c 4 vm-2
   PING vm-2.3bnkevn3313ujpr5l1kqop4n4d.cx.internal.cloudapp.net (10.0.0.5)
   56(84) bytes of data.
   64 bytes from vm-2.internal.cloudapp.net (10.0.0.5): icmp_seq=1 ttl=64
   time=1.83 ms
   64 bytes from vm-2.internal.cloudapp.net (10.0.0.5): icmp_seq=2 ttl=64
   time=0.987 ms
   64 bytes from vm-2.internal.cloudapp.net (10.0.0.5): icmp_seq=3 ttl=64
   time=0.864 ms
   64 bytes from vm-2.internal.cloudapp.net (10.0.0.5): icmp_seq=4 ttl=64
   time=0.890 ms
   ```

2. Close the Bastion session.

3. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.

4. In **Virtual machines**, select **vm-2**.

5. Select **Connect** then **Connect via Bastion** in the **Overview** section.

6. In the **Bastion** connection page, enter or select the following information:

Expand table

Setting	Value
Authentication Type	Select **SSH Private Key from Local File**.
Username	Enter **azureuser**.
Local File	Select the private key file you downloaded or created.

7. Select **Connect**.

8. At the bash prompt for **vm-2**, enter `ping -c 4 vm-1`.

   You get a reply similar to the following message:

   Output

   ```
   azureuser@vm-2:~$ ping -c 4 vm-1
   PING vm-1.3bnkevn3313ujpr5l1kqop4n4d.cx.internal.cloudapp.net (10.0.0.4)
   56(84) bytes of data.
   64 bytes from vm-1.internal.cloudapp.net (10.0.0.4): icmp_seq=1 ttl=64
   time=0.695 ms
   64 bytes from vm-1.internal.cloudapp.net (10.0.0.4): icmp_seq=2 ttl=64
   time=0.896 ms
   64 bytes from vm-1.internal.cloudapp.net (10.0.0.4): icmp_seq=3 ttl=64
   time=3.43 ms
   64 bytes from vm-1.internal.cloudapp.net (10.0.0.4): icmp_seq=4 ttl=64
   time=0.780 ms
   ```

9. Close the Bastion session.

Clean up resources

Portal

When you finish using the resources that you created, you can delete the resource group and all its resources.

1. In the Azure portal, search for and select **Resource groups**.

2. On the **Resource groups** page, select the **<resource-group>** resource group.

3. On the **<resource-group>** page, select **Delete resource group**.

4. Enter <**resource-group**> in **Enter resource group name to confirm deletion**, and then select **Delete**.

Related content

- Filter network traffic
- Learn more about using Terraform on Azure

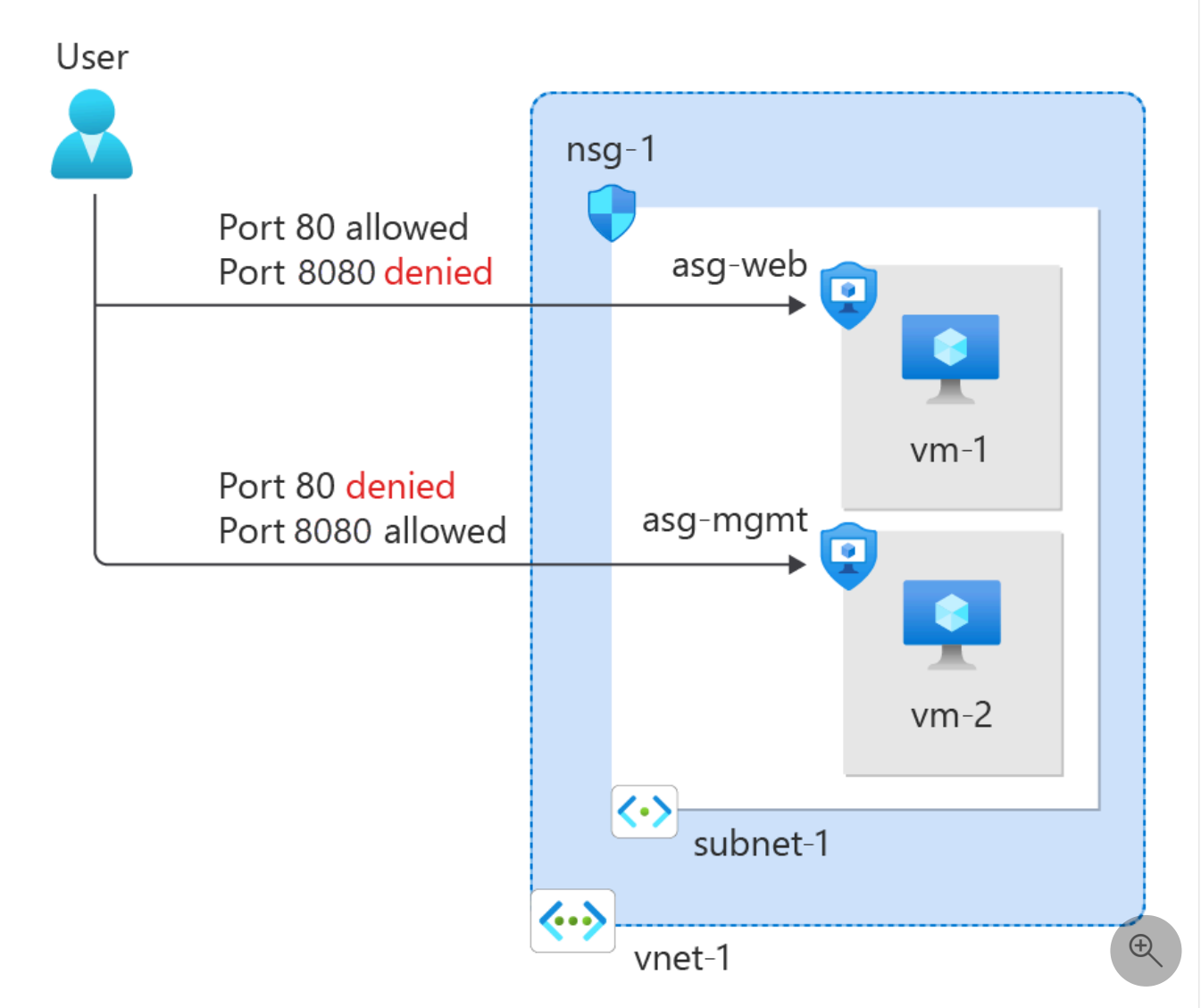
Last updated on 03/27/2026

Tutorial: Filter network traffic with a network security group

Summarize this article for me

You can use a network security group to filter inbound and outbound network traffic to and from Azure resources in an Azure virtual network.

Network security groups contain security rules that filter network traffic by IP address, port, and protocol. When a network security group is associated with a subnet, security rules are applied to resources deployed in that subnet.



In this tutorial, you learn how to:

- ✓ Create a network security group and security rules
- ✓ Create application security groups
- ✓ Create a virtual network and associate a network security group to a subnet

- ✓ Deploy virtual machines and associate their network interfaces to the application security groups

Prerequisites

Portal

- An Azure account with an active subscription. You can create an account for free.

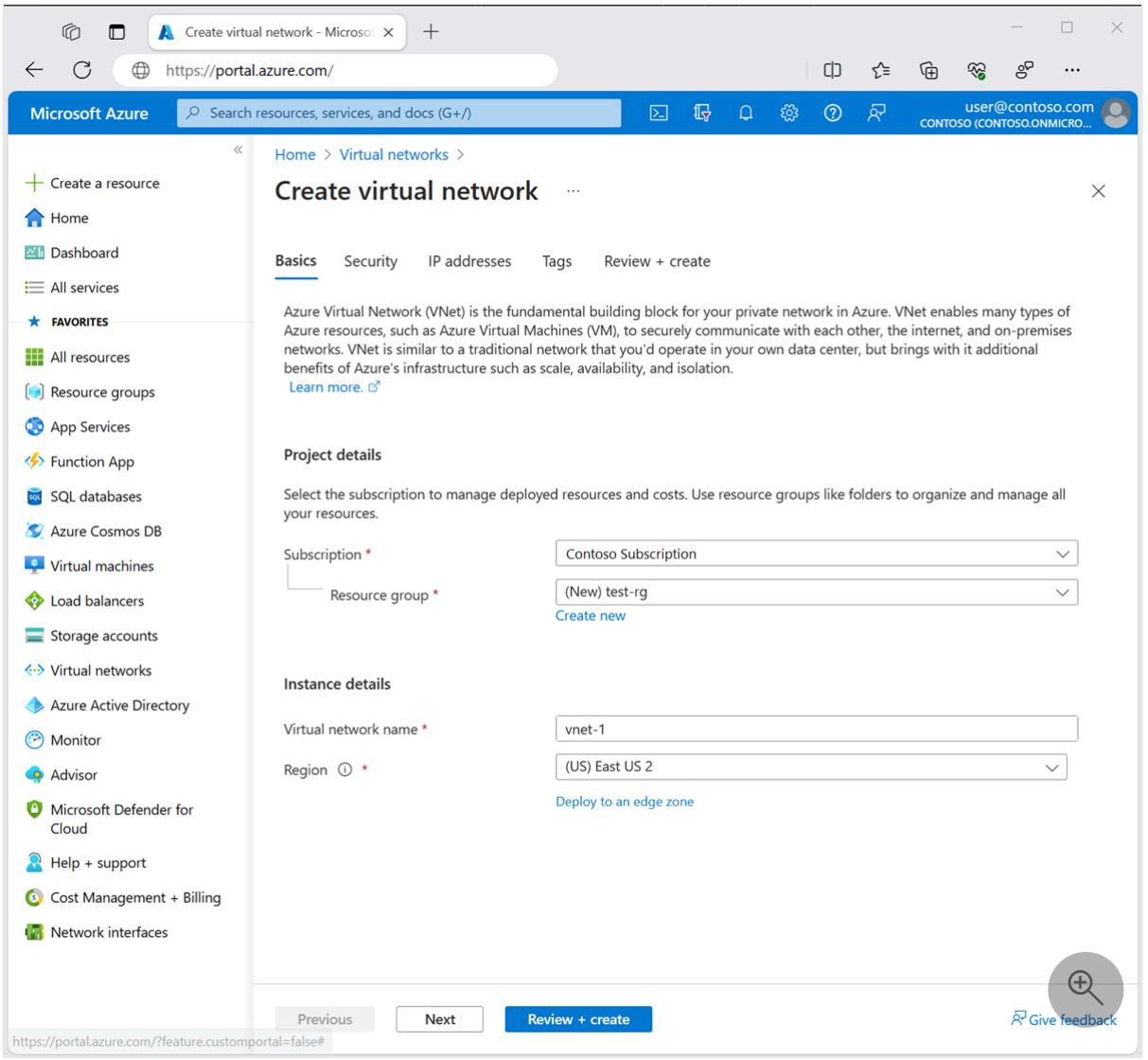
Portal

The following procedure creates a virtual network with a resource subnet.

1. In the portal, search for and select **Virtual networks**.
2. On the **Virtual networks** page, select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter or select the following information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **Create new**. Enter **test-rg** in **Name**. Select **OK**.
Instance details	
Name	Enter **vnet-1**.
Region	Select **East US 2**.



4. Select **Next** to proceed to the **Security** tab.

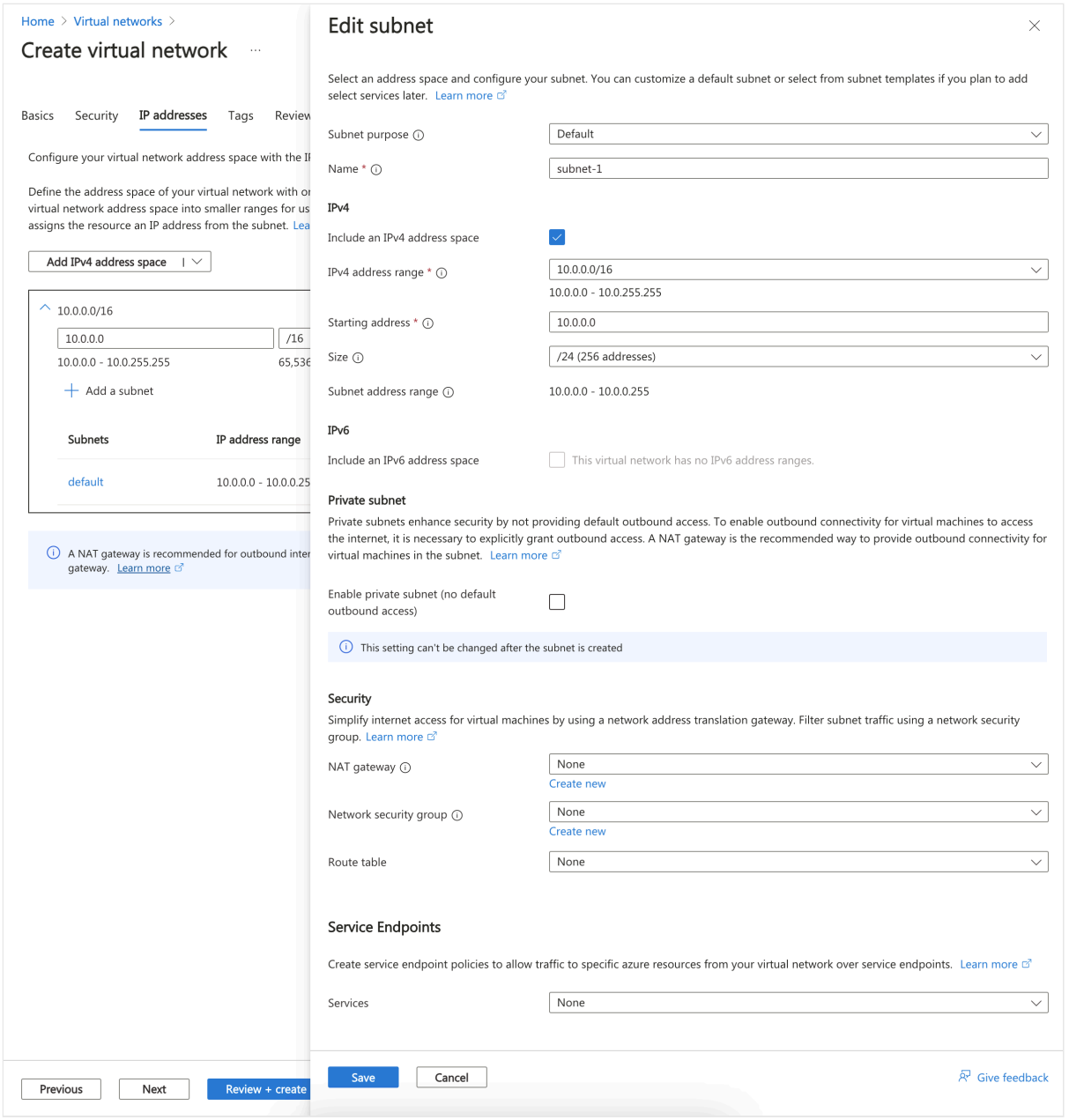
5. Select **Next** to proceed to the **IP addresses** tab.

6. In the address space box under **Subnets**, select the **default** subnet.

7. On the **Edit subnet** pane, enter or select the following information:

Expand table

Setting	Value
Subnet details	
Subnet template	Leave the default as **Default**.
Name	Enter **subnet-1**.
Starting address	Leave the default of **10.0.0.0**.
Subnet size	Leave the default of **/24(256 addresses)**.



8. Select **Save**.

9. Select **Review + create** at the bottom of the screen. After validation passes, select **Create**.

Create application security groups

An application security group (ASGs) enables you to group together servers with similar functions, such as web servers.

Portal

1. In the search box at the top of the portal, enter **Application security group**. Select **Application security groups** in the search results.
2. Select **+ Create**.
3. On the **Basics** tab of **Create an application security group**, enter, or select this information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **asg-web**.
Region	Select **West US 2**.

4. Select **Review + create**.
5. Select **+ Create**.
6. Repeat the previous steps, specifying the following values:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **asg-mgmt**.
Region	Select **West US 2**.

7. Select **Review + create**.

8. Select **Create**.

Create a network security group

A network security group (NSG) secures network traffic in your virtual network.

Portal

1. In the search box at the top of the portal, enter **Network security group**. Select **Network security groups** in the search results.

   > **Note**
   >
   > In the search results for **Network security groups**, you may see **Network security groups (classic)**. Select **Network security groups**.

2. Select **+ Create**.

3. On the **Basics** tab of **Create network security group**, enter, or select this information:

   Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **nsg-1**.
Location	Select **West US 2**.

4. Select **Review + create**.

5. Select **Create**.

Associate network security group to subnet

In this section, you associate the network security group with the subnet of the virtual network you created earlier.

Portal

1. In the search box at the top of the portal, enter **Network security group**. Select **Network security groups** in the search results.
2. Select **nsg-1**.
3. Select **Subnets** from the **Settings** section of **nsg-1**.
4. In the **Subnets** page, select + **Associate**:
5. Under **Associate subnet**, select **vnet-1 (test-rg)** for **Virtual network**.
6. Select **subnet-1** for **Subnet**, and then select **OK**.

Create security rules

Portal

1. Select **Inbound security rules** from the **Settings** section of **nsg-1**.
2. In **Inbound security rules** page, select + **Add**.
3. Create a security rule that allows port 80 to the **asg-web** application security group. In **Add inbound security rule** page, enter or select the following information:

Expand table

Setting	Value
Source	Leave the default of **Any**.
Source port ranges	Leave the default of **(*)**.
Destination	Select **Application security group**.
Destination application security groups	Select **asg-web**.

Setting	Value
Service	Leave the default of **Custom**.
Destination port ranges	Enter **80**.
Protocol	Select **TCP**.
Action	Leave the default of **Allow**.
Priority	Leave the default of **100**.
Name	Enter **allow-http-web**.

4. Select **Add**.

5. Complete the previous steps with the following information:

Expand table

Setting	Value
Source	Leave the default of **Any**.
Source port ranges	Leave the default of **(*)**.
Destination	Select **Application security group**.
Destination application security group	Select **asg-mgmt**.
Service	Leave the default of **Custom**.
Destination port ranges	Enter **8080**.
Protocol	Select **TCP**.
Action	Leave the default of **Allow**.
Priority	Leave the default of **110**.
Name	Enter **allow-8080-mgmt**.

6. Select **Add**.

Create virtual machines

Create two virtual machines (VMs) in the virtual network.

Portal

1. In the portal, search for and select **Virtual machines**.
2. In **Virtual machines**, select + **Create**, then **Virtual machine**.
3. In **Create a virtual machine**, enter or select this information in the **Basics** tab:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Virtual machine name	Enter **vm-web**.
Region	Select **(US) West US 2**.
Availability options	Leave the default of **No infrastructure redundancy required**.
Security type	Select **Standard**.
Image	Select **Ubuntu Server 24.04 LTS - x64 Gen2**.
Azure Spot instance	Leave the default of unchecked.
Size	Select a size.
Administrator account	
Authentication type	Select **SSH public key**.
Username	Enter **azureuser**.
SSH public key source	Select **Generate new key pair**.
Key pair name	Enter **vm-web-key**.
Inbound port rules	
Select inbound ports	Select **None**.

4. Select **Next: Disks** then **Next: Networking**.

5. In the **Networking** tab, enter or select the following information:

Expand table

Setting	Value
Network interface	
Virtual network	Select **vnet-1**.
Subnet	Select **subnet-1 (10.0.0.0/24)**.
Public IP	Leave the default of a new public IP.
NIC network security group	Select **None**.

6. Select the **Review + create** tab, or select the blue **Review + create** button at the bottom of the page.
7. Select **Create**.
8. When prompted to **Generate new key pair**, select **Download private key and create resource**. The private key is downloaded to your local computer. The VM might take a few minutes to deploy.
9. Repeat the previous steps to create a second virtual machine named **vm-mgmt** with key pair name **vm-mgmt-key**.

Associate network interfaces to an ASG

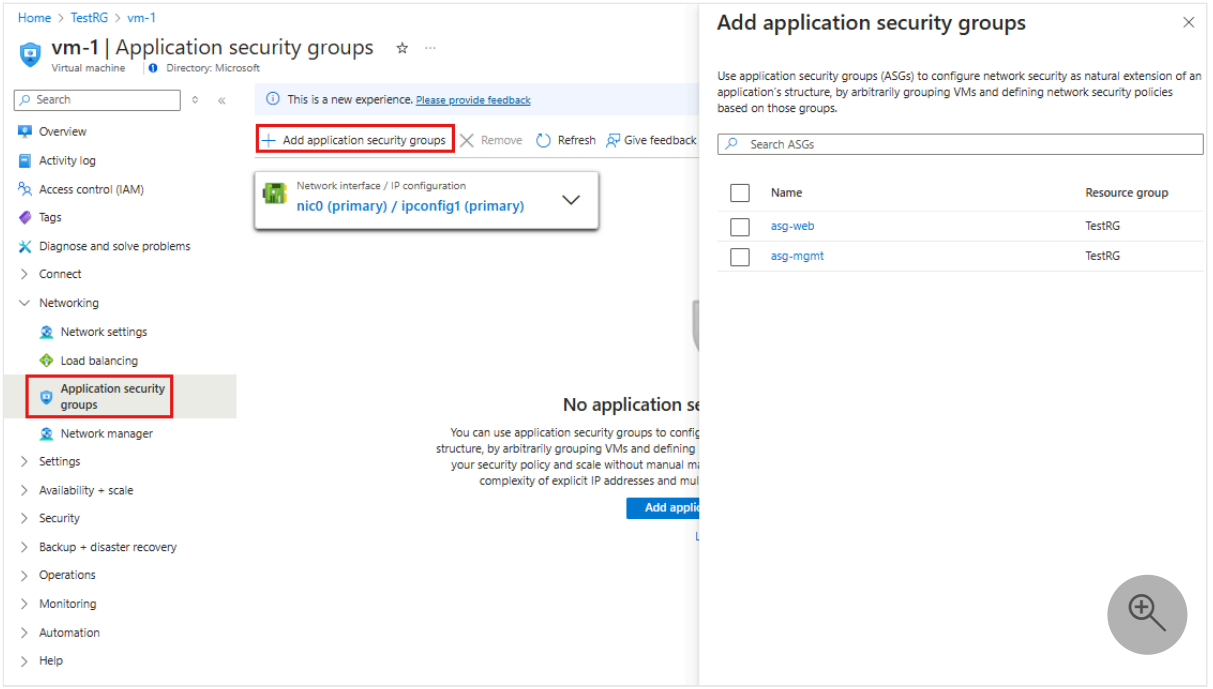
Portal

When you created the VMs, Azure created a network interface for each VM, and attached it to the VM.

Add the network interface of each VM to one of the application security groups you created previously:

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results, then select **vm-web**.
2. Select **Application security groups** from the **Networking** section of **vm-web**.

3. Select **Add application security groups**, then in the **Add application security groups** tab, select **asg-web**. Finally, select **Add**.



4. Repeat the previous steps for **vm-mgmt**, selecting **asg-mgmt** in the **Add application security groups** tab.

Test traffic filters

Portal

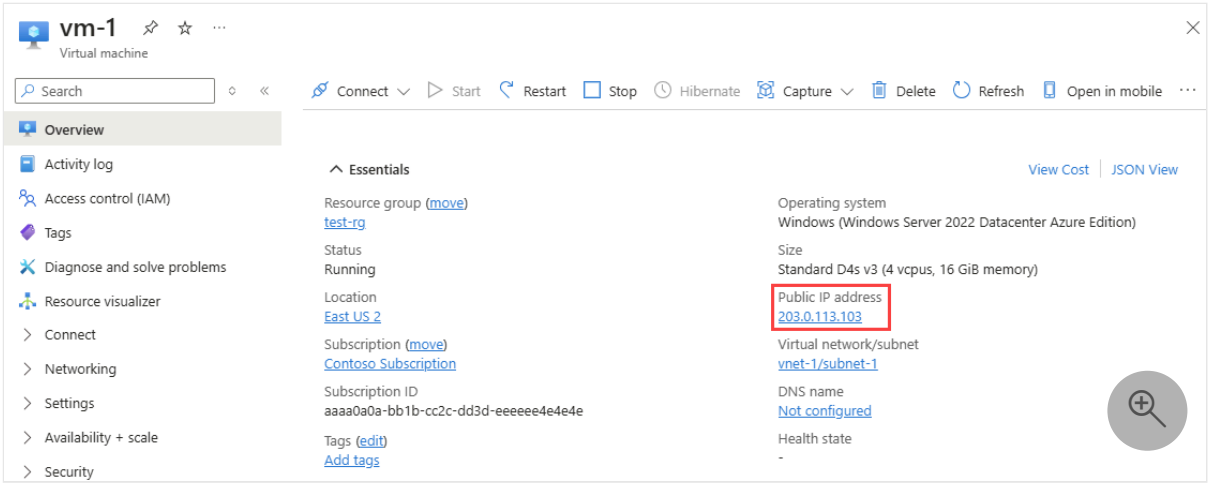
1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **vm-web**.
3. Select **Run command** from the **Operations** section.
4. Select **RunShellScript**.
5. In the **Run Command Script** pane, enter the following commands:

Bash

```
sudo apt-get update -y
sudo apt-get install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
```

6. Select **Run**. Wait for the script to complete successfully.

7. On the **Overview** page of **vm-web**, note the **Public IP address** for your VM.



8. To confirm that you can access the **vm-web** web server from the internet on port 80, open an internet browser on your computer and browse to `http://<public-ip-address-from-previous-step>`.

   You see the nginx default page, because inbound traffic from the internet to the **asg-web** application security group is allowed through port 80.

   The network interface attached for **vm-web** is associated with the **asg-web** application security group and allows the connection.

9. Try to access **vm-web** on port 443 by browsing to `https://<public-ip-address-vm-web>` in your browser. The connection fails or times out because the security rule for the **asg-web** doesn't allow port 443 inbound from the internet.

10. Now configure **vm-mgmt** with nginx on port 8080. Select **vm-mgmt** from the **Virtual machines** page.

11. Select **Run command** from the **Operations** section.

12. Select **RunShellScript**.

13. In the **Run Command Script** pane, enter the following commands to install nginx on port 8080:

```Bash
sudo apt-get update -y
sudo apt-get install -y nginx

# Configure nginx to listen on port 8080
sudo tee /etc/nginx/sites-available/default > /dev/null <<EOF
```

```
server {
    listen 8080 default_server;
    listen [::]:8080 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location / {
        try_files \$uri \$uri/ =404;
    }
}
EOF

sudo systemctl restart nginx
```

14. Select **Run**. Wait for the script to complete successfully.

15. On the **Overview** page of **vm-mgmt**, note the **Public IP address** for your VM.

16. To confirm that you can access the **vm-mgmt** web server from the internet on port 8080, open an internet browser on your computer and browse to `http://<public-ip-address-vm-mgmt>:8080`.

    You see the nginx default page because inbound traffic from the internet to the **asg-mgmt** application security group is allowed through port 8080.

17. Try to access **vm-mgmt** on port 80 by browsing to `http://<public-ip-address-vm-mgmt>` in your browser. The connection fails or times out because no security rule allows port 80 inbound to the **asg-mgmt** application security group.

The network interface attached for **vm-web** is associated with the **asg-web** application security group and allows the connection.

Portal

When you finish using the resources that you created, you can delete the resource group and all its resources.

1. In the Azure portal, search for and select **Resource groups**.

2. On the **Resource groups** page, select the **test-rg** resource group.

3. On the **test-rg** page, select **Delete resource group**.

4. Enter **test-rg** in **Enter resource group name to confirm deletion**, and then select **Delete**.

Next steps

In this tutorial, you:

- Created a network security group and associated it to a virtual network subnet.
- Created application security groups for web and management traffic.
- Created two Linux virtual machines with SSH key authentication and associated their network interfaces with the application security groups.
- Installed nginx web servers on both VMs with different port configurations.
- Tested the application security group network filtering by demonstrating that vm-web allows port 80 (HTTP) but denies port 443, while vm-mgmt allows port 8080 but denies port 80 (HTTP).

To learn more about network security groups, see Network security group overview and Manage a network security group.

Azure routes traffic between subnets by default. You might instead, choose to route traffic between subnets through a VM, serving as a firewall, for example.

To learn how to create a route table, advance to the next tutorial.

Create a route table

ⓘ **Note:** The author created this article with assistance from AI. Learn more

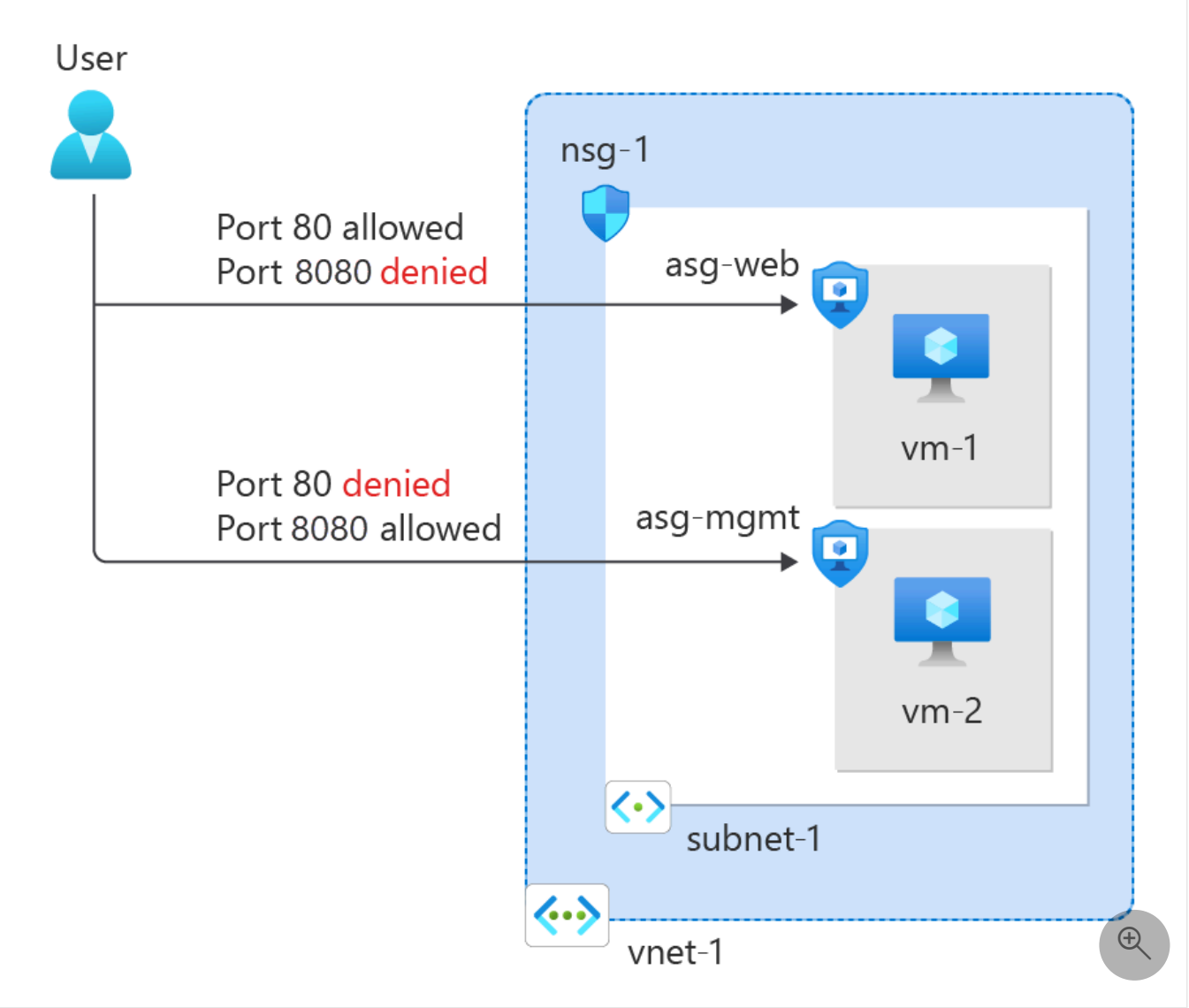
---

Last updated on 02/02/2026

Tutorial: Filter network traffic with a network security group

You can use a network security group to filter inbound and outbound network traffic to and from Azure resources in an Azure virtual network.

Network security groups contain security rules that filter network traffic by IP address, port, and protocol. When a network security group is associated with a subnet, security rules are applied to resources deployed in that subnet.



In this tutorial, you learn how to:

- ✓ Create a network security group and security rules
- ✓ Create application security groups
- ✓ Create a virtual network and associate a network security group to a subnet
- ✓ Deploy virtual machines and associate their network interfaces to the application security groups

Prerequisites

Portal

- An Azure account with an active subscription. You can create an account for free.

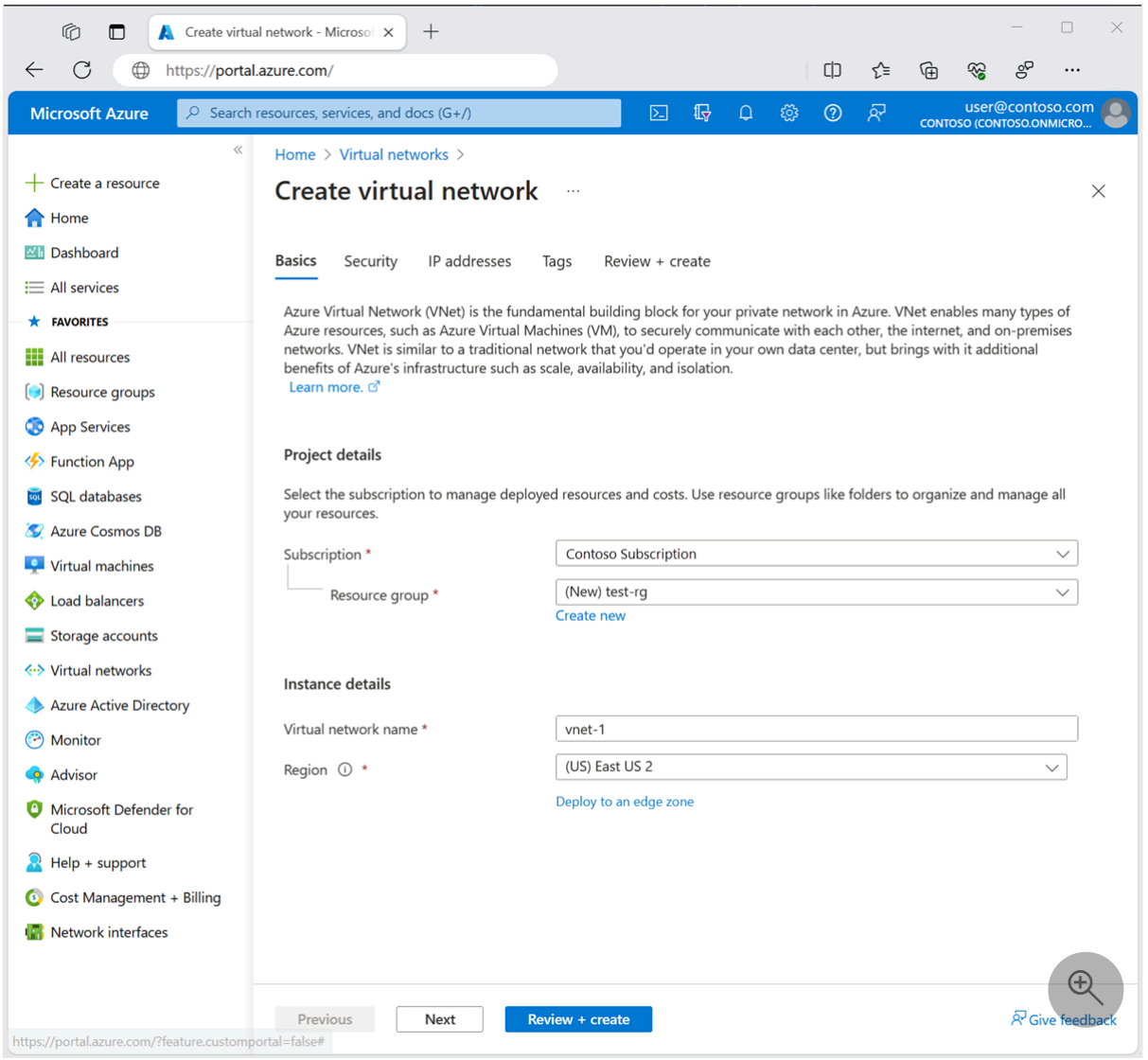
Portal

The following procedure creates a virtual network with a resource subnet.

1. In the portal, search for and select **Virtual networks**.
2. On the **Virtual networks** page, select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter or select the following information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **Create new**. Enter **test-rg** in **Name**. Select **OK**.
Instance details	
Name	Enter **vnet-1**.
Region	Select **East US 2**.



4. Select **Next** to proceed to the **Security** tab.

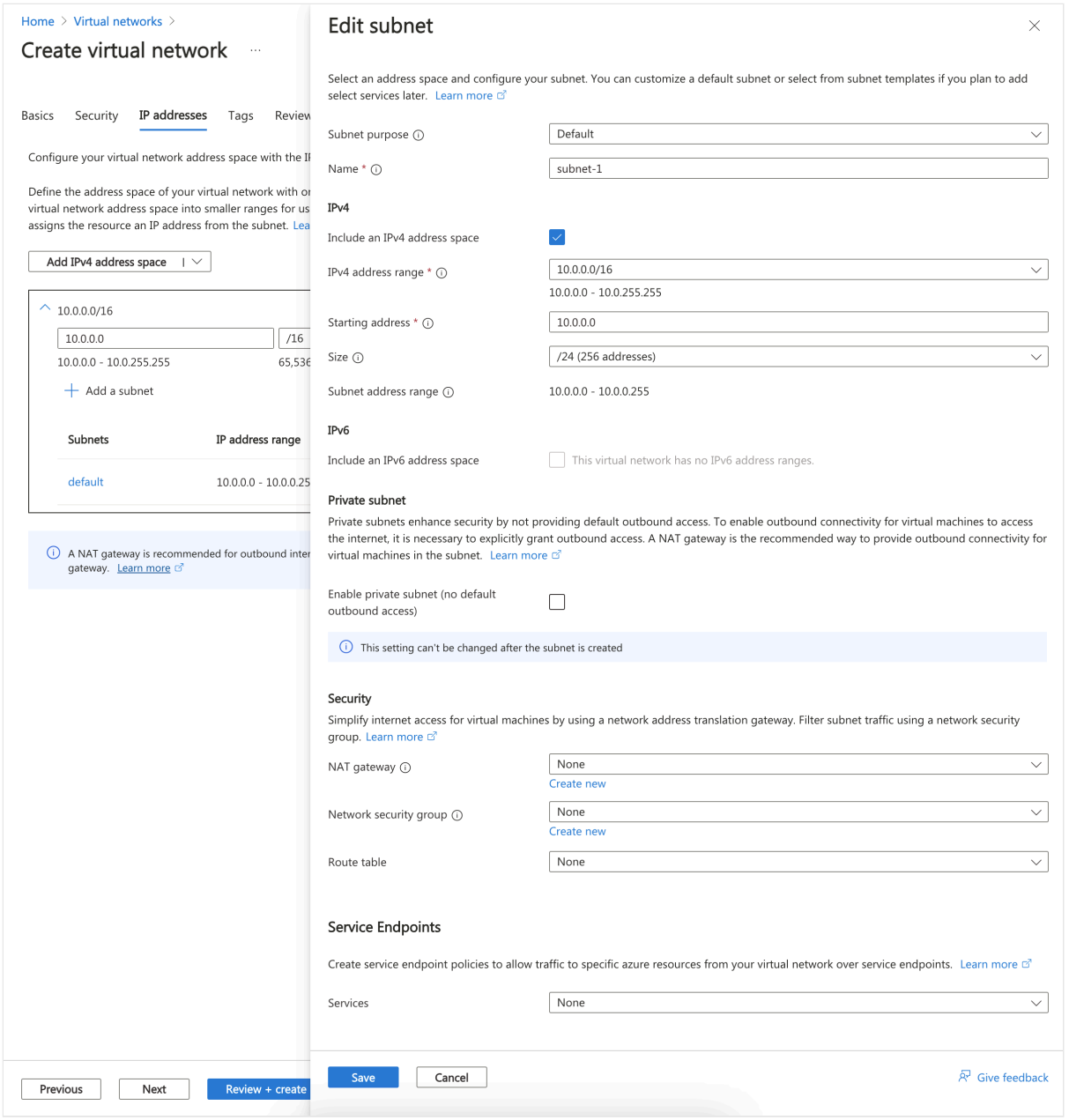
5. Select **Next** to proceed to the **IP addresses** tab.

6. In the address space box under **Subnets**, select the **default** subnet.

7. On the **Edit subnet** pane, enter or select the following information:

Expand table

Setting	Value
Subnet details	
Subnet template	Leave the default as **Default**.
Name	Enter **subnet-1**.
Starting address	Leave the default of **10.0.0.0**.
Subnet size	Leave the default of **/24(256 addresses)**.



8. Select **Save**.

9. Select **Review + create** at the bottom of the screen. After validation passes, select **Create**.

Create application security groups

An application security group (ASGs) enables you to group together servers with similar functions, such as web servers.

Portal

1. In the search box at the top of the portal, enter **Application security group**. Select **Application security groups** in the search results.
2. Select **+ Create**.
3. On the **Basics** tab of **Create an application security group**, enter, or select this information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **asg-web**.
Region	Select **West US 2**.

4. Select **Review + create**.
5. Select **+ Create**.
6. Repeat the previous steps, specifying the following values:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **asg-mgmt**.
Region	Select **West US 2**.

7. Select **Review + create**.

8. Select **Create**.

Create a network security group

A network security group (NSG) secures network traffic in your virtual network.

Portal

1. In the search box at the top of the portal, enter **Network security group**. Select **Network security groups** in the search results.

   > **Note**
   >
   > In the search results for **Network security groups**, you may see **Network security groups (classic)**. Select **Network security groups**.

2. Select **+ Create**.

3. On the **Basics** tab of **Create network security group**, enter, or select this information:

   Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **nsg-1**.
Location	Select **West US 2**.

4. Select **Review + create**.

5. Select **Create**.

Associate network security group to subnet

In this section, you associate the network security group with the subnet of the virtual network you created earlier.

Portal

1. In the search box at the top of the portal, enter **Network security group**. Select **Network security groups** in the search results.
2. Select **nsg-1**.
3. Select **Subnets** from the **Settings** section of **nsg-1**.
4. In the **Subnets** page, select + **Associate**:
5. Under **Associate subnet**, select **vnet-1 (test-rg)** for **Virtual network**.
6. Select **subnet-1** for **Subnet**, and then select **OK**.

Create security rules

Portal

1. Select **Inbound security rules** from the **Settings** section of **nsg-1**.
2. In **Inbound security rules** page, select + **Add**.
3. Create a security rule that allows port 80 to the **asg-web** application security group. In **Add inbound security rule** page, enter or select the following information:

Expand table

Setting	Value
Source	Leave the default of **Any**.
Source port ranges	Leave the default of **(*)**.
Destination	Select **Application security group**.
Destination application security groups	Select **asg-web**.

Setting	Value
Service	Leave the default of **Custom**.
Destination port ranges	Enter **80**.
Protocol	Select **TCP**.
Action	Leave the default of **Allow**.
Priority	Leave the default of **100**.
Name	Enter **allow-http-web**.

4. Select **Add**.

5. Complete the previous steps with the following information:

Expand table

Setting	Value
Source	Leave the default of **Any**.
Source port ranges	Leave the default of **(*)**.
Destination	Select **Application security group**.
Destination application security group	Select **asg-mgmt**.
Service	Leave the default of **Custom**.
Destination port ranges	Enter **8080**.
Protocol	Select **TCP**.
Action	Leave the default of **Allow**.
Priority	Leave the default of **110**.
Name	Enter **allow-8080-mgmt**.

6. Select **Add**.

Create virtual machines

Create two virtual machines (VMs) in the virtual network.

Portal

1. In the portal, search for and select **Virtual machines**.
2. In **Virtual machines**, select + **Create**, then **Virtual machine**.
3. In **Create a virtual machine**, enter or select this information in the **Basics** tab:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Virtual machine name	Enter **vm-web**.
Region	Select **(US) West US 2**.
Availability options	Leave the default of **No infrastructure redundancy required**.
Security type	Select **Standard**.
Image	Select **Ubuntu Server 24.04 LTS - x64 Gen2**.
Azure Spot instance	Leave the default of unchecked.
Size	Select a size.
Administrator account	
Authentication type	Select **SSH public key**.
Username	Enter **azureuser**.
SSH public key source	Select **Generate new key pair**.
Key pair name	Enter **vm-web-key**.
Inbound port rules	
Select inbound ports	Select **None**.

4. Select **Next: Disks** then **Next: Networking**.

5. In the **Networking** tab, enter or select the following information:

Expand table

Setting	Value
Network interface	
Virtual network	Select **vnet-1**.
Subnet	Select **subnet-1 (10.0.0.0/24)**.
Public IP	Leave the default of a new public IP.
NIC network security group	Select **None**.

6. Select the **Review + create** tab, or select the blue **Review + create** button at the bottom of the page.
7. Select **Create**.
8. When prompted to **Generate new key pair**, select **Download private key and create resource**. The private key is downloaded to your local computer. The VM might take a few minutes to deploy.
9. Repeat the previous steps to create a second virtual machine named **vm-mgmt** with key pair name **vm-mgmt-key**.

Associate network interfaces to an ASG

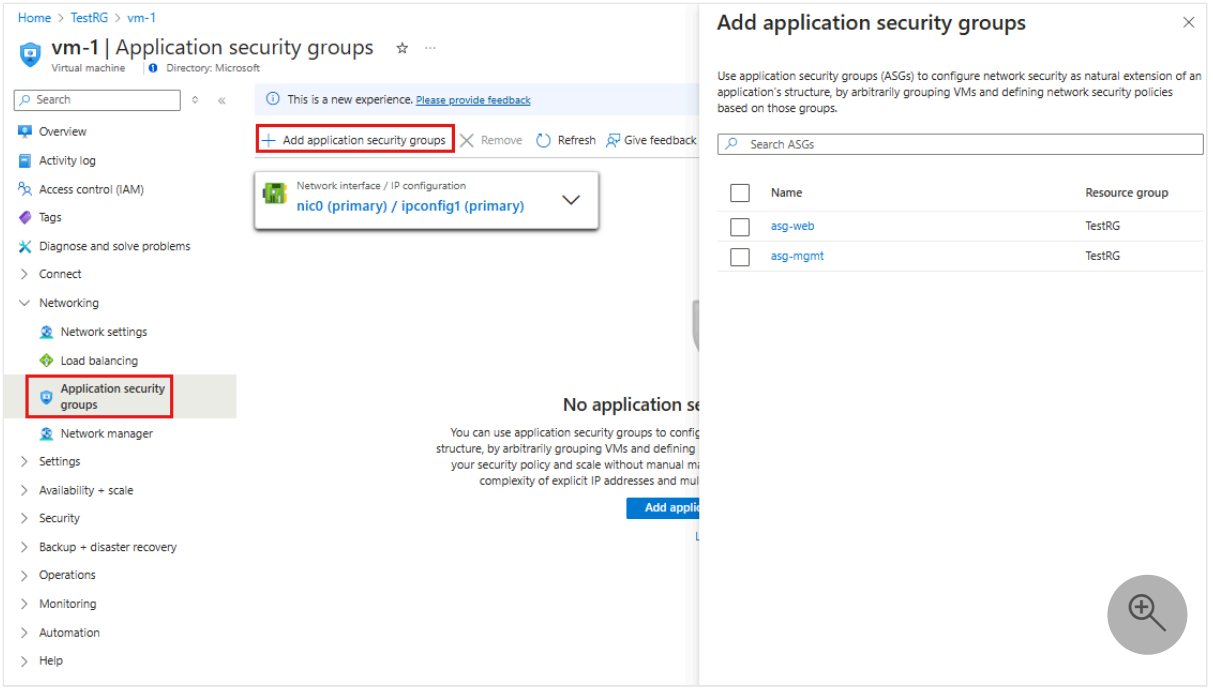
Portal

When you created the VMs, Azure created a network interface for each VM, and attached it to the VM.

Add the network interface of each VM to one of the application security groups you created previously:

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results, then select **vm-web**.
2. Select **Application security groups** from the **Networking** section of **vm-web**.

3. Select **Add application security groups**, then in the **Add application security groups** tab, select **asg-web**. Finally, select **Add**.



4. Repeat the previous steps for **vm-mgmt**, selecting **asg-mgmt** in the **Add application security groups** tab.

Test traffic filters

Portal

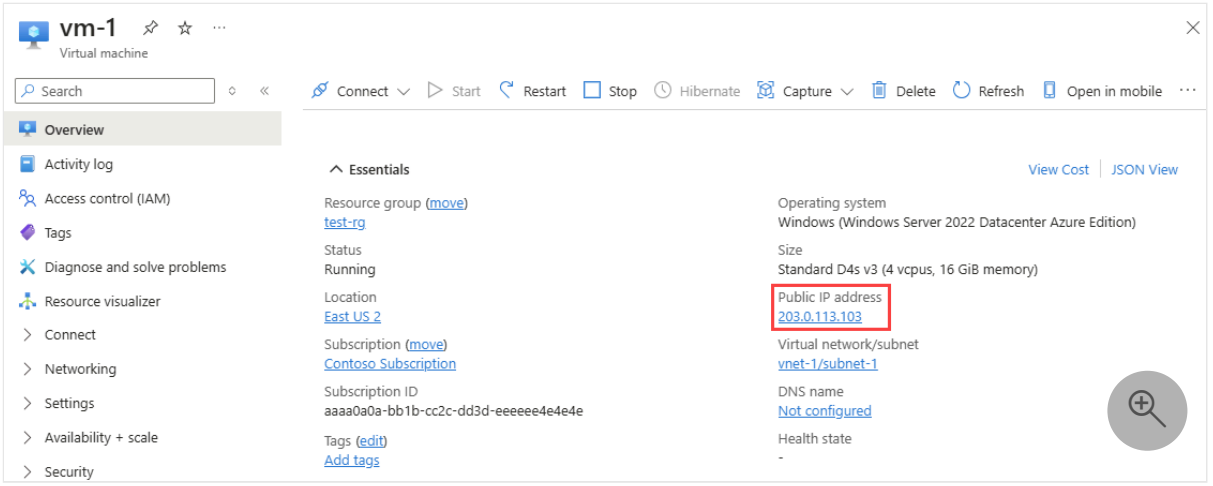
1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **vm-web**.
3. Select **Run command** from the **Operations** section.
4. Select **RunShellScript**.
5. In the **Run Command Script** pane, enter the following commands:

Bash

```
sudo apt-get update -y
sudo apt-get install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
```

6. Select **Run**. Wait for the script to complete successfully.

7. On the **Overview** page of **vm-web**, note the **Public IP address** for your VM.



8. To confirm that you can access the **vm-web** web server from the internet on port 80, open an internet browser on your computer and browse to `http://<public-ip-address-from-previous-step>`.

   You see the nginx default page, because inbound traffic from the internet to the **asg-web** application security group is allowed through port 80.

   The network interface attached for **vm-web** is associated with the **asg-web** application security group and allows the connection.

9. Try to access **vm-web** on port 443 by browsing to `https://<public-ip-address-vm-web>` in your browser. The connection fails or times out because the security rule for the **asg-web** doesn't allow port 443 inbound from the internet.

10. Now configure **vm-mgmt** with nginx on port 8080. Select **vm-mgmt** from the **Virtual machines** page.

11. Select **Run command** from the **Operations** section.

12. Select **RunShellScript**.

13. In the **Run Command Script** pane, enter the following commands to install nginx on port 8080:

```bash
sudo apt-get update -y
sudo apt-get install -y nginx

# Configure nginx to listen on port 8080
sudo tee /etc/nginx/sites-available/default > /dev/null <<EOF
```

```
server {
    listen 8080 default_server;
    listen [::]:8080 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location / {
        try_files \$uri \$uri/ =404;
    }
}
EOF

sudo systemctl restart nginx
```

14. Select **Run**. Wait for the script to complete successfully.

15. On the **Overview** page of **vm-mgmt**, note the **Public IP address** for your VM.

16. To confirm that you can access the **vm-mgmt** web server from the internet on port 8080, open an internet browser on your computer and browse to `http://<public-ip-address-vm-mgmt>:8080`.

    You see the nginx default page because inbound traffic from the internet to the **asg-mgmt** application security group is allowed through port 8080.

17. Try to access **vm-mgmt** on port 80 by browsing to `http://<public-ip-address-vm-mgmt>` in your browser. The connection fails or times out because no security rule allows port 80 inbound to the **asg-mgmt** application security group.

The network interface attached for **vm-web** is associated with the **asg-web** application security group and allows the connection.

Portal

When you finish using the resources that you created, you can delete the resource group and all its resources.

1. In the Azure portal, search for and select **Resource groups**.

2. On the **Resource groups** page, select the **test-rg** resource group.

3. On the **test-rg** page, select **Delete resource group**.

4. Enter **test-rg** in **Enter resource group name to confirm deletion**, and then select **Delete**.

Next steps

In this tutorial, you:

- Created a network security group and associated it to a virtual network subnet.
- Created application security groups for web and management traffic.
- Created two Linux virtual machines with SSH key authentication and associated their network interfaces with the application security groups.
- Installed nginx web servers on both VMs with different port configurations.
- Tested the application security group network filtering by demonstrating that vm-web allows port 80 (HTTP) but denies port 443, while vm-mgmt allows port 8080 but denies port 80 (HTTP).

To learn more about network security groups, see Network security group overview and Manage a network security group.

Azure routes traffic between subnets by default. You might instead, choose to route traffic between subnets through a VM, serving as a firewall, for example.

To learn how to create a route table, advance to the next tutorial.

Create a route table

ⓘ **Note:** The author created this article with assistance from AI. Learn more

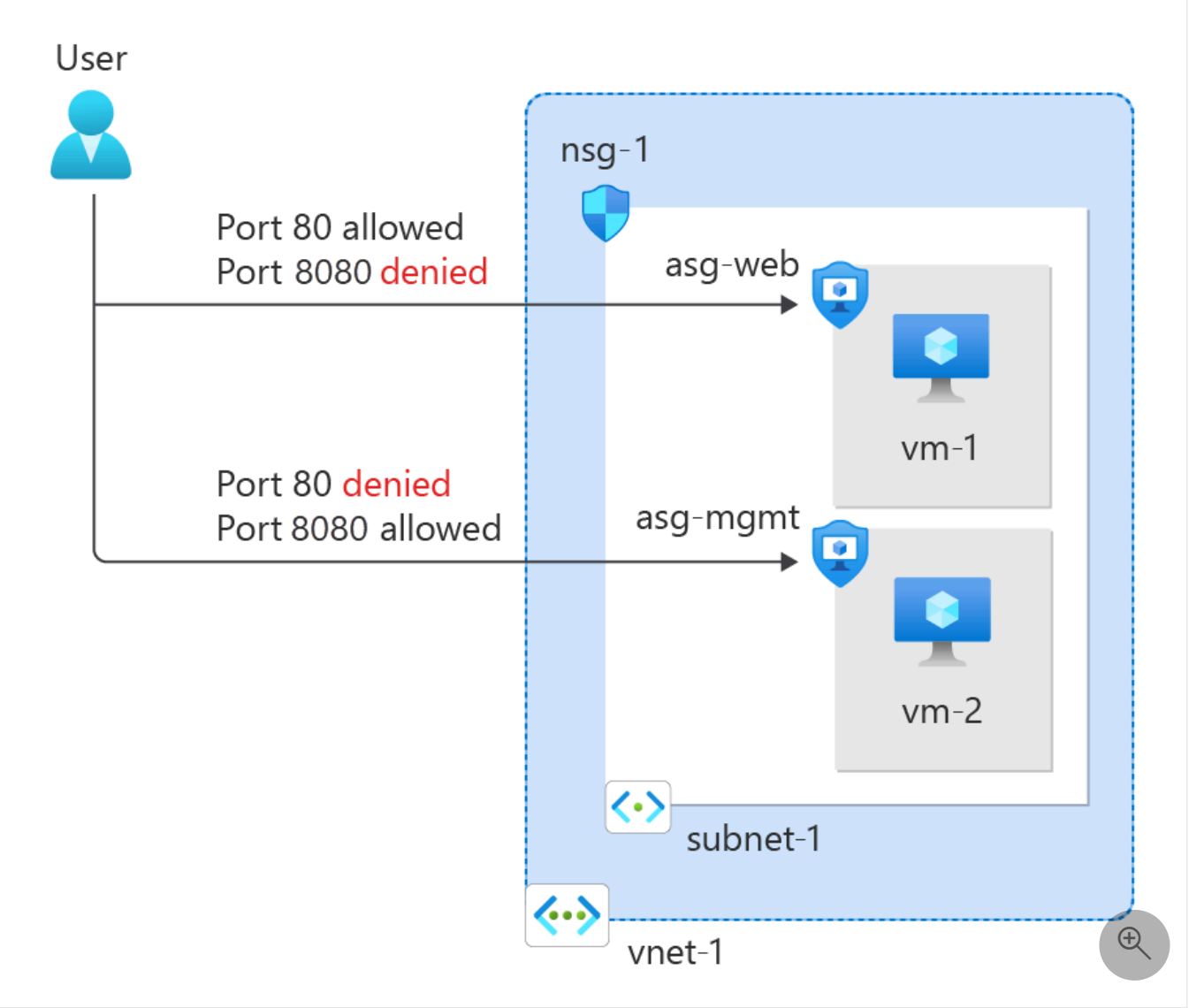
---

Last updated on 02/02/2026

Tutorial: Filter network traffic with a network security group

You can use a network security group to filter inbound and outbound network traffic to and from Azure resources in an Azure virtual network.

Network security groups contain security rules that filter network traffic by IP address, port, and protocol. When a network security group is associated with a subnet, security rules are applied to resources deployed in that subnet.



In this tutorial, you learn how to:

- ✓ Create a network security group and security rules
- ✓ Create application security groups
- ✓ Create a virtual network and associate a network security group to a subnet
- ✓ Deploy virtual machines and associate their network interfaces to the application security groups

Prerequisites

Portal

- An Azure account with an active subscription. You can create an account for free.

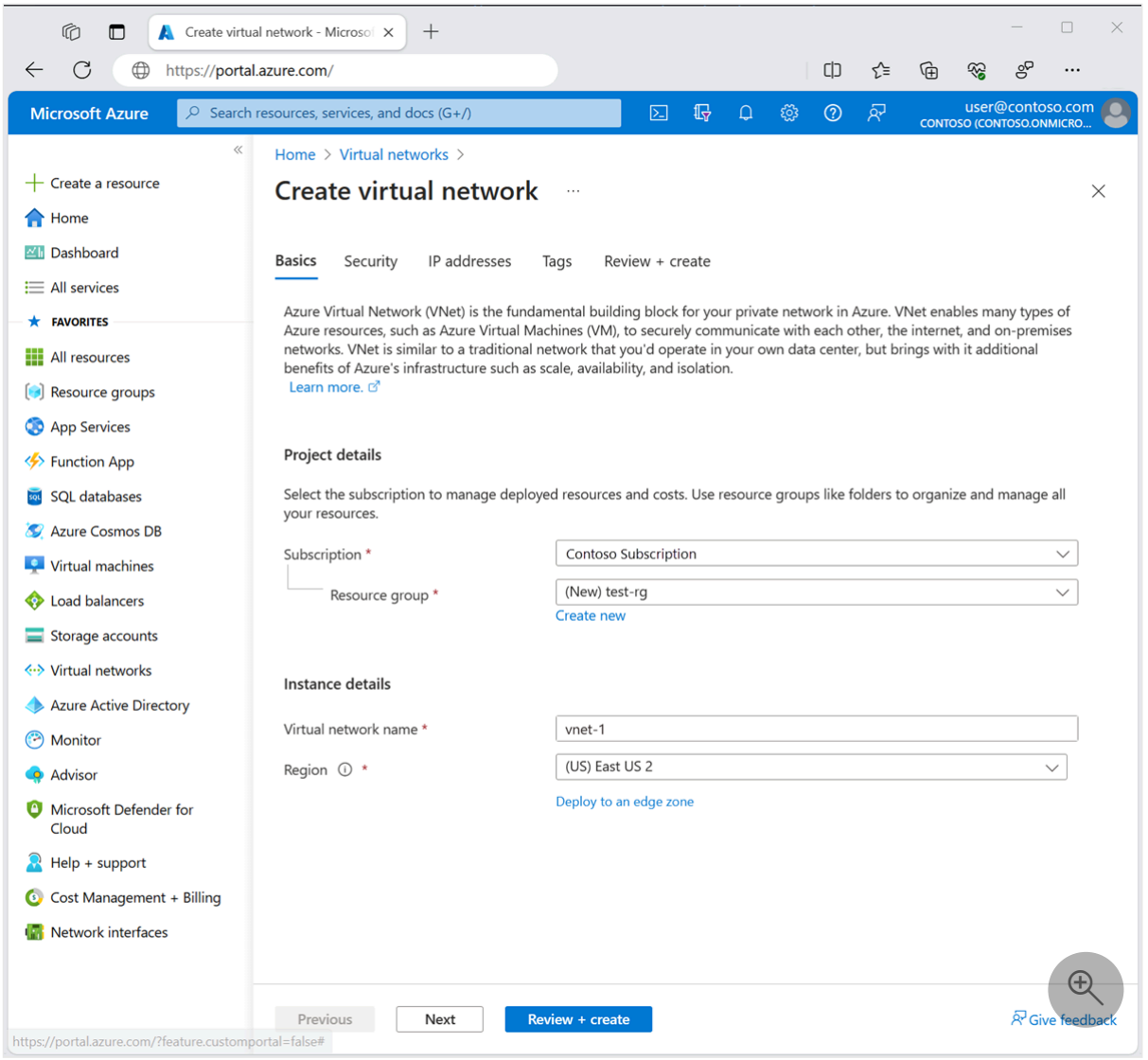
Portal

The following procedure creates a virtual network with a resource subnet.

1. In the portal, search for and select **Virtual networks**.
2. On the **Virtual networks** page, select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter or select the following information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **Create new**. Enter **test-rg** in **Name**. Select **OK**.
Instance details	
Name	Enter **vnet-1**.
Region	Select **East US 2**.



4. Select **Next** to proceed to the **Security** tab.

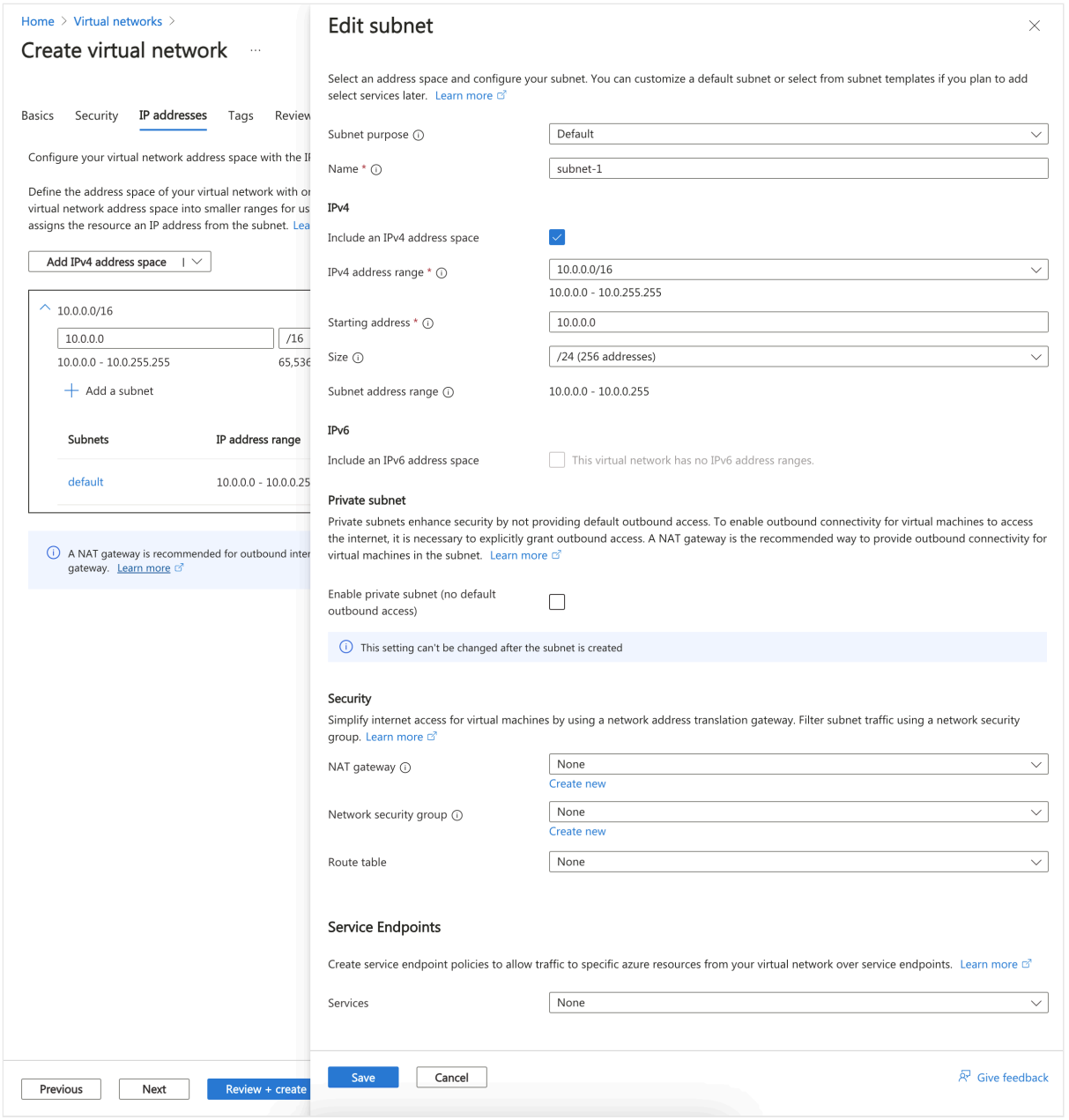
5. Select **Next** to proceed to the **IP addresses** tab.

6. In the address space box under **Subnets**, select the **default** subnet.

7. On the **Edit subnet** pane, enter or select the following information:

Expand table

Setting	Value
Subnet details	
Subnet template	Leave the default as **Default**.
Name	Enter **subnet-1**.
Starting address	Leave the default of **10.0.0.0**.
Subnet size	Leave the default of **/24(256 addresses)**.



8. Select **Save**.

9. Select **Review + create** at the bottom of the screen. After validation passes, select **Create**.

Create application security groups

An application security group (ASGs) enables you to group together servers with similar functions, such as web servers.

Portal

1. In the search box at the top of the portal, enter **Application security group**. Select **Application security groups** in the search results.
2. Select **+ Create**.
3. On the **Basics** tab of **Create an application security group**, enter, or select this information:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **asg-web**.
Region	Select **West US 2**.

4. Select **Review + create**.
5. Select **+ Create**.
6. Repeat the previous steps, specifying the following values:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **asg-mgmt**.
Region	Select **West US 2**.

7. Select **Review + create**.

8. Select **Create**.

Create a network security group

A network security group (NSG) secures network traffic in your virtual network.

Portal

1. In the search box at the top of the portal, enter **Network security group**. Select **Network security groups** in the search results.

   > **Note**
   >
   > In the search results for **Network security groups**, you may see **Network security groups (classic)**. Select **Network security groups**.

2. Select **+ Create**.

3. On the **Basics** tab of **Create network security group**, enter, or select this information:

   Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Name	Enter **nsg-1**.
Location	Select **West US 2**.

4. Select **Review + create**.

5. Select **Create**.

Associate network security group to subnet

In this section, you associate the network security group with the subnet of the virtual network you created earlier.

Portal

1. In the search box at the top of the portal, enter **Network security group**. Select **Network security groups** in the search results.
2. Select **nsg-1**.
3. Select **Subnets** from the **Settings** section of **nsg-1**.
4. In the **Subnets** page, select + **Associate**:
5. Under **Associate subnet**, select **vnet-1 (test-rg)** for **Virtual network**.
6. Select **subnet-1** for **Subnet**, and then select **OK**.

Create security rules

Portal

1. Select **Inbound security rules** from the **Settings** section of **nsg-1**.
2. In **Inbound security rules** page, select + **Add**.
3. Create a security rule that allows port 80 to the **asg-web** application security group. In **Add inbound security rule** page, enter or select the following information:

Expand table

Setting	Value
Source	Leave the default of **Any**.
Source port ranges	Leave the default of **(*)**.
Destination	Select **Application security group**.
Destination application security groups	Select **asg-web**.

Setting	Value
Service	Leave the default of **Custom**.
Destination port ranges	Enter **80**.
Protocol	Select **TCP**.
Action	Leave the default of **Allow**.
Priority	Leave the default of **100**.
Name	Enter **allow-http-web**.

4. Select **Add**.

5. Complete the previous steps with the following information:

Expand table

Setting	Value
Source	Leave the default of **Any**.
Source port ranges	Leave the default of **(*)**.
Destination	Select **Application security group**.
Destination application security group	Select **asg-mgmt**.
Service	Leave the default of **Custom**.
Destination port ranges	Enter **8080**.
Protocol	Select **TCP**.
Action	Leave the default of **Allow**.
Priority	Leave the default of **110**.
Name	Enter **allow-8080-mgmt**.

6. Select **Add**.

Create virtual machines

Create two virtual machines (VMs) in the virtual network.

Portal

1. In the portal, search for and select **Virtual machines**.
2. In **Virtual machines**, select + **Create**, then **Virtual machine**.
3. In **Create a virtual machine**, enter or select this information in the **Basics** tab:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Instance details	
Virtual machine name	Enter **vm-web**.
Region	Select **(US) West US 2**.
Availability options	Leave the default of **No infrastructure redundancy required**.
Security type	Select **Standard**.
Image	Select **Ubuntu Server 24.04 LTS - x64 Gen2**.
Azure Spot instance	Leave the default of unchecked.
Size	Select a size.
Administrator account	
Authentication type	Select **SSH public key**.
Username	Enter **azureuser**.
SSH public key source	Select **Generate new key pair**.
Key pair name	Enter **vm-web-key**.
Inbound port rules	
Select inbound ports	Select **None**.

4. Select **Next: Disks** then **Next: Networking**.

5. In the **Networking** tab, enter or select the following information:

Expand table

Setting	Value
Network interface	
Virtual network	Select **vnet-1**.
Subnet	Select **subnet-1 (10.0.0.0/24)**.
Public IP	Leave the default of a new public IP.
NIC network security group	Select **None**.

6. Select the **Review + create** tab, or select the blue **Review + create** button at the bottom of the page.
7. Select **Create**.
8. When prompted to **Generate new key pair**, select **Download private key and create resource**. The private key is downloaded to your local computer. The VM might take a few minutes to deploy.
9. Repeat the previous steps to create a second virtual machine named **vm-mgmt** with key pair name **vm-mgmt-key**.

Associate network interfaces to an ASG

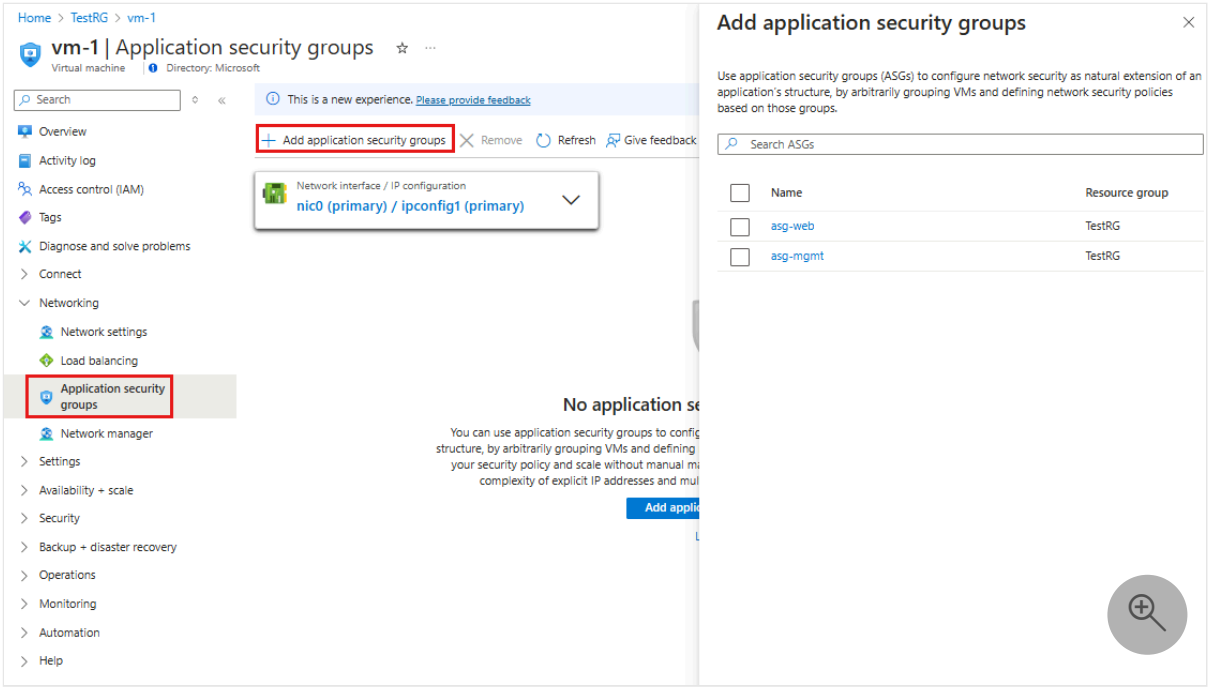
Portal

When you created the VMs, Azure created a network interface for each VM, and attached it to the VM.

Add the network interface of each VM to one of the application security groups you created previously:

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results, then select **vm-web**.
2. Select **Application security groups** from the **Networking** section of **vm-web**.

3. Select **Add application security groups**, then in the **Add application security groups** tab, select **asg-web**. Finally, select **Add**.



4. Repeat the previous steps for **vm-mgmt**, selecting **asg-mgmt** in the **Add application security groups** tab.

Test traffic filters

Portal

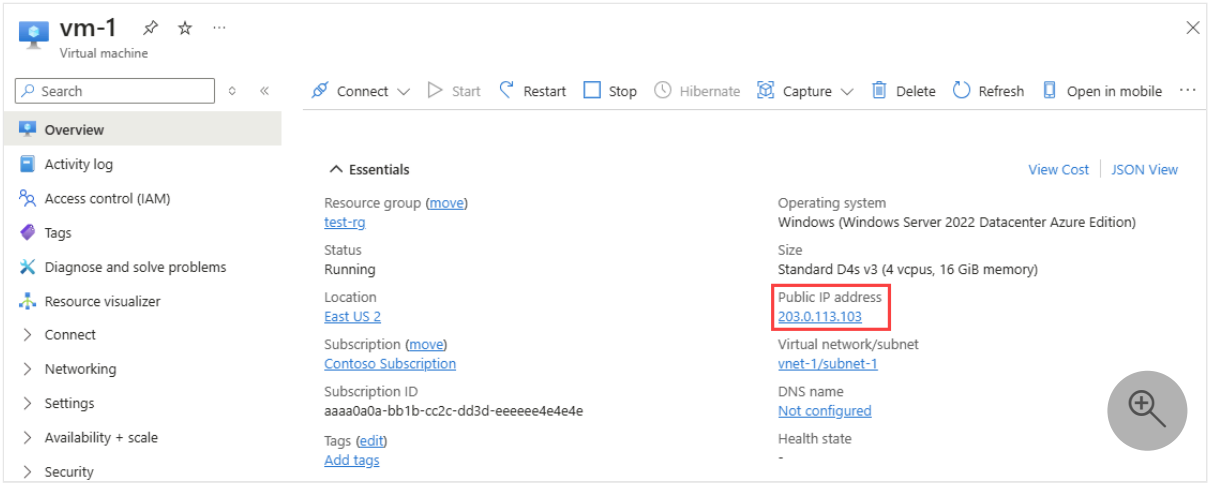
1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **vm-web**.
3. Select **Run command** from the **Operations** section.
4. Select **RunShellScript**.
5. In the **Run Command Script** pane, enter the following commands:

Bash

```
sudo apt-get update -y
sudo apt-get install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
```

6. Select **Run**. Wait for the script to complete successfully.

7. On the **Overview** page of **vm-web**, note the **Public IP address** for your VM.



8. To confirm that you can access the **vm-web** web server from the internet on port 80, open an internet browser on your computer and browse to `http://<public-ip-address-from-previous-step>`.

   You see the nginx default page, because inbound traffic from the internet to the **asg-web** application security group is allowed through port 80.

   The network interface attached for **vm-web** is associated with the **asg-web** application security group and allows the connection.

9. Try to access **vm-web** on port 443 by browsing to `https://<public-ip-address-vm-web>` in your browser. The connection fails or times out because the security rule for the **asg-web** doesn't allow port 443 inbound from the internet.

10. Now configure **vm-mgmt** with nginx on port 8080. Select **vm-mgmt** from the **Virtual machines** page.

11. Select **Run command** from the **Operations** section.

12. Select **RunShellScript**.

13. In the **Run Command Script** pane, enter the following commands to install nginx on port 8080:

```bash
sudo apt-get update -y
sudo apt-get install -y nginx

# Configure nginx to listen on port 8080
sudo tee /etc/nginx/sites-available/default > /dev/null <<EOF
```

```
server {
    listen 8080 default_server;
    listen [::]:8080 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location / {
        try_files \$uri \$uri/ =404;
    }
}
EOF

sudo systemctl restart nginx
```

14. Select **Run**. Wait for the script to complete successfully.

15. On the **Overview** page of **vm-mgmt**, note the **Public IP address** for your VM.

16. To confirm that you can access the **vm-mgmt** web server from the internet on port 8080, open an internet browser on your computer and browse to `http://<public-ip-address-vm-mgmt>:8080`.

    You see the nginx default page because inbound traffic from the internet to the **asg-mgmt** application security group is allowed through port 8080.

17. Try to access **vm-mgmt** on port 80 by browsing to `http://<public-ip-address-vm-mgmt>` in your browser. The connection fails or times out because no security rule allows port 80 inbound to the **asg-mgmt** application security group.

The network interface attached for **vm-web** is associated with the **asg-web** application security group and allows the connection.

Portal

When you finish using the resources that you created, you can delete the resource group and all its resources.

1. In the Azure portal, search for and select **Resource groups**.

2. On the **Resource groups** page, select the **test-rg** resource group.

3. On the **test-rg** page, select **Delete resource group**.

4. Enter **test-rg** in **Enter resource group name to confirm deletion**, and then select **Delete**.

Next steps

In this tutorial, you:

- Created a network security group and associated it to a virtual network subnet.
- Created application security groups for web and management traffic.
- Created two Linux virtual machines with SSH key authentication and associated their network interfaces with the application security groups.
- Installed nginx web servers on both VMs with different port configurations.
- Tested the application security group network filtering by demonstrating that vm-web allows port 80 (HTTP) but denies port 443, while vm-mgmt allows port 8080 but denies port 80 (HTTP).

To learn more about network security groups, see Network security group overview and Manage a network security group.

Azure routes traffic between subnets by default. You might instead, choose to route traffic between subnets through a VM, serving as a firewall, for example.

To learn how to create a route table, advance to the next tutorial.

Create a route table

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 02/02/2026

Network security group test

Article • 09/20/2024

You can use network security groups to filter and control inbound and outbound network traffic to and from your virtual machines (VMs). You can also use Azure Virtual Network Manager to apply admin security rules to your VMs to control network traffic.

In this article, you learn how to use **Network security group test** to check if a security rule is blocking traffic to or from your virtual machine by checking what security rules are applied to your VM traffic.

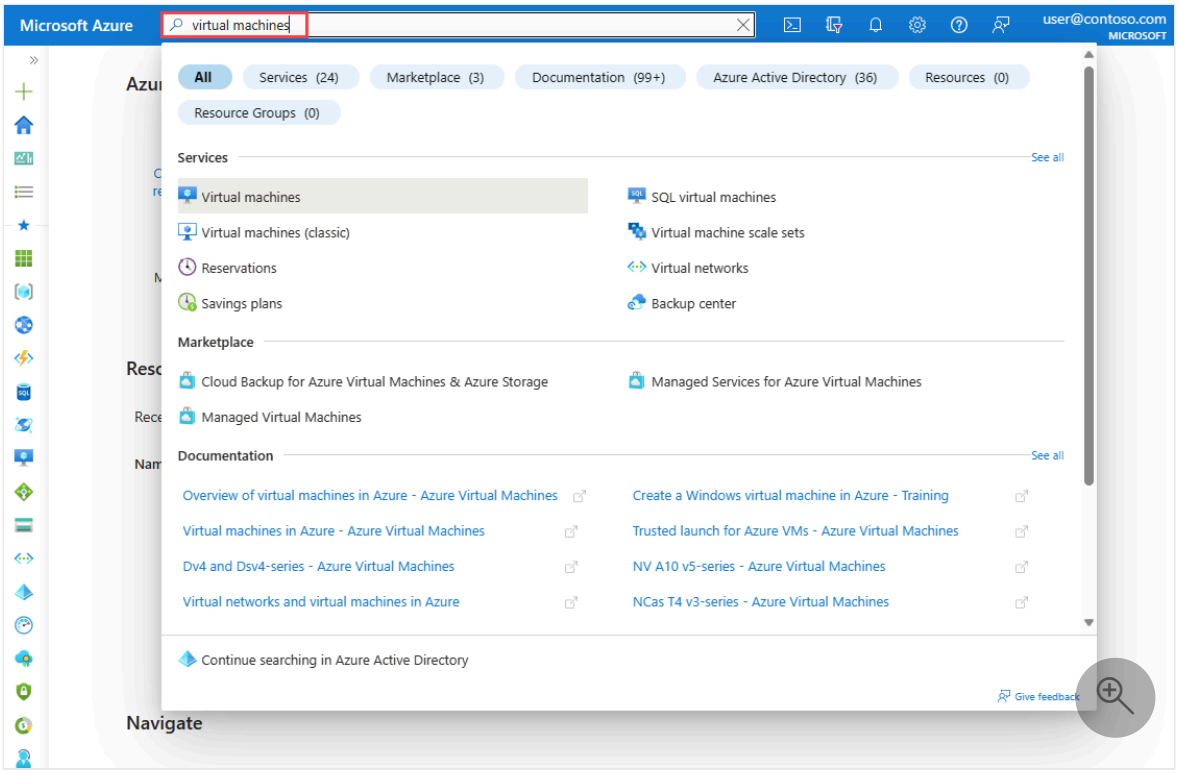
Prerequisites

- An Azure account with an active subscription. If you don't have one, create a free account before you begin.
- Sign in to the Azure portal with your Azure account.
- An Azure virtual machine (VM). If you don't have one, create a Linux VM or a Windows VM.

Test inbound connections

In this section, you test if RDP connections are allowed to your VM from a remote IP address.

1. In the search box at the top of the portal, enter *virtual machines*. Select **Virtual machines** from the search results.

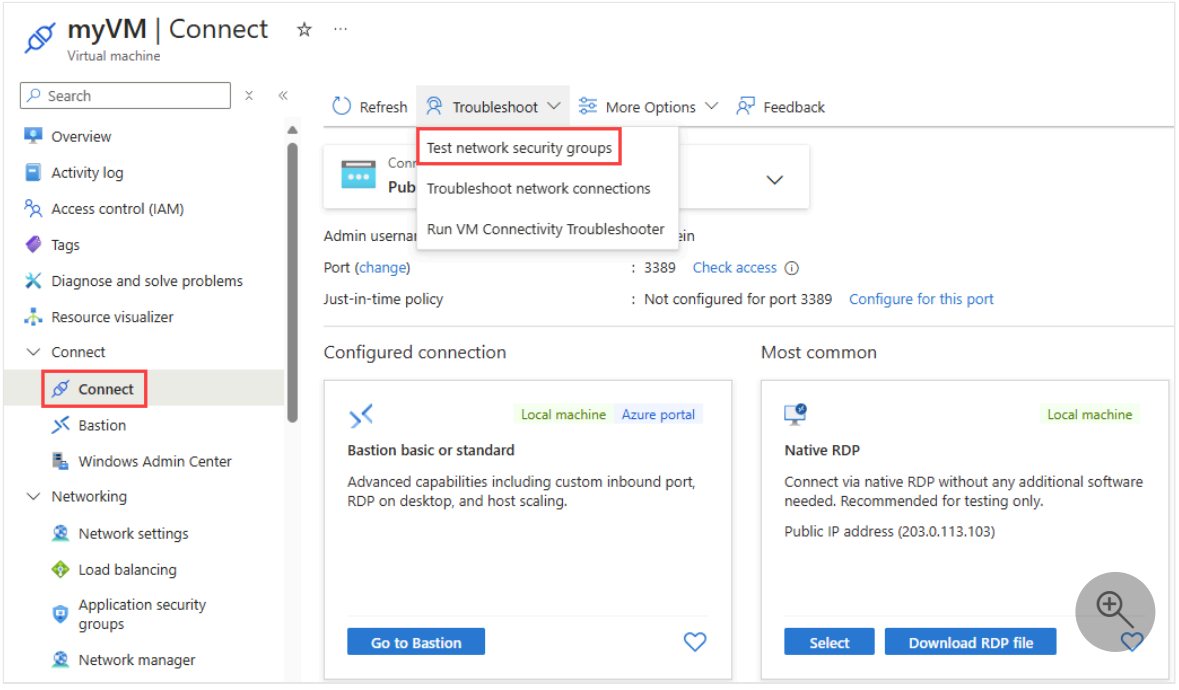


2. Select the VM that you want to test.

3. Under **Connect**, select **Connect**.

> **Note**
>
> The virtual machine must be in running state.

4. Under **Troubleshoot**, select **Test network security groups**.



5. Select **Inbound connections**. The following options are available for **Inbound** tests:

Expand table

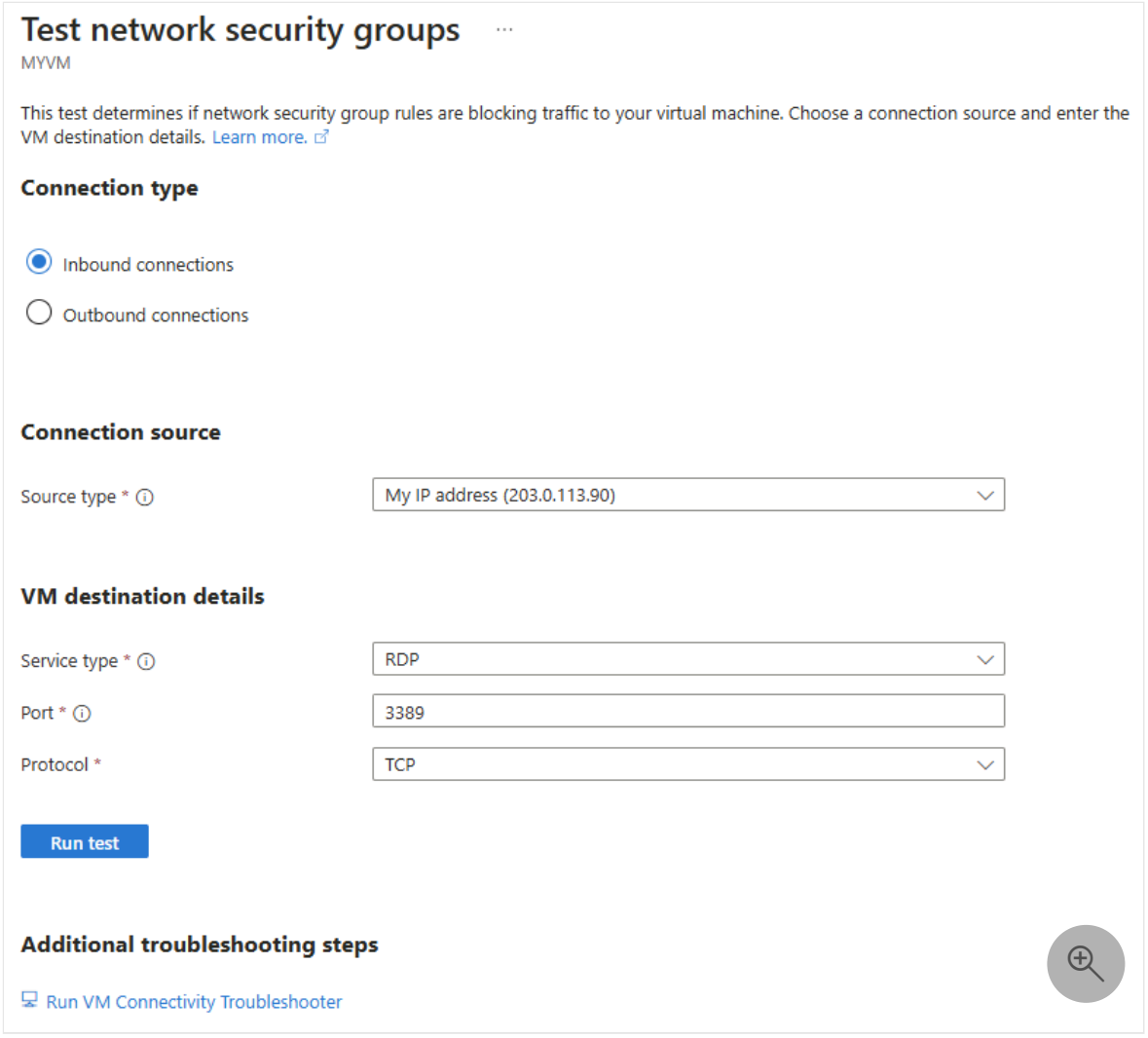
Setting	Value
Source type	- **My IP address**: your public IP address that you're using to access the Azure portal. - **Any IP address**: any source IP address. - **Other IP address/CIDR**: Source IP address or address prefix. - **Service tag**: Source service tag.
IP address/CIDR	The IP address or address prefix that you want to use as the source. **Note**: You see this option if you select **Other IP address/CIDR** for **Source type**.
Service tag	The service tag that you want to use as the source. **Note**: You see this option if you select **Service tag** for **Source type**.
Service type	List of predefined services available for the test. **Notes**: - If you select a predefined service, the service port number and protocol are automatically selected. - If you don't see the port and protocol information that you want, select **Custom**, and then enter the port number and select the protocol that you want.
Port	VM port number. **Note**: If you select one of the predefined services, the correct port number is automatically selected. Manually enter the port number when you select **Custom** for **Service type**.
Protocol	Connection protocol. Available options are: **Any**, **TCP**, and **UDP**. **Note**: If you select one of the predefined services, the correct protocol used by the service is automatically selected. Manually select the protocol when you select **Custom** for **Service type**.

6. To test if RDP connection is allowed to the VM from a remote IP address, select the following values:

Expand table

Setting	Value
Source type	Select **My IP address**.

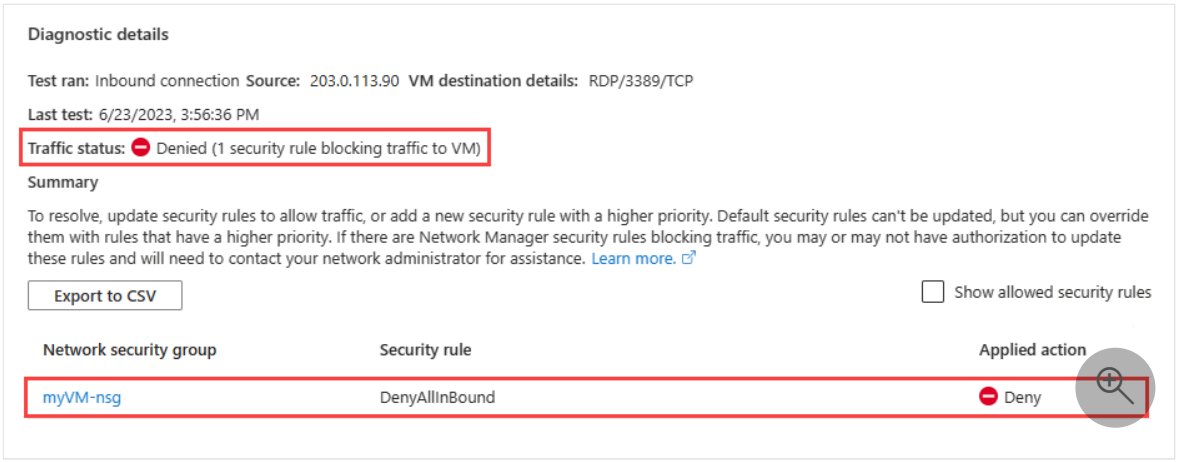
Setting	Value
Service type	Select **RDP**.
Port	Leave the default of **3389**.
Protocol	Leave the default of **TCP**.



7. Select **Run test**.

   After a few seconds, you see the details of the test:

   - If RDP connections are allowed to the VM from the remote IP address, you see **Traffic status: Allowed**.
   - If RDP connections are blocked, you see **Traffic status: Denied**. In the Summary section, you see the security rules that are blocking the traffic.

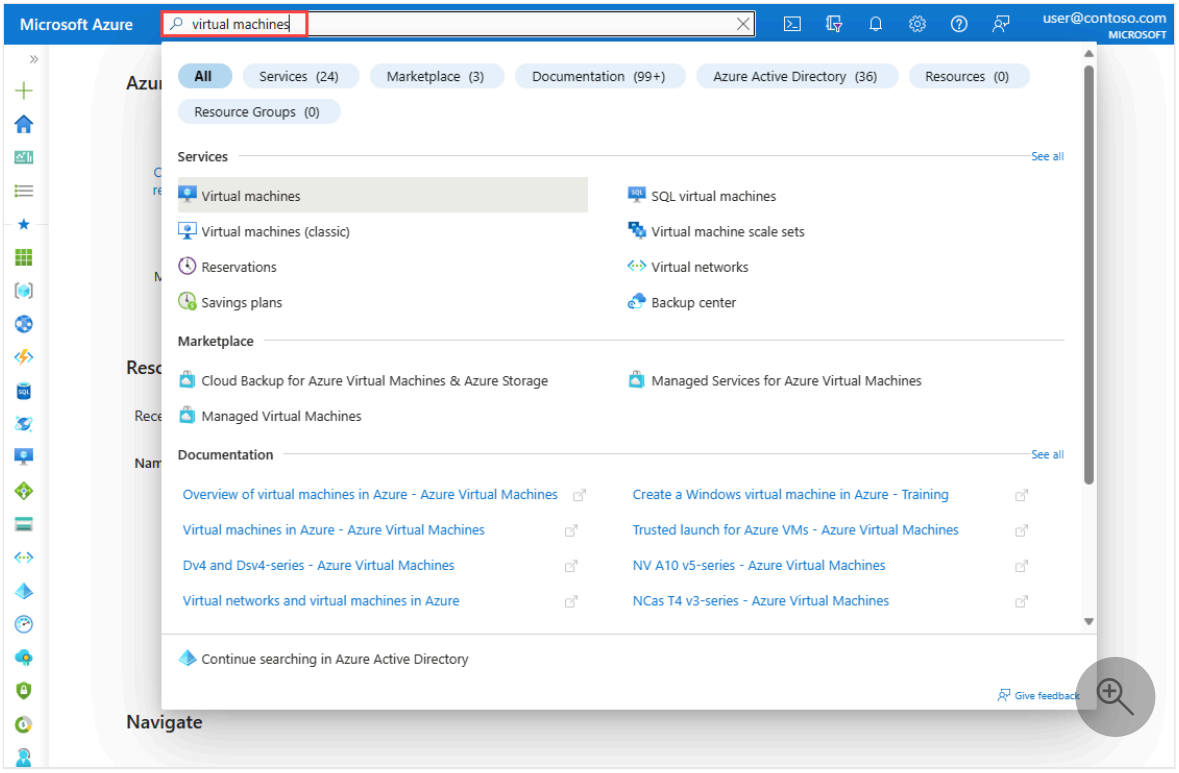


To allow the RDP connection to the VM from the remote IP address, add to the network security group a security rule that allows RDP connections from the remote IP address. This security rule must have a higher priority than the one that's blocking the traffic. For more information, see Create, change, or delete a network security group.

Test outbound connections

In this section, you test your VM can have connect to the internet.

1. In the search box at the top of the portal, enter *virtual machines*. Select **Virtual machines** from the search results.

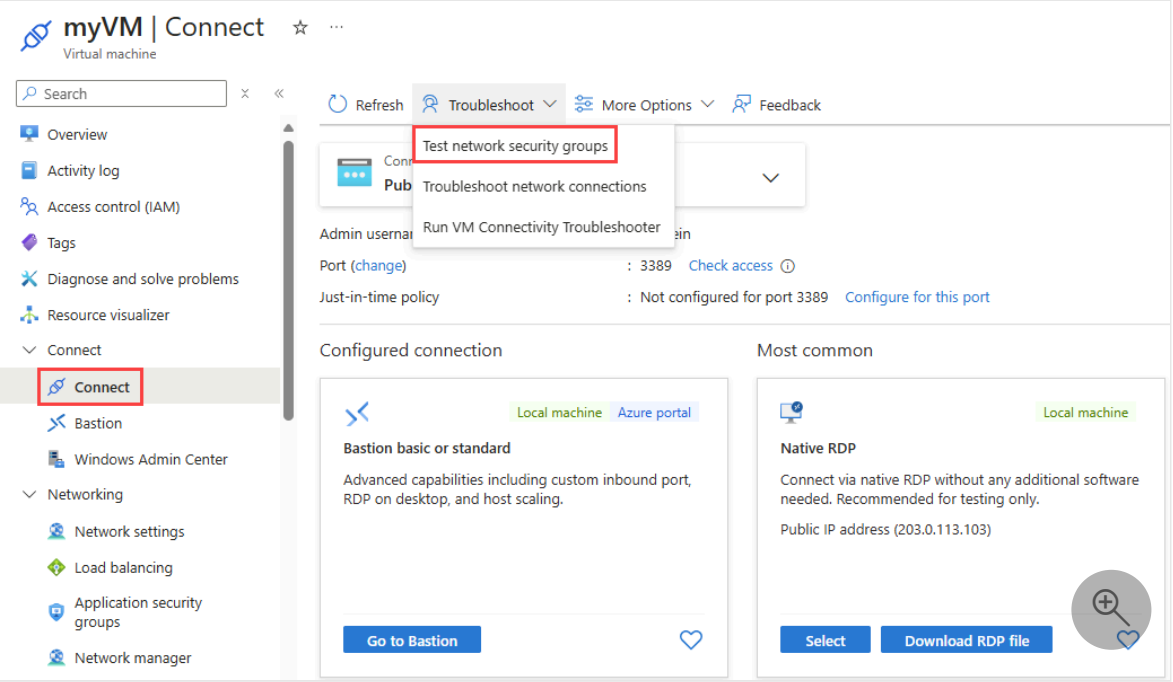


2. Select the VM you want to test.

3. Under **Connect**, select **Connect**.

> **Note**
>
> The virtual machine must be in running state.

4. Under **Troubleshoot**, select **Test network security groups**.



5. Select **Outbound connections**. The following options are available for **Outbound** tests:

Expand table

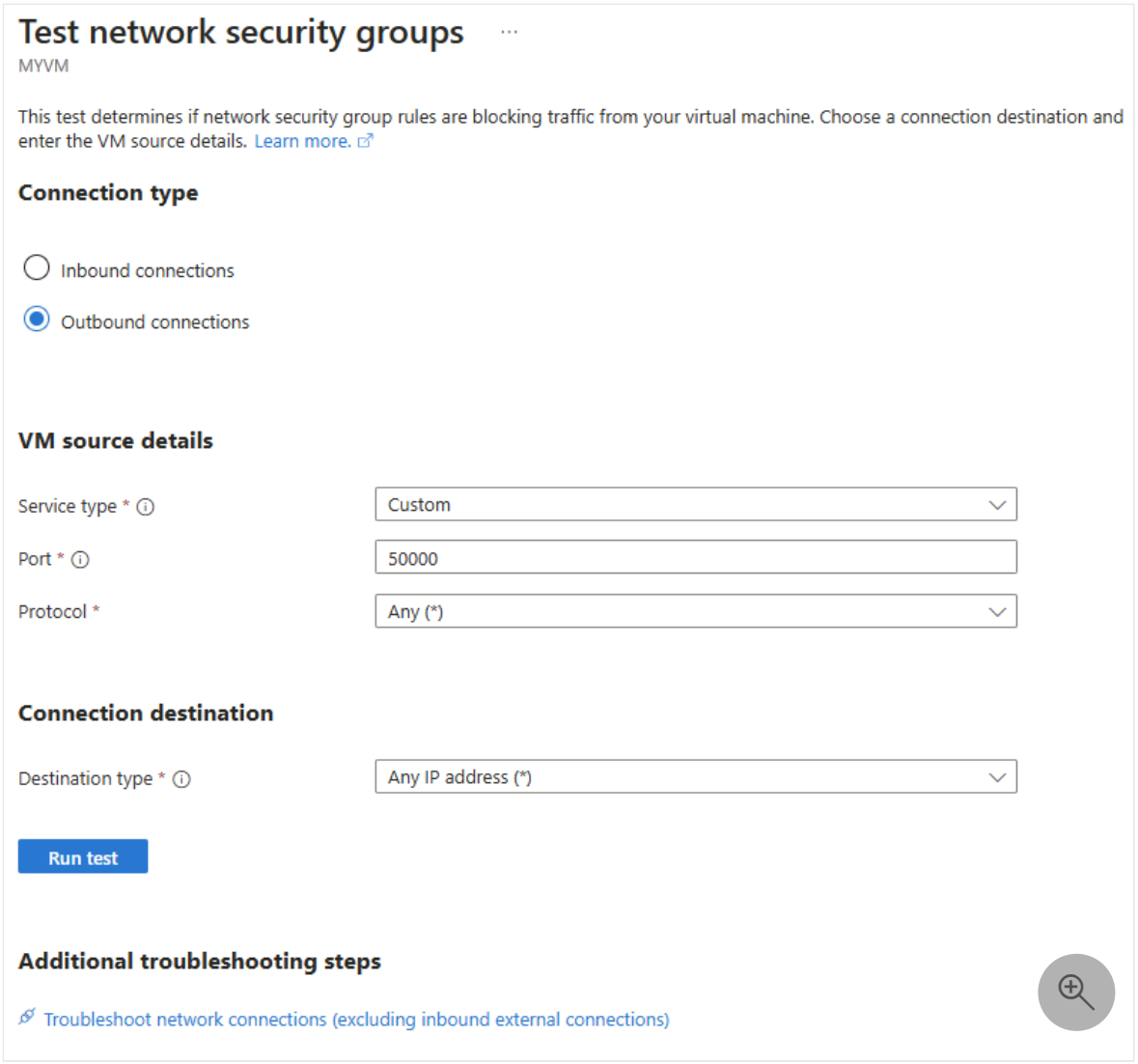
Setting	Value
Service type	List of predefined services available for the test. **Notes**: - If you select a predefined service, the service port number and protocol are automatically selected. - If you don't see the port and protocol information that you want, select **Custom**, and then enter the port number and select the protocol that you want.
Port	VM port number. **Note**: If you select one of the predefined services, the correct port number is automatically selected. Manually enter the port number when you select **Custom** for **Service type**.
Protocol	Connection protocol. Available options are: **Any**, **TCP**, and **UDP**.

Setting	Value
	Note: If you select one of the predefined services, the correct protocol used by the service is automatically selected. Manually select the protocol when you select **Custom** for **Service type**.
Destination type	- **My IP address**: your public IP address that you're using to access the Azure portal. - **Any IP address**: any source IP address. - **Other IP address/CIDR**: Source IP address or address prefix. - **Service tag**: Source service tag.
IP address/CIDR	The IP address or address prefix that you want to use as the destination. **Note**: You see this option if you select **Other IP address/CIDR** for **Source type**.
Service tag	The service tag that you want to use as the destination. **Note**: You see this option if you select **Service tag** for **Source type**.

6. To test if the VM can connect to the internet, select the following values:

Expand table

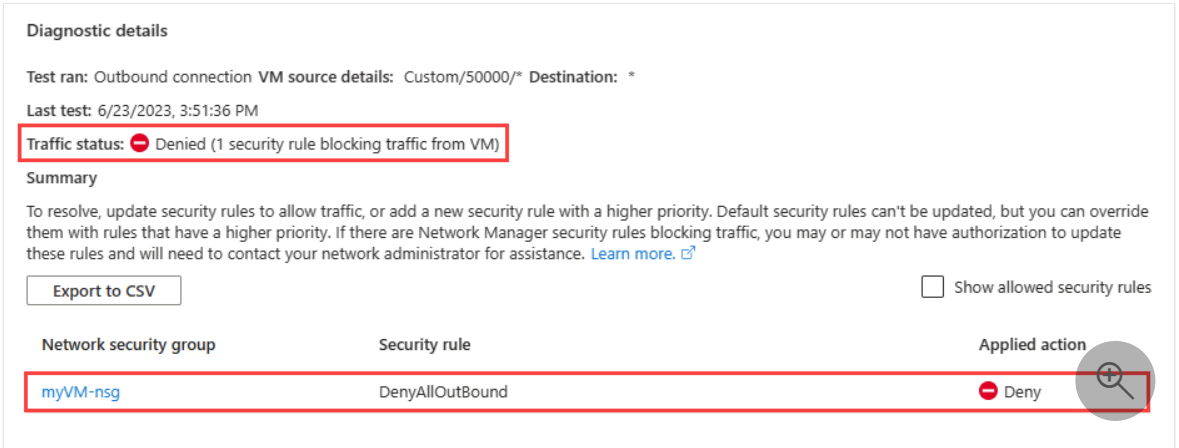
Setting	Value
Service type	Select **Custom**.
Port	Leave the default of **50000**.
Protocol	Leave the default of **Any**.
Destination type	Select **Any IP address**.



7. Select **Run test**.

   After a few seconds, you see the details of the test:

   - If connections to the internet are allowed from the VM, you see **Traffic status: Allowed**.
   - If connections to the internet are blocked, you see **Traffic status: Denied**. In the Summary section, you see the security rules that are blocking the traffic.



To allow internet connections from the VM, add to the network security group a security rule that allows connections to the internet service tag. This security rule must have a higher priority than the one that's blocking the traffic. For more information, see Create, change, or delete a network security group.

Related content

- To learn how to troubleshoot VM connections, see Troubleshoot connections with Azure Network Watcher.
- To learn more about network security groups, see Network security groups overview.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI

Summarize this article for me

In this article, you create a virtual machine (VM) with a static public IP address. A public IP address enables you to communicate to a VM from the internet. Assign a static public IP address, rather than a dynamic address, to ensure the address never changes.

Public IP addresses have a nominal charge. There's a limit to the number of public IP addresses that you can use per subscription.

You can download the list of ranges (prefixes) for the Azure Public, US government, China, and Germany clouds.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a virtual machine with a static public IP address

In this section, you create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI. Along with the virtual machine, you create a public IP address and the other required resources.

Azure portal

Sign in to Azure

Sign in to the Azure portal.

Create a resource group

1. In the search box at the top of the portal, enter *Resource group*.
2. In the search results, select **Resource groups**.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter or select the following:

Expand table

Setting	Value
Subscription	Select your Azure subscription
Resource Group	Enter *myResourceGroup*.
Region	Select **East US**.

5. Select **Review + create**, then select **Create**.

Create a virtual machine

1. In the search box at the top of the portal, enter *Virtual machine*.
2. In the search results, select **Virtual machines**.
3. Select **+ Create**, then select **Azure virtual machine**.
4. In **Basics** tab of **Create a virtual machine**, enter or select the following:

Expand table

Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select **myResourceGroup**.
Instance details	
Virtual machine name	Enter *myVM*.
Region	Select **East US**.
Availability Options	Select **No infrastructure redundancy required**.
Security type	Select **Standard**.

Setting	Value
Image	Select **Windows Server 2019 Datacenter - x64 Gen2**.
Size	Choose VM size or take default setting.
Administrator account	
Username	Enter a username.
Password	Enter a password.
Confirm password	Reenter password.
Public inbound ports	Select **None**.

> **Note**
>
> All public inbound ports are closed for this virtual machine. To manage your virtual machines, deploy Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion from the Azure portal.

5. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

6. In the **Networking** tab, enter or select the following:

Expand table

Setting	Value
Network interface	
Virtual network	Accept the default network name.
Subnet	Accept the default subnet configuration.
Public IP	Select **Create new**. In **Create public IP address**, enter *myPublicIP* in **Name**. **SKU**: select **Standard**. **Assignment**: select **Static**. Select **OK**.

> **Note**

The SKU of the virtual machine's public IP address must match the public IP SKU of Azure public load balancer when added to the backend pool of the load balancer. For details, see Azure Load Balancer.

7. Select **Review + create**.

8. Review the settings, and then select **Create**.

> ⚠ **Warning**
>
> Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary. For more information, see Add a private IP address to an operating system.

> ⓘ **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Clean up resources

When resources are no longer needed, delete all resources created in this article to avoid incurring charges.

Azure portal

Use the Azure portal to delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the search box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this article, you learned how to create a VM with a static public IP.

- Learn how to Configure IP addresses for an Azure network interface.
- Learn how to Assign multiple IP addresses to virtual machines using the Azure portal.
- Learn more about public IP addresses in Azure.

Last updated on 02/25/2026

Create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI

Summarize this article for me

In this article, you create a virtual machine (VM) with a static public IP address. A public IP address enables you to communicate to a VM from the internet. Assign a static public IP address, rather than a dynamic address, to ensure the address never changes.

Public IP addresses have a nominal charge. There's a limit to the number of public IP addresses that you can use per subscription.

You can download the list of ranges (prefixes) for the Azure Public, US government, China, and Germany clouds.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a virtual machine with a static public IP address

In this section, you create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI. Along with the virtual machine, you create a public IP address and the other required resources.

Azure portal

Sign in to Azure

Sign in to the Azure portal.

Create a resource group

1. In the search box at the top of the portal, enter *Resource group*.
2. In the search results, select **Resource groups**.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter or select the following:

Expand table

Setting	Value
Subscription	Select your Azure subscription
Resource Group	Enter *myResourceGroup*.
Region	Select **East US**.

5. Select **Review + create**, then select **Create**.

Create a virtual machine

1. In the search box at the top of the portal, enter *Virtual machine*.
2. In the search results, select **Virtual machines**.
3. Select **+ Create**, then select **Azure virtual machine**.
4. In **Basics** tab of **Create a virtual machine**, enter or select the following:

Expand table

Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select **myResourceGroup**.
Instance details	
Virtual machine name	Enter *myVM*.
Region	Select **East US**.
Availability Options	Select **No infrastructure redundancy required**.
Security type	Select **Standard**.

Setting	Value
Image	Select **Windows Server 2019 Datacenter - x64 Gen2**.
Size	Choose VM size or take default setting.
Administrator account	
Username	Enter a username.
Password	Enter a password.
Confirm password	Reenter password.
Public inbound ports	Select **None**.

> **Note**
>
> All public inbound ports are closed for this virtual machine. To manage your virtual machines, deploy Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion from the Azure portal.

5. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

6. In the **Networking** tab, enter or select the following:

Expand table

Setting	Value
Network interface	
Virtual network	Accept the default network name.
Subnet	Accept the default subnet configuration.
Public IP	Select **Create new**. In **Create public IP address**, enter *myPublicIP* in **Name**. **SKU**: select **Standard**. **Assignment**: select **Static**. Select **OK**.

> **Note**

The SKU of the virtual machine's public IP address must match the public IP SKU of Azure public load balancer when added to the backend pool of the load balancer. For details, see Azure Load Balancer.

7. Select **Review + create**.

8. Review the settings, and then select **Create**.

> ⚠ **Warning**
>
> Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary. For more information, see Add a private IP address to an operating system.

> ⓘ **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Clean up resources

When resources are no longer needed, delete all resources created in this article to avoid incurring charges.

Azure portal

Use the Azure portal to delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the search box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this article, you learned how to create a VM with a static public IP.

- Learn how to Configure IP addresses for an Azure network interface.
- Learn how to Assign multiple IP addresses to virtual machines using the Azure portal.
- Learn more about public IP addresses in Azure.

Last updated on 02/25/2026

Create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI

Summarize this article for me

In this article, you create a virtual machine (VM) with a static public IP address. A public IP address enables you to communicate to a VM from the internet. Assign a static public IP address, rather than a dynamic address, to ensure the address never changes.

Public IP addresses have a nominal charge. There's a limit to the number of public IP addresses that you can use per subscription.

You can download the list of ranges (prefixes) for the Azure Public, US government, China, and Germany clouds.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a virtual machine with a static public IP address

In this section, you create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI. Along with the virtual machine, you create a public IP address and the other required resources.

Azure portal

Sign in to Azure

Sign in to the Azure portal.

Create a resource group

1. In the search box at the top of the portal, enter *Resource group*.
2. In the search results, select **Resource groups**.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter or select the following:

Expand table

Setting	Value
Subscription	Select your Azure subscription
Resource Group	Enter *myResourceGroup*.
Region	Select **East US**.

5. Select **Review + create**, then select **Create**.

Create a virtual machine

1. In the search box at the top of the portal, enter *Virtual machine*.
2. In the search results, select **Virtual machines**.
3. Select **+ Create**, then select **Azure virtual machine**.
4. In **Basics** tab of **Create a virtual machine**, enter or select the following:

Expand table

Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select **myResourceGroup**.
Instance details	
Virtual machine name	Enter *myVM*.
Region	Select **East US**.
Availability Options	Select **No infrastructure redundancy required**.
Security type	Select **Standard**.

Setting	Value
Image	Select **Windows Server 2019 Datacenter - x64 Gen2**.
Size	Choose VM size or take default setting.
Administrator account	
Username	Enter a username.
Password	Enter a password.
Confirm password	Reenter password.
Public inbound ports	Select **None**.

> **Note**
>
> All public inbound ports are closed for this virtual machine. To manage your virtual machines, deploy Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion from the Azure portal.

5. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

6. In the **Networking** tab, enter or select the following:

Expand table

Setting	Value
Network interface	
Virtual network	Accept the default network name.
Subnet	Accept the default subnet configuration.
Public IP	Select **Create new**. In **Create public IP address**, enter *myPublicIP* in **Name**. **SKU**: select **Standard**. **Assignment**: select **Static**. Select **OK**.

> **Note**

The SKU of the virtual machine's public IP address must match the public IP SKU of Azure public load balancer when added to the backend pool of the load balancer. For details, see Azure Load Balancer.

7. Select **Review + create**.

8. Review the settings, and then select **Create**.

> ⚠ **Warning**
>
> Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary. For more information, see Add a private IP address to an operating system.

> ⓘ **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Clean up resources

When resources are no longer needed, delete all resources created in this article to avoid incurring charges.

Azure portal

Use the Azure portal to delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the search box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this article, you learned how to create a VM with a static public IP.

- Learn how to Configure IP addresses for an Azure network interface.
- Learn how to Assign multiple IP addresses to virtual machines using the Azure portal.
- Learn more about public IP addresses in Azure.

Last updated on 02/25/2026

Associate a public IP address to a virtual machine

In this article, you learn how to associate a public IP address to an existing virtual machine (VM). To do so, you associate the public IP address to an IP configuration of a network interface attached to a VM. You can use the Azure portal, the Azure CLI, or Azure PowerShell by selecting the tab for the method you want to use.

If you want to instead create a new VM with a public IP address, you can use the Azure portal, the Azure CLI, or Azure PowerShell.

Public IP addresses have a nominal fee. For details, see pricing. There's a limit to the number of public IP addresses that you can use per subscription. For details, see limits.

> **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Prerequisites

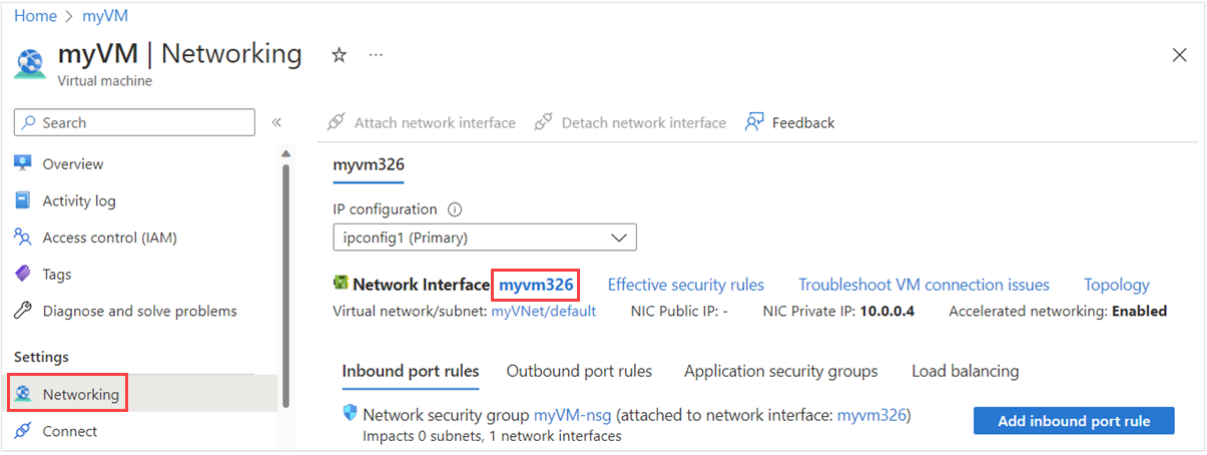
- An Azure account with an active subscription. You can create an account for free.

Azure portal

1. Sign in to the Azure portal.

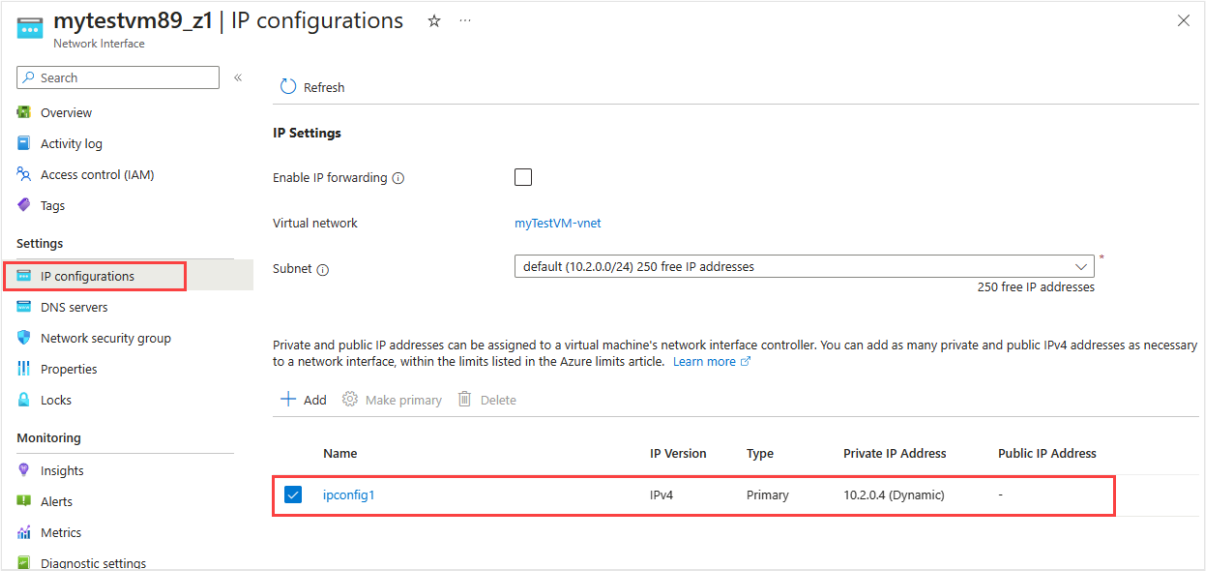
2. In the portal, search for and select the VM that you want to add the public IP address to.

3. Under **Settings** in the left pane, select **Networking**, and then select the network interface you want to add the public IP address to.



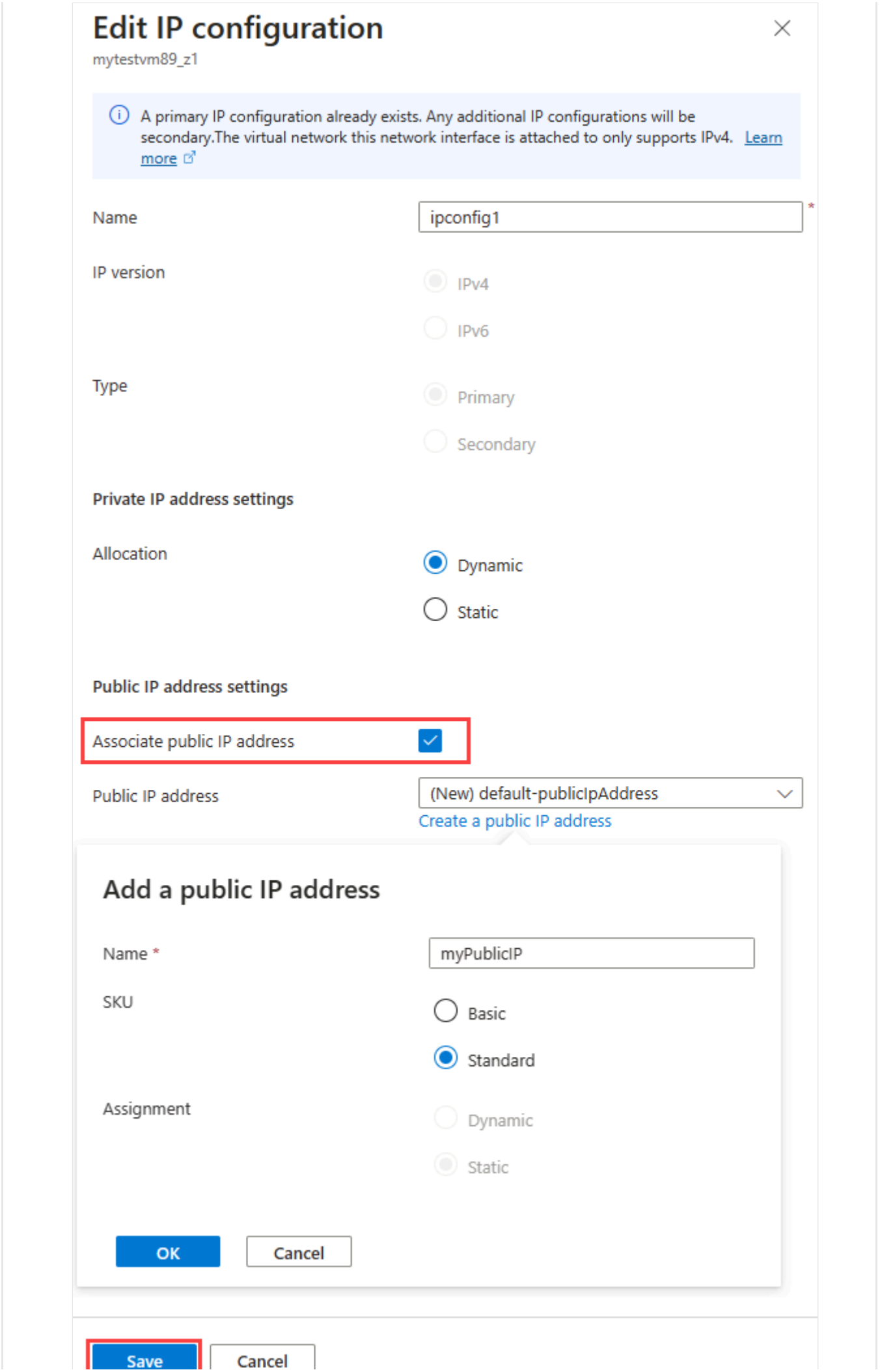
> **Note**
>
> Public IP addresses are associated to the network interfaces that are attached to a VM. In this screenshot, the VM has only one network interface. If the VM had multiple network interfaces, they would all appear, and you'd select the network interface you want to associate the public IP address to.

4. From the **Network interface** window, under **Settings**, select **IP configurations***, and then select an IP configuration from the list.



> **Note**
>
> Public IP addresses are associated to the IP configurations for a network interface. In this screenshot, the network interface has only one IP configuration. If the network interface had multiple IP configurations, they would all appear in the list, and you'd select the IP configuration that you want to associate the public IP address to.

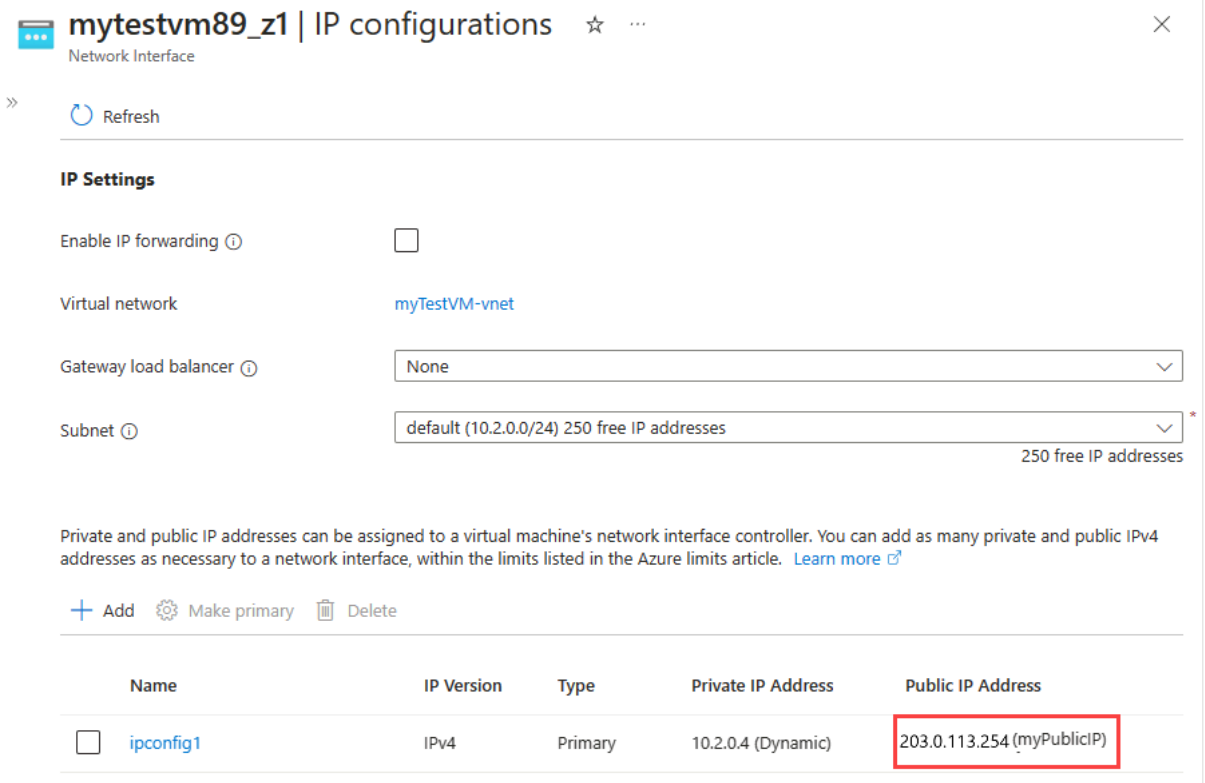
5. In the **Edit IP configuration** window, select **Associate public IP address**, then select **Public IP address** to choose an existing public IP address from the drop-down list. If no public IP addresses are listed, you need to create one. To learn how, see Create a public IP address.


Edit IP configuration
mytestvm89_z1
A primary IP configuration already exists. Any additional IP configurations will be secondary.The virtual network this network interface is attached to only supports IPv4. Learn more
Name
ipconfig1
IP version
IPv4
IPv6
Type
Primary
Secondary
Private IP address settings
Allocation
Dynamic
Static
Public IP address settings
Associate public IP address
Public IP address
(New) default-publicIpAddress
Create a public IP address
Add a public IP address
Name *
myPublicIP
SKU
Basic
Standard
Assignment
Dynamic
Static
OK
Cancel
Save
Cancel

> **Note**
>
> The public IP addresses that appear in the drop-down list exist in the same region as the VM. If you have multiple public IP addresses created in the region, all appear here. Any address already associated to a different resource is grayed out.

6. Select **Save**.

7. In the **IP Configurations** window, view the public IP address assigned to the IP configuration. It might take a few seconds for a newly associated IP address to appear.



> **Note**
>
> An IP address is assigned from the pool of public IP addresses reserved for an Azure region. For a list of the address pools used in each region, see Azure IP ranges and service tags. If you need the address to be assigned from a specific prefix, use a Public IP address prefix.

8. Open the necessary ports in your security groups by adjusting the security rules in the network security groups. For information, see Allow network traffic to the VM.

> **Note**
>
> To share a VM with an external user, you must add a public IP address to the VM. Alternatively, external users can connect to VM's private IP address through Azure Bastion.

Allow network traffic to the VM

Before you can connect to a public IP address from the internet, you must open the necessary ports/protocols in your network security groups. These ports must be open in any network security group that you might have associated to the network interface, the subnet of the network interface, or both. Although network security groups filter traffic to the private IP address of the network interface, after inbound internet traffic arrives at the public IP address, Azure translates the public address to the private IP address. Therefore, if a network security group prevents the traffic flow, the communication with the public IP address fails.

You can view the effective security rules for a network interface and its subnet for the Azure portal, the Azure CLI, or Azure PowerShell.

Limits

- If a new public IP address is added to the VM and traffic needs to switch over to the new IP address, the VM needs to be restarted as this will reset all existing flows.

Next steps

In this article, you learned how to associate a public IP address to a VM using the Azure portal, Azure CLI, or Azure PowerShell.

- Use a network security group to allow inbound internet traffic to your VM.
-
- Learn how to create a network security group.

Last updated on 11/20/2025

Dissociate a public IP address from an Azure VM

Article • 11/25/2024

In this article, you learn how to dissociate a public IP address from an Azure virtual machine (VM). Removing the public IP address of your VM removes access to the Internet.

You can use the Azure portal, the Azure CLI, or Azure PowerShell to dissociate a public IP address from a VM.

Prerequisites

Azure portal

- An Azure account with an active subscription. Create an account for free.
- A virtual machine with a public IP address associated to it.

Dissociate a public IP address from a VM

In this step, you dissociate a public IP address from a virtual machine using the Azure portal, Azure CLI, or Azure PowerShell. The IP address is associated to an IP configuration of a network interface attached to the VM.

Azure portal

1. Sign in to the Azure portal.
2. Browse to, or search for the virtual machine that you want to disassociate the public IP address from and then select it.
3. In the VM page, select **Overview**, and then select the public IP address.
4. In the public IP address page, select **Overview**, and then select **Dissociate**.
5. In **Dissociate public IP address**, select **Yes**.

Next steps

In this article, you learned how to dissociate a public IP address from a virtual machine.

- Learn more about public IP addresses in Azure.
- Learn how to associate a public IP address to a VM.
- Learn how to configure IP addresses for an Azure network interface.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a virtual machine with a static private IP address

Summarize this article for me

When you create a virtual machine (VM), it's automatically assigned a private IP address from a range that you specify. This IP address is based on the subnet in which the VM is deployed, and the VM keeps this address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want to assign a specific IP address in this subnet for your VM, use a static IP address.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a resource group and a virtual machine

Azure portal

Create a resource group

1. Sign in to the Azure portal.
2. In the portal, search for and select **Resource groups**.
3. Select + **Create**.
4. On the **Basics** tab, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Enter *myResourceGroup*

Setting	Value
Region	Select **(US) East US**

5. Select **Review + create**, and then select **Create**.

Create a virtual network

1. In the portal, search for and select **Virtual networks**.
2. Select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Select **myResourceGroup**
Virtual network name	Enter *myVNet*
Region	Select **(US) East US**

4. Select **Review + create**, and then select **Create**.

Create a virtual machine

1. In the portal, search for and select **Virtual machines**.
2. Select **Create** > **Azure virtual machine**.
3. On the **Basics** tab of the **Create a virtual machine** screen, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Select **myResourceGroup**
Virtual machine name	Enter *myVM*

Setting	Value
Region	Select **(US) East US 2**
Availability options	Select **No infrastructure redundancy required**
Security type	Select **Standard**
Image	Select **Ubuntu Server 22.04 LTS - x64 Gen2**
Size	Accept the default, or drop down and select a size
Authentication type	Select **SSH public key**
Username	Enter *azureuser*
SSH public key source	Select **Generate new key pair**
Key pair name	Enter *mySSHKey*
Public inbound ports	Select **None**

4. Select the **Networking** tab at the top of the page.

5. On the **Networking** page, enter or select the following values:

    - **Virtual network**: Select **myVNet**.
    - **Subnet**: Select **default** if not already selected.
    - **Public IP**: Select **None**.

6. Select **Review + create**. Review the settings, and then select **Create**.

> ⓘ **Note**
>
> The virtual machine is created without a public IP address and with no public inbound ports. To connect to the virtual machine, use Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion with default settings.

Change private IP address to static

Azure portal

In the following steps, you change the private IP address **static** for the VM created previously:

1. In the portal, search for and select **Virtual machines**.
2. In **Virtual machines**, select **myVM** from the list.
3. On the **myVM** page, select **Network settings** under **Networking**.
4. On the **Network settings** page, select the name of the network interface next to **Network interface**.
5. On the **Network interface** page, under **Settings**, select **IP configurations**.
6. In **IP configurations**, select **ipconfig1** in the list.
7. Under **Assignment**, select **Static**. Change the private **IP address** if you want a different one, and then select **Save**.

> ⚠ **Warning**
>
> If you change the private IP address, the VM associated with the network interface will be restarted to utilize the new IP address.
>
> From within the operating system of a VM, avoid associating a static *private* IP address on an Azure VM. Only assign a static private IP when it's necessary, such as when assigning many IP addresses to VMs.
>
> If you manually set the private IP address within the operating system, make sure it matches the private IP address assigned to the Azure network interface. Otherwise, you can lose connectivity to the VM. For more information, see private IP address settings.

Clean up resources

Azure portal

When all the resources are no longer need, delete the resource group and all of the resources it contains:

1. In the portal, search for and select **myResourceGroup**.
2. From the **myResourceGroup** screen, select **Delete resource group**.

3. Enter *myResourceGroup* for **Enter resource group name to confirm deletion**, and then select **Delete**.

Next steps

- Learn more about static public IP addresses in Azure.
- Learn more about public IP addresses in Azure.
- Learn more about Azure public IP address settings.
- Learn more about private IP addresses and assigning a static private IP address to an Azure VM.

Last updated on 02/25/2026

Create a virtual machine with a static private IP address

Summarize this article for me

When you create a virtual machine (VM), it's automatically assigned a private IP address from a range that you specify. This IP address is based on the subnet in which the VM is deployed, and the VM keeps this address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want to assign a specific IP address in this subnet for your VM, use a static IP address.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a resource group and a virtual machine

Azure portal

Create a resource group

1. Sign in to the Azure portal.
2. In the portal, search for and select **Resource groups**.
3. Select + **Create**.
4. On the **Basics** tab, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Enter *myResourceGroup*

Setting	Value
Region	Select **(US) East US**

5. Select **Review + create**, and then select **Create**.

Create a virtual network

1. In the portal, search for and select **Virtual networks**.
2. Select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Select **myResourceGroup**
Virtual network name	Enter *myVNet*
Region	Select **(US) East US**

4. Select **Review + create**, and then select **Create**.

Create a virtual machine

1. In the portal, search for and select **Virtual machines**.
2. Select **Create** > **Azure virtual machine**.
3. On the **Basics** tab of the **Create a virtual machine** screen, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Select **myResourceGroup**
Virtual machine name	Enter *myVM*

Setting	Value
Region	Select **(US) East US 2**
Availability options	Select **No infrastructure redundancy required**
Security type	Select **Standard**
Image	Select **Ubuntu Server 22.04 LTS - x64 Gen2**
Size	Accept the default, or drop down and select a size
Authentication type	Select **SSH public key**
Username	Enter *azureuser*
SSH public key source	Select **Generate new key pair**
Key pair name	Enter *mySSHKey*
Public inbound ports	Select **None**

4. Select the **Networking** tab at the top of the page.

5. On the **Networking** page, enter or select the following values:

    - **Virtual network**: Select **myVNet**.
    - **Subnet**: Select **default** if not already selected.
    - **Public IP**: Select **None**.

6. Select **Review + create**. Review the settings, and then select **Create**.

> ⓘ **Note**
>
> The virtual machine is created without a public IP address and with no public inbound ports. To connect to the virtual machine, use Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion with default settings.

Change private IP address to static

Azure portal

In the following steps, you change the private IP address **static** for the VM created previously:

1. In the portal, search for and select **Virtual machines**.
2. In **Virtual machines**, select **myVM** from the list.
3. On the **myVM** page, select **Network settings** under **Networking**.
4. On the **Network settings** page, select the name of the network interface next to **Network interface**.
5. On the **Network interface** page, under **Settings**, select **IP configurations**.
6. In **IP configurations**, select **ipconfig1** in the list.
7. Under **Assignment**, select **Static**. Change the private **IP address** if you want a different one, and then select **Save**.

> ⚠ **Warning**
>
> If you change the private IP address, the VM associated with the network interface will be restarted to utilize the new IP address.
>
> From within the operating system of a VM, avoid associating a static *private* IP address on an Azure VM. Only assign a static private IP when it's necessary, such as when assigning many IP addresses to VMs.
>
> If you manually set the private IP address within the operating system, make sure it matches the private IP address assigned to the Azure network interface. Otherwise, you can lose connectivity to the VM. For more information, see private IP address settings.

Clean up resources

Azure portal

When all the resources are no longer need, delete the resource group and all of the resources it contains:

1. In the portal, search for and select **myResourceGroup**.
2. From the **myResourceGroup** screen, select **Delete resource group**.

3. Enter *myResourceGroup* for **Enter resource group name to confirm deletion**, and then select **Delete**.

Next steps

- Learn more about static public IP addresses in Azure.
- Learn more about public IP addresses in Azure.
- Learn more about Azure public IP address settings.
- Learn more about private IP addresses and assigning a static private IP address to an Azure VM.

Last updated on 02/25/2026

Create a virtual machine with a static private IP address

Summarize this article for me

When you create a virtual machine (VM), it's automatically assigned a private IP address from a range that you specify. This IP address is based on the subnet in which the VM is deployed, and the VM keeps this address until the VM is deleted. Azure dynamically assigns the next available private IP address from the subnet you create a VM in. If you want to assign a specific IP address in this subnet for your VM, use a static IP address.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a resource group and a virtual machine

Azure portal

Create a resource group

1. Sign in to the Azure portal.
2. In the portal, search for and select **Resource groups**.
3. Select + **Create**.
4. On the **Basics** tab, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Enter *myResourceGroup*

Setting	Value
Region	Select **(US) East US**

5. Select **Review + create**, and then select **Create**.

Create a virtual network

1. In the portal, search for and select **Virtual networks**.
2. Select **+ Create**.
3. On the **Basics** tab of **Create virtual network**, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Select **myResourceGroup**
Virtual network name	Enter *myVNet*
Region	Select **(US) East US**

4. Select **Review + create**, and then select **Create**.

Create a virtual machine

1. In the portal, search for and select **Virtual machines**.
2. Select **Create** > **Azure virtual machine**.
3. On the **Basics** tab of the **Create a virtual machine** screen, enter or select the following values:

Expand table

Setting	Value
Subscription	Select your subscription
Resource group	Select **myResourceGroup**
Virtual machine name	Enter *myVM*

Setting	Value
Region	Select **(US) East US 2**
Availability options	Select **No infrastructure redundancy required**
Security type	Select **Standard**
Image	Select **Ubuntu Server 22.04 LTS - x64 Gen2**
Size	Accept the default, or drop down and select a size
Authentication type	Select **SSH public key**
Username	Enter *azureuser*
SSH public key source	Select **Generate new key pair**
Key pair name	Enter *mySSHKey*
Public inbound ports	Select **None**

4. Select the **Networking** tab at the top of the page.

5. On the **Networking** page, enter or select the following values:

    - **Virtual network**: Select **myVNet**.
    - **Subnet**: Select **default** if not already selected.
    - **Public IP**: Select **None**.

6. Select **Review + create**. Review the settings, and then select **Create**.

> ⓘ **Note**
>
> The virtual machine is created without a public IP address and with no public inbound ports. To connect to the virtual machine, use Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion with default settings.

Change private IP address to static

Azure portal

In the following steps, you change the private IP address **static** for the VM created previously:

1. In the portal, search for and select **Virtual machines**.
2. In **Virtual machines**, select **myVM** from the list.
3. On the **myVM** page, select **Network settings** under **Networking**.
4. On the **Network settings** page, select the name of the network interface next to **Network interface**.
5. On the **Network interface** page, under **Settings**, select **IP configurations**.
6. In **IP configurations**, select **ipconfig1** in the list.
7. Under **Assignment**, select **Static**. Change the private **IP address** if you want a different one, and then select **Save**.

> ⚠ **Warning**
>
> If you change the private IP address, the VM associated with the network interface will be restarted to utilize the new IP address.
>
> From within the operating system of a VM, avoid associating a static *private* IP address on an Azure VM. Only assign a static private IP when it's necessary, such as when assigning many IP addresses to VMs.
>
> If you manually set the private IP address within the operating system, make sure it matches the private IP address assigned to the Azure network interface. Otherwise, you can lose connectivity to the VM. For more information, see private IP address settings.

Clean up resources

Azure portal

When all the resources are no longer need, delete the resource group and all of the resources it contains:

1. In the portal, search for and select **myResourceGroup**.
2. From the **myResourceGroup** screen, select **Delete resource group**.

3. Enter *myResourceGroup* for **Enter resource group name to confirm deletion**, and then select **Delete**.

Next steps

- Learn more about static public IP addresses in Azure.
- Learn more about public IP addresses in Azure.
- Learn more about Azure public IP address settings.
- Learn more about private IP addresses and assigning a static private IP address to an Azure VM.

Last updated on 02/25/2026

Assign multiple IP addresses to virtual machines using the Azure portal

Summarize this article for me

An Azure Virtual Machine (VM) has one or more network interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it.

Assigning multiple IP addresses to a VM enables the following capabilities:

- Hosting multiple websites or services with different IP addresses and TLS/SSL certificates on a single server.
- Serve as a network virtual appliance, such as a firewall or load balancer.
- The ability to add any (primary or secondary) private IP addresses of the NICs to an Azure Load Balancer backend pool. For more information about load balancing multiple IP configurations, see Load balancing multiple IP configurations and Outbound rules.

Every NIC attached to a VM has one or more IP configurations associated to it. Each configuration is assigned one static or dynamic private IP address. Each configuration may also have one public IP address resource associated to it. To learn more about IP addresses in Azure, see IP addresses in Azure.

> **Note**
>
> All IP configurations on a single NIC must be associated to the same subnet. If multiple IPs on different subnets are desired, multiple NICs on a VM can be used. To learn more about multiple NICs on a VM in Azure, see Create VM with Multiple NICs.

There's a limit to how many private IP addresses can be assigned to a NIC. There's also a limit to how many public IP addresses that can be used in an Azure subscription. See the Azure limits article for details.

This article explains how to add multiple IP addresses to a virtual machine using the Azure portal.

> **Note**
>
> If you want to create a virtual machine with multiple IP addresses, or a static private IP address, you must create it using PowerShell or the Azure CLI.

Prerequisites

- An Azure account with an active subscription. Create an account for free.
- An existing Azure virtual machine with a Public IP. For more information about creating a virtual machine, see Create a Windows VM or Create a Linux VM.
  - The example used in this article is named **myVM**. Replace this value with your virtual machine name.
  - Windows Server virtual machines require port **3389** to be open for Remote Desktop Protocol (RDP) connections. Linux virtual machines require port **22** to be open for Secure Shell (SSH) connections.
- An Azure Bastion instance deployed for VM management in the same subnet as your virtual machine. For more information, see Deploy Azure Bastion automatically

> **Note**
>
> Though the steps in this article assign all IP configurations to a single NIC, you can also assign multiple IP configurations to any NIC in a multi-NIC VM. To learn how to create a VM with multiple NICs, see Create a VM with multiple NICs.

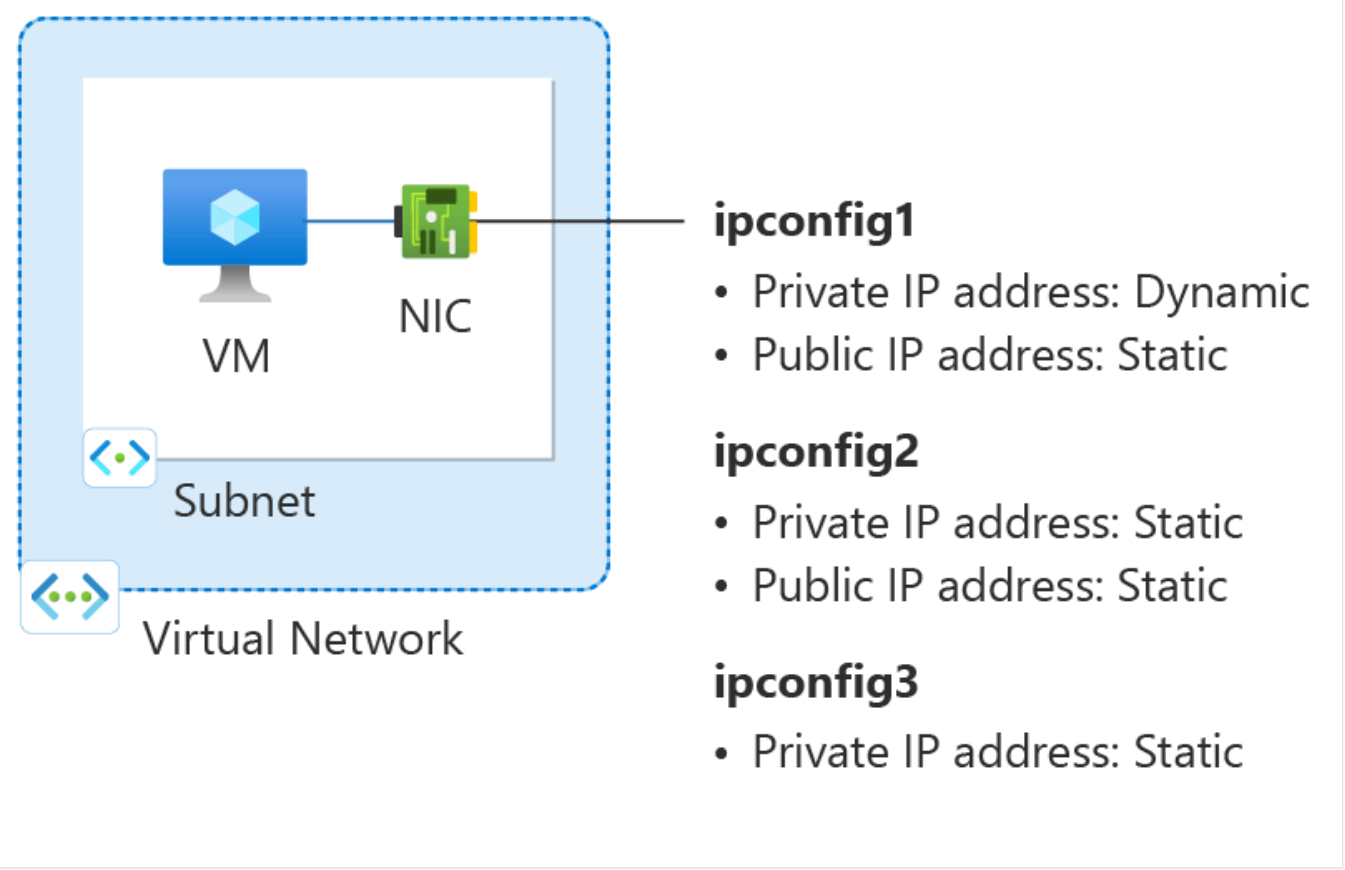


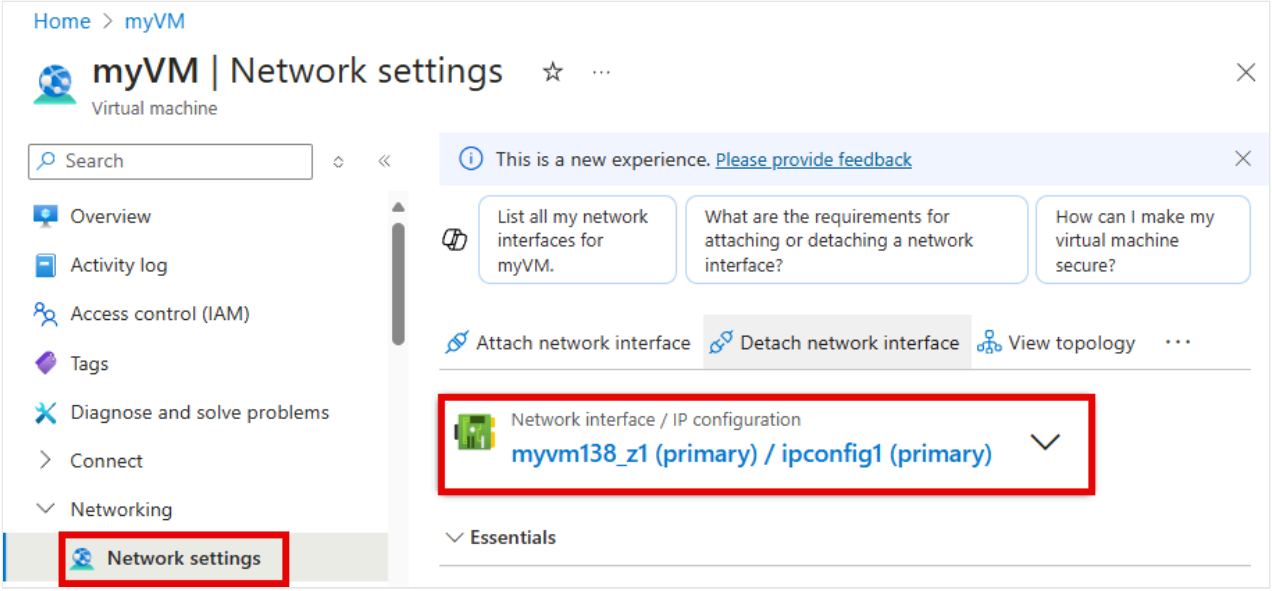
Figure: Diagram of network configuration resources created in this How-to article.

> **ⓘ Important**
>
> Avoid breaking connectivity when adding IP addresses
> When adding additional IP configurations to a VM, you **do not need to convert the existing (primary) private IP address from Dynamic to Static**.
>
> Changing the primary IP allocation type can disrupt RDP/SSH connectivity and may require recovery steps such as adding a temporary IP address. This is unnecessary when your goal is only to add secondary IP addresses.
>
> Recommended approach:
>
> - Leave the primary private IP configuration unchanged.
> - Add new **secondary IP configurations** (static or dynamic) directly to the NIC.
> - Only change the primary IP to Static if a fixed primary address is explicitly required and you understand the impact.

Add public and private IP address to a VM

You can add a private and public IP address to an Azure network interface by completing the following steps.

1. Sign in to the Azure portal.
2. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
3. In **Virtual machines**, select **myVM** or the name of your virtual machine.
4. Select **Network settings** under **Networking**.
5. In **Network settings**, Select the name of the network interface and IP configuration of the virtual machine in the **Network interface / IP configuration** dropdown menu. The existing IP configuration is displayed. In this example, it's named **myvm138_z1 (primary) / ipconfig1 (primary)**.



6. In **IP configurations**, select **+ Add** to create a new secondary IP configuration.

7. In **Add IP configuration**, enter or select the following information.

Expand table

Setting	Value
Name	Enter **ipconfig2**.
Private IP address settings	
Allocation	Select **Static**.
Private IP address	Enter an unused address in the network for your virtual machine. For the 10.1.0.0/24 subnet in the example, an IP would be **10.1.0.5**.
Public IP address	Select **Associate public IP address**
Public IP address	Select **Create a public IP address**. Enter **public-ip-02** in **Name**. Select **Standard** in **SKU**. Select **OK**.

8. Select **OK**.

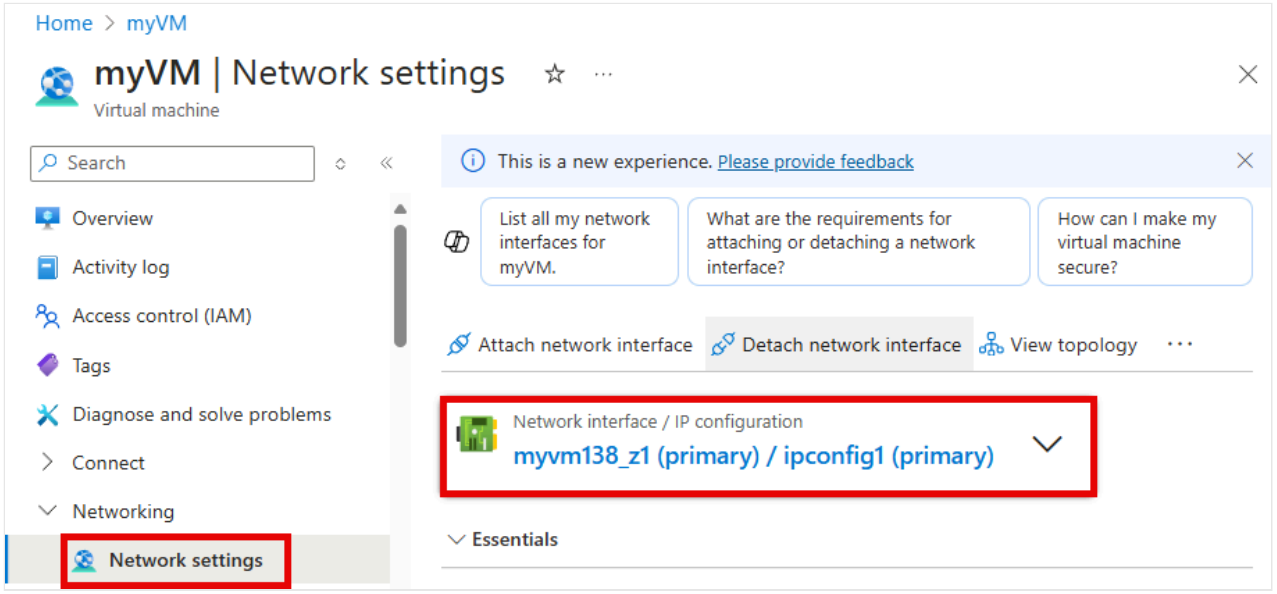
> **Note**
>
> When adding a static IP address, specify an unused and valid address in the subnet. This applies only to the **new secondary IP configuration**. The primary IP configuration does not need to be changed.

> **Important**
>
> Restarting the VM is required for the operating system to recognize newly added IP addresses.
> Adding secondary IP configurations does **not** require changing the primary IP allocation and does not inherently break existing connectivity.

Add private IP address to a VM

You can add a private IP address to a virtual machine by completing the following steps.

1. Sign in to the Azure portal.
2. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
3. In **Virtual machines**, select **myVM** or the name of your virtual machine.
4. Select **Networking** in **Settings**.
5. Select the name of the network interface of the virtual machine. In this example, it's named **myvm889_z1**.



6. In the network interface, select **IP configurations** in **Settings**.
7. The existing IP configuration is displayed. This configuration is created when the virtual machine is created. To add a private and public IP address to the virtual machine, select **+ Add**.
8. In **Add IP configuration**, enter or select the following information.

Expand table

Setting	Value
Name	Enter **ipconfig3**.
Private IP address settings	
Allocation	Select **Static**.
IP address	Enter an unused address in the network for your virtual machine. For the 10.1.0.0/24 subnet in the example, an IP would be **10.1.0.6**.

9. Select **OK**.

> **Note**
>
> When adding a static IP address, specify an unused and valid address in the subnet. This applies only to the **new secondary IP configuration**. The primary IP configuration does not need to be changed.

> **Important**
>
> Restarting the VM is required for the operating system to recognize newly added IP addresses.
> Adding secondary IP configurations does **not** require changing the primary IP allocation and does not inherently break existing connectivity.

Add IP addresses to a VM operating system

Connect and sign in to a VM you created with multiple private IP addresses. You must manually add all the private IP addresses, including the primary, that you added to the VM. Complete the following steps for your VM operating system.

Windows Server

▼ Expand

1. Open a command prompt or PowerShell.
2. Enter `ipconfig /all` at the command line. You see the **Primary** private IP address that was assigned through DHCP.

3. Enter `ncpa.cpl` at the command line to open the **Network Connections** configuration.
4. Open the **Properties** for the network adapter assigned the new IP addresses.
5. Double-click **Internet Protocol Version 4 (TCP/IPv4)**.
6. Select **Use the following IP address:**. Enter the following values.

Expand table

Setting	Value
IP address:	Enter the **Primary** private IP address.
Subnet mask:	Enter a subnet mask based on your IP address. For example, if the subnet is a **/24** subnet then the subnet mask is **255.255.255.0**.
Default gateway:	The first IP address in the subnet. If your subnet is **10.0.0.0/24**, then the gateway IP address is **10.0.0.1**.

7. Select **Use the following DNS server addresses:**. Enter the following values.

Expand table

Setting	Value
Preferred DNS server:	Enter your primary DNS server. Enter the IP address of **168.63.129.16** to use the default Azure provided DNS.

8. Select the **Advanced** button.
9. Select **Add**.
10. Enter the private **IP address** you added to the Azure network interface. Enter the corresponding **Subnet mask**. Select **Add**.
11. Repeat the previous steps to add any more private IP addresses that you added to the Azure network interface.

> **Important**
>
> You should never manually assign the public IP address assigned to an Azure virtual machine within the virtual machine's operating system. When you manually set the IP address within the operating system, ensure that it's the same address as the private IP

address assigned to the Azure network interface. Failure to assign the address correctly can cause loss of connectivity to the virtual machine. For more information, see Change IP address settings.

For more information about private IP addresses, see Private IP address.

12. Select **OK** to close the secondary IP address settings.
13. Select **OK** to close the adapter settings. Your RDP connection re-establishes.
14. Open a command prompt or PowerShell.
15. Enter `ipconfig /all` at the command line.
16. Verify the primary and secondary private IP addresses are present in the configuration.

PowerShell

```
PS C:\Users\azureuser> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : myVM
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Hyper-V Network Adapter
   Physical Address. . . . . . . . . : 00-0D-3A-E6-CE-A3
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::a8d1:11d5:3ab2:6a51%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.1.0.4(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.1.0.5(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.1.0.6(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.0.1
   DHCPv6 IAID . . . . . . . . . . . : 100666682
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-A8-26-B1-00-0D-3A-E6-CE-
A3
   DNS Servers . . . . . . . . . . . : 168.63.129.16
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

17. Ensure the primary private IP address used in windows is the same as the primary IP address of the Azure VM network interface. For more information, see No Internet access from Azure Windows VM that has multiple IP addresses.

Validation (Windows Server)

To validate connectivity to the internet from the secondary IP configuration via the public IP, use the following command. Replace 10.1.0.5 with the secondary private IP address you added to the Azure VM network interface.

PowerShell

```
ping -S 10.1.0.5 outlook.com
```

> **Note**
>
> For secondary IP configurations, you can ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

SUSE Linux Enterprise and openSUSE

▼ Expand

SUSE-based distributions use the `cloud-netconfig` plugin from the `cloud-netconfig-azure` package to manage the IP configuration. No manual steps are required on the part of the administrator. The first IP address of an interface set on the platform is assigned via DHCP. The cloud-netconfig plugin then probes the Azure Instance Metadata Service API continuously (once per minute) for more IP addresses assigned to the interface and adds/removes them as secondary IP addresses automatically.

This plugin should be installed and enabled on new images by default. Configuration steps for old workloads can be found here: https://www.suse.com/c/multi-nic-cloud-netconfig-ec2-azure/.

Ubuntu 14/16

▼ Expand

We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.

2. Ensure you're the root user. If you aren't, enter the following command:

```
Bash

sudo -i
```

3. Update the configuration file of the network interface (assuming **'eth0'**).

   - Keep the existing line item for dhcp. The primary IP address remains configured as it was previously.
   - Add a configuration for another static IP address with the following commands:

     ```
     Bash

     cd /etc/network/interfaces.d/
     ls
     ```

     You should see a .cfg file.

4. Open the file. You should see the following lines at the end of the file:

```
Bash

auto eth0
iface eth0 inet dhcp
```

5. Add the following lines after the lines that exist in the file. Replace `10.1.0.5` with your private IP address and subnet mask.

```
Bash

iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
```

   To add other private IP addresses, edit the file and add the new private IP addresses on subsequent lines:

```
Bash

iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
iface eth0 inet static
```

```
address 10.1.0.6
netmask 255.255.255.0
```

6. Save the file by using the following command:

```Bash
:wq
```

7. Reset the network interface with the following command:

```Bash
ifdown eth0 && ifup eth0
```

> **Important**
>
> Execute both ifdown and ifup in the same line if using a remote connection.

8. Verify the IP address is added to the network interface with the following command:

```Bash
ip addr list eth0
```

You should see the IP address you added as part of the list. Example:

```Bash
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Ubuntu 14/16)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

Bash

```
ping -I 10.1.0.5 outlook.com
```

> ⓘ **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Ubuntu 18.04+

▼ Expand

Starting on 18.04, `netplan` is used in Ubuntu for network management. We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.
2. Ensure you're the root user. If you are not, enter the following command:

   Bash

```
sudo -i
```

3. Create a file for the second interface and open it in a text editor:

```Bash
vi /etc/netplan/60-static.yaml
```

4. Add the following lines to the file, replacing `10.1.0.5/24` with your IP and subnet mask:

```Bash
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.1.0.5/24
```

   To add private IP addresses, edit the file and add the new private IP addresses on subsequent lines:

```Bash
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.1.0.5/24
                - 10.1.0.6/24
```

5. Save the file by using the following command:

```Bash
:wq
```

6. Test the changes with netplan try to confirm syntax:

```Bash
netplan try
```

> ⓘ Note

`netplan try` will apply the changes temporarily and roll back the changes after 120 seconds. If there's a loss of connectivity, wait 2 minutes, and then reconnect. At that time, the changes will have been rolled back.

7. Assuming no issues with **`netplan try`**, apply the configuration changes:

```
Bash

netplan apply
```

8. Verify the IP address is added to the network interface with the following command:

```
Bash

ip addr list eth0
```

You should see the IP address you added as part of the list. Example:

```
Bash

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Ubuntu 18.04+)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

```
Bash

ping -I 10.1.0.5 outlook.com
```

ⓘ Note

For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when trying to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. Follow the appropriate documentation for your Linux distribution. The following method is one way to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure you replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Red Hat Enterprise Linux and others

▼ Expand

ⓘ Note

To configure the extra IP addresses in RHEL10.x, it's enough to restart NetworkManger with: `systemctl restart NetworkManger.service` or reboot the system. No other steps are required.

1. Open a terminal window.
2. Ensure you're the root user. If you aren't, enter the following command:

   Bash

   ```
   sudo -i
   ```

3. Enter your password and follow instructions as prompted. Once you're the root user, go to the network scripts folder with the following command:

```bash
cd /etc/sysconfig/network-scripts
```

4. List the related ifcfg files using the following command:

```bash
ls ifcfg-*
```

   You should see **ifcfg-eth0** as one of the files.

5. Create a new configuration file for each IP added to the system.

```bash
touch ifcfg-eth0:0
```

6. Open the *ifcfg-eth0:0* file with the following command:

```bash
vi ifcfg-eth0:0
```

7. Add content to the file, **eth0:0** in this case, with the following command. Replace `10.1.0.5` with your new private IP address and subnet mask.

```bash
DEVICE=eth0:0
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.5
NETMASK=255.255.255.0
```

8. Save the file with the following command:

```bash
:wq
```

9. Create a config file per IP address to add with their corresponding values:

```bash
```

```
touch ifcfg-eth0:1
```

Bash

```
vi ifcfg-eth0:1
```

Bash

```
DEVICE=eth0:1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.6
NETMASK=255.255.255.0
```

Bash

```
:wq
```

10. Restart the network services and make sure the changes are successful by running the following commands:

Bash

```
systemctl restart NetworkManager.service
ifconfig
```

You should see the IP address or addresses you added in the list returned.

Bash

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.4  netmask 255.255.255.0  broadcast 10.1.0.255
    inet6 fe80::6245:bdff:fe7d:704a  prefixlen 64  scopeid 0x20<link>
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
    RX packets 858  bytes 244215 (238.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1021  bytes 262077 (255.9 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.5  netmask 255.255.255.0  broadcast 10.1.0.255
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.6  netmask 255.255.255.0  broadcast 10.1.0.255
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
```

Validation (Red Hat and others)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

Bash

```
ping -I 10.0.0.5 outlook.com
```

> ⓘ **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See the appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.0.0.5** with the private IP address that has a public IP address associated to it
  - **10.0.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Debian GNU/Linux

▼ Expand

We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.
2. Ensure you're the root user. If you aren't, enter the following command:

```bash
sudo -i
```

3. Update the configuration file of the network interface (assuming '**eth0**').

   - Keep the existing line item for dhcp. The primary IP address remains configured as it was previously.
   - Add a configuration for each static IP address using the following commands:

     ```bash
     cd /etc/network/interfaces.d/
     ls
     ```

     You should see a .cfg file.

4. Open the file. You should see the following lines at the end of the file:

```bash
auto eth0
iface eth0 inet dhcp
```

5. Add the following lines after the lines that exist in the file. Replace `10.1.0.5` with your private IP address and subnet mask.

```bash
iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
```

   Add the new IP addresses information in the configuration file:

```bash
iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
iface eth0 inet static
address 10.1.0.6
netmask 255.255.255.0
```

6. Save the file by using the following command:

```
Bash
:wq
```

7. Restart networking services for the changes to take effect. For Debian 8 and above, use:

```
Bash
systemctl restart networking
```

   For prior versions of Debian, you can use below commands:

```
Bash
service networking restart
```

8. Verify the IP address is added to the network interface with the following command:

```
Bash
ip addr list eth0
```

   You should see the IP address you added as part of the list. Example:

```
Bash
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Debian GNU/Linux)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

```
Bash
```

```
ping -I 10.1.0.5 outlook.com
```

> **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Next steps

- Learn more about public IP addresses in Azure.
- Learn more about private IP addresses in Azure.
- Learn how to Configure IP addresses for an Azure network interface.

Last updated on 02/19/2026

Assign multiple IP addresses to virtual machines using Azure PowerShell

An Azure Virtual Machine (VM) has one or more network interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it.

Assigning multiple IP addresses to a VM enables the following capabilities:

- Hosting multiple websites or services with different IP addresses and TLS/SSL certificates on a single server.
- Serve as a network virtual appliance, such as a firewall or load balancer.
- The ability to add any of the private IP addresses for any of the NICs to an Azure Load Balancer back-end pool. In the past, only the primary IP address for the primary NIC could be added to a back-end pool. For more information about load balancing multiple IP configurations, see Load balancing multiple IP configurations.

Every NIC attached to a VM has one or more IP configurations associated to it. Each configuration is assigned one static or dynamic private IP address. Each configuration may also have one public IP address resource associated to it. To learn more about IP addresses in Azure, see IP addresses in Azure.

> **Note**
>
> All IP configurations on a single NIC must be associated to the same subnet. If multiple IPs on different subnets are desired, multiple NICs on a VM can be used. To learn more about multiple NICs on a VM in Azure, see Create VM with Multiple NICs.

There's a limit to how many private IP addresses can be assigned to a NIC. There's also a limit to how many public IP addresses that can be used in an Azure subscription. See the Azure limits article for details.

This article explains how to add multiple IP addresses to a virtual machine using PowerShell.

Prerequisites

- An Azure account with an active subscription. Create an account for free.

- PowerShell environment in Azure Cloud Shell or Azure PowerShell installed locally. To learn more about using PowerShell in Azure Cloud Shell, see Azure Cloud Shell Quickstart.
  - If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 5.4.1 or later. Run `Get-InstalledModule -Name Az` to find the installed version. If you need to upgrade, see Install Azure PowerShell module. Ensure your Az.Network module is 4.3.0 or later. To verify the installed module, use the command `Get-InstalledModule -Name "Az.Network"`. If the module requires an update, use the command `Update-Module -Name "Az.Network"` if necessary.
- Sign in to Azure PowerShell and ensure you've selected the subscription with which you want to use this feature. For more information, see Sign in with Azure PowerShell.

> **Note**
>
> Though the steps in this article assigns all IP configurations to a single NIC, you can also assign multiple IP configurations to any NIC in a multi-NIC VM. To learn how to create a VM with multiple NICs, see Create a VM with multiple NICs.

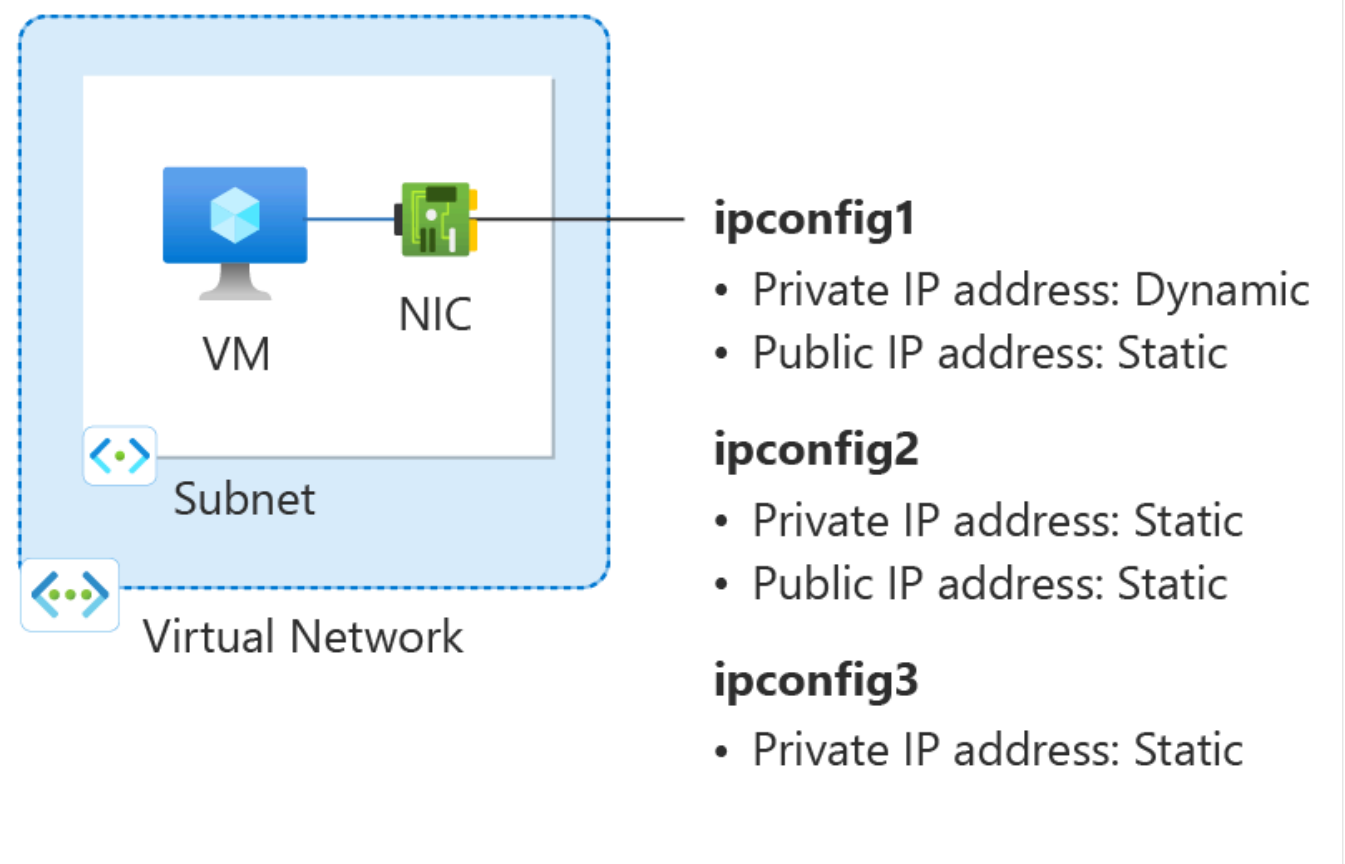


Figure: Diagram of network configuration resources created in this How-to article.

Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with New-AzResourceGroup named **myResourceGroup** in the **eastus2** location.

Azure PowerShell

```
$rg =@{
    Name = 'myResourceGroup'
    Location = 'eastus2'
}
New-AzResourceGroup @rg
```

Create a virtual network

In this section, you create a virtual network for the virtual machine.

Use New-AzVirtualNetwork and New-AzVirtualNetworkSubnetConfig to create a virtual network with one subnet.

Azure PowerShell

```
## Create backend subnet config ##
$subnet = @{
    Name = 'myBackendSubnet'
    AddressPrefix = '10.1.0.0/24'
}
$subnetConfig = New-AzVirtualNetworkSubnetConfig @subnet

## Create the virtual network ##
$vnet = @{
    Name = 'myVNet'
    ResourceGroupName = 'myResourceGroup'
    Location = 'eastus2'
    AddressPrefix = '10.1.0.0/16'
    Subnet = $subnetConfig
}
New-AzVirtualNetwork @vnet
```

Create a primary public IP address

Use New-AzPublicIpAddress to create a primary public IP address.

Azure PowerShell

```
$ip1 = @{
    Name = 'myPublicIP-1'
    ResourceGroupName = 'myResourceGroup'
    Location = 'eastus2'
    Sku = 'Standard'
    AllocationMethod = 'Static'
    IpAddressVersion = 'IPv4'
    Zone = 1,2,3
}
New-AzPublicIpAddress @ip1
```

Create a network security group

In this section, you create a network security group for the virtual machine and virtual network. You create a rule to allow connections to the virtual machine on port 22 for SSH.

Use New-AzNetworkSecurityGroup and New-AzNetworkSecurityRuleConfig to create the network security group and rules.

Azure PowerShell

```
## Create rule for network security group and place in variable. ##
$nsgrule1 = @{
    Name = 'myNSGRuleSSH'
    Description = 'Allow SSH'
    Protocol = '*'
    SourcePortRange = '*'
    DestinationPortRange = '22'
    SourceAddressPrefix = 'Internet'
    DestinationAddressPrefix = '*'
    Access = 'Allow'
    Priority = '200'
    Direction = 'Inbound'
}
$rule1 = New-AzNetworkSecurityRuleConfig @nsgrule1

## Create network security group ##
$nsg = @{
    Name = 'myNSG'
    ResourceGroupName = 'myResourceGroup'
    Location = 'eastus2'
    SecurityRules = $rule1
}
New-AzNetworkSecurityGroup @nsg
```

Create a network interface

Use New-AzNetworkInterface and New-AzNetworkInterfaceIpConfig to create a network interface (NIC) for the virtual machine. The public IP address and network security group created previously are associated with the network interface. The network interface is attached to the virtual network you created previously.

Azure PowerShell

```
## Place the virtual network into a variable. ##
$net = @{
    Name = 'myVNet'
    ResourceGroupName = 'myResourceGroup'
}
$vnet = Get-AzVirtualNetwork @net

## Place the network security group into a variable. ##
$ns = @{
    Name = 'myNSG'
    ResourceGroupName = 'myResourceGroup'
}
$nsg = Get-AzNetworkSecurityGroup @ns

## Place the primary public IP address into a variable. ##
$pub1 = @{
    Name = 'myPublicIP-1'
    ResourceGroupName = 'myResourceGroup'
}
$pubIP1 = Get-AzPublicIPAddress @pub1

## Create a primary IP configuration for the network interface. ##
$IP1 = @{
    Name = 'ipconfig1'
    Subnet = $vnet.Subnets[0]
    PrivateIpAddressVersion = 'IPv4'
    PublicIPAddress = $pubIP1
}
$IP1Config = New-AzNetworkInterfaceIpConfig @IP1 -Primary

## Create a secondary IP configuration for the network interface. ##
$IP3 = @{
    Name = 'ipconfig3'
    Subnet = $vnet.Subnets[0]
    PrivateIpAddressVersion = 'IPv4'
    PrivateIpAddress = '10.1.0.6'
}
$IP3Config = New-AzNetworkInterfaceIpConfig @IP3

## Command to create a network interface. ##
$nic = @{
    Name = 'myNIC1'
    ResourceGroupName = 'myResourceGroup'
    Location = 'eastus2'
    NetworkSecurityGroup = $nsg
```

```
    IpConfiguration = $IP1Config,$IP3Config
}
New-AzNetworkInterface @nic
```

> **Note**
>
> When adding a static IP address, you must specify an unused, valid address on the subnet the NIC is connected to.

Create a virtual machine

Use the following commands to create the virtual machine:

- New-AzVM
- New-AzVMConfig
- Set-AzVMOperatingSystem
- Set-AzVMSourceImage
- Add-AzVMNetworkInterface

Azure PowerShell

```
$cred = Get-Credential

## Place network interface into a variable. ##
$nic = @{
    Name = 'myNIC1'
    ResourceGroupName = 'myResourceGroup'
}
$nicVM = Get-AzNetworkInterface @nic

## Create a virtual machine configuration for VMs ##
$vmsz = @{
    VMName = 'myVM'
    VMSize = 'Standard_DS1_v2'
}
$vmos = @{
    ComputerName = 'myVM'
    Credential = $cred
}
$vmimage = @{
    PublisherName = 'Debian'
    Offer = 'debian-11'
    Skus = '11'
    Version = 'latest'
```

```
}
$vmConfig = New-AzVMConfig @vmsz `
      | Set-AzVMOperatingSystem @vmos -Linux `
      | Set-AzVMSourceImage @vmimage `
      | Add-AzVMNetworkInterface -Id $nicVM.Id

## Create the virtual machine for VMs ##
$vm = @{
    ResourceGroupName = 'myResourceGroup'
    Location = 'eastus2'
    VM = $vmConfig
    SshKeyName = 'mySSHKey'
    }
New-AzVM @vm -GenerateSshKey
```

Add secondary private and public IP address

Use New-AzPublicIpAddress to create a secondary public IP address.

Azure PowerShell

```
$ip2 = @{
    Name = 'myPublicIP-2'
    ResourceGroupName = 'myResourceGroup'
    Location = 'eastus2'
    Sku = 'Standard'
    AllocationMethod = 'Static'
    IpAddressVersion = 'IPv4'
    Zone = 1,2,3
}
New-AzPublicIpAddress @ip2
```

Use New-AzNetworkInterfaceIpConfig to create the secondary IP configuration for the virtual machine.

Azure PowerShell

```
## Place the virtual network into a variable. ##
$net = @{
    Name = 'myVNet'
    ResourceGroupName = 'myResourceGroup'
}
$vnet = Get-AzVirtualNetwork @net

## Place your virtual network subnet into a variable. ##
$sub = @{
    Name = 'myBackendSubnet'
    VirtualNetwork = $vnet
}
$subnet = Get-AzVirtualNetworkSubnetConfig @sub
```

```
## Place the secondary public IP address you created previously into a variable. ##
$pip = @{
    Name = 'myPublicIP-2'
    ResourceGroupName = 'myResourceGroup'
}
$pubIP2 = Get-AzPublicIPAddress @pip

## Place the network interface into a variable. ##
$net = @{
    Name = 'myNIC1'
    ResourceGroupName = 'myResourceGroup'
}
$nic = Get-AzNetworkInterface @net

## Create a secondary IP configuration for the network interface. ##
$IPc2 = @{
    Name = 'ipconfig2'
    Subnet = $vnet.Subnets[0]
    PrivateIpAddressVersion = 'IPv4'
    PrivateIpAddress = '10.1.0.5'
    PublicIPAddress = $pubIP2
}
$IP2Config = New-AzNetworkInterfaceIpConfig @IPc2

## Add the IP configuration to the network interface. ##
$nic.IpConfigurations.Add($IP2Config)

## Save the configuration to the network interface. ##
$nic | Set-AzNetworkInterface
```

Add IP addresses to a VM operating system

Connect and sign in to a VM you created with multiple private IP addresses. You must manually add all the private IP addresses, including the primary, that you added to the VM. Complete the following steps for your VM operating system.

Windows Server

▼ Expand

1. Open a command prompt or PowerShell.
2. Enter `ipconfig /all` at the command line. You see the **Primary** private IP address that was assigned through DHCP.
3. Enter `ncpa.cpl` at the command line to open the **Network Connections** configuration.
4. Open the **Properties** for the network adapter assigned the new IP addresses.

5. Double-click **Internet Protocol Version 4 (TCP/IPv4)**.

6. Select **Use the following IP address:**. Enter the following values.

Expand table

Setting	Value
IP address:	Enter the **Primary** private IP address.
Subnet mask:	Enter a subnet mask based on your IP address. For example, if the subnet is a **/24** subnet then the subnet mask is **255.255.255.0**.
Default gateway:	The first IP address in the subnet. If your subnet is **10.0.0.0/24**, then the gateway IP address is **10.0.0.1**.

7. Select **Use the following DNS server addresses:**. Enter the following values.

Expand table

Setting	Value
Preferred DNS server:	Enter your primary DNS server. Enter the IP address of **168.63.129.16** to use the default Azure provided DNS.

8. Select the **Advanced** button.

9. Select **Add**.

10. Enter the private **IP address** you added to the Azure network interface. Enter the corresponding **Subnet mask**. Select **Add**.

11. Repeat the previous steps to add any more private IP addresses that you added to the Azure network interface.

> **Important**
>
> You should never manually assign the public IP address assigned to an Azure virtual machine within the virtual machine's operating system. When you manually set the IP address within the operating system, ensure that it's the same address as the private IP address assigned to the Azure network interface. Failure to assign the address correctly

can cause loss of connectivity to the virtual machine. For more information, see Change IP address settings.

For more information about private IP addresses, see Private IP address.

12. Select **OK** to close the secondary IP address settings.
13. Select **OK** to close the adapter settings. Your RDP connection re-establishes.
14. Open a command prompt or PowerShell.
15. Enter `ipconfig /all` at the command line.
16. Verify the primary and secondary private IP addresses are present in the configuration.

PowerShell

```
PS C:\Users\azureuser> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : myVM
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Hyper-V Network Adapter
   Physical Address. . . . . . . . . : 00-0D-3A-E6-CE-A3
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::a8d1:11d5:3ab2:6a51%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.1.0.4(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.1.0.5(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.1.0.6(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.0.1
   DHCPv6 IAID . . . . . . . . . . . : 100666682
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-A8-26-B1-00-0D-3A-E6-CE-
A3
   DNS Servers . . . . . . . . . . . : 168.63.129.16
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

17. Ensure the primary private IP address used in windows is the same as the primary IP address of the Azure VM network interface. For more information, see No Internet access from Azure Windows VM that has multiple IP addresses.

Validation (Windows Server)

To validate connectivity to the internet from the secondary IP configuration via the public IP, use the following command. Replace 10.1.0.5 with the secondary private IP address you added to the Azure VM network interface.

PowerShell

```
ping -S 10.1.0.5 outlook.com
```

> **Note**
>
> For secondary IP configurations, you can ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

SUSE Linux Enterprise and openSUSE

▼ Expand

SUSE-based distributions use the `cloud-netconfig` plugin from the `cloud-netconfig-azure` package to manage the IP configuration. No manual steps are required on the part of the administrator. The first IP address of an interface set on the platform is assigned via DHCP. The cloud-netconfig plugin then probes the Azure Instance Metadata Service API continuously (once per minute) for more IP addresses assigned to the interface and adds/removes them as secondary IP addresses automatically.

This plugin should be installed and enabled on new images by default. Configuration steps for old workloads can be found here: https://www.suse.com/c/multi-nic-cloud-netconfig-ec2-azure/.

Ubuntu 14/16

▼ Expand

We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.

2. Ensure you're the root user. If you aren't, enter the following command:

```
Bash
sudo -i
```

3. Update the configuration file of the network interface (assuming '**eth0**').

   - Keep the existing line item for dhcp. The primary IP address remains configured as it was previously.

   - Add a configuration for another static IP address with the following commands:

     ```
     Bash
     cd /etc/network/interfaces.d/
     ls
     ```

     You should see a .cfg file.

4. Open the file. You should see the following lines at the end of the file:

```
Bash
auto eth0
iface eth0 inet dhcp
```

5. Add the following lines after the lines that exist in the file. Replace `10.1.0.5` with your private IP address and subnet mask.

```
Bash
iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
```

   To add other private IP addresses, edit the file and add the new private IP addresses on subsequent lines:

```
Bash
iface eth0 inet static
address 10.1.0.5
```

```
netmask 255.255.255.0
iface eth0 inet static
address 10.1.0.6
netmask 255.255.255.0
```

6. Save the file by using the following command:

Bash

```
:wq
```

7. Reset the network interface with the following command:

Bash

```
ifdown eth0 && ifup eth0
```

> **ⓘ Important**
>
> Execute both ifdown and ifup in the same line if using a remote connection.

8. Verify the IP address is added to the network interface with the following command:

Bash

```
ip addr list eth0
```

You should see the IP address you added as part of the list. Example:

Bash

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Ubuntu 14/16)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

Bash

```
ping -I 10.1.0.5 outlook.com
```

> ⓘ Note
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Ubuntu 18.04+

▼ Expand

Starting on 18.04, `netplan` is used in Ubuntu for network management. We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.
2. Ensure you're the root user. If you are not, enter the following command:

```bash
sudo -i
```

3. Create a file for the second interface and open it in a text editor:

```bash
vi /etc/netplan/60-static.yaml
```

4. Add the following lines to the file, replacing `10.1.0.5/24` with your IP and subnet mask:

```bash
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.1.0.5/24
```

   To add private IP addresses, edit the file and add the new private IP addresses on subsequent lines:

```bash
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.1.0.5/24
                - 10.1.0.6/24
```

5. Save the file by using the following command:

```bash
:wq
```

6. Test the changes with netplan try to confirm syntax:

```bash
netplan try
```

> ⓘ Note
>
> `netplan try` will apply the changes temporarily and roll back the changes after 120 seconds. If there's a loss of connectivity, wait 2 minutes, and then reconnect. At that time, the changes will have been rolled back.

7. Assuming no issues with **`netplan try`**, apply the configuration changes:

```
Bash

netplan apply
```

8. Verify the IP address is added to the network interface with the following command:

```
Bash

ip addr list eth0
```

   You should see the IP address you added as part of the list. Example:

```
Bash

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Ubuntu 18.04+)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

```
Bash

ping -I 10.1.0.5 outlook.com
```

> ⓘ Note
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when trying to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. Follow the appropriate documentation for your Linux distribution. The following method is one way to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure you replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Red Hat Enterprise Linux and others

▼ Expand

> ⓘ Note
>
> To configure the extra IP addresses in RHEL10.x, it's enough to restart NetworkManger with: `systemctl restart NetworkManger.service` or reboot the system. No other steps are required.

RHEL 8.6 & above, RHEL 9

Starting RHEL 8.6 & above and RHEL 9, "networkmanager-cloud-setup" package can handle multiple IP configurations and its associated route configuration.

1. Disabled cloud-init to handle secondary IP configuration in the respective files /etc/cloud/cloud.cfg.d/99-apply-network-config.cfg or /etc/cloud/cloud.cfg.d/91-azure_datasource.cfg. This will let NetworkManager and nm-cloud-setup manage the entire network configuration

Bash

```
[root@rhel8 ~]# cat /etc/cloud/cloud.cfg.d/99-apply-network-config.cfg
datasource:
  Azure:
     apply_network_config: False
```

2. Install "NetworkManager-cloud-setup" package if not already installed

Bash

```
]# yum install NetworkManager-cloud-setup
Red Hat Enterprise Linux 8 for x86_64 - BaseOS from RHUI (RPMs)
39 MB/s |  66 MB     00:01
Red Hat Enterprise Linux 8 for x86_64 - Supplementary (RPMs) from RHUI
2.3 MB/s | 340 kB     00:00
Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (RPMs)
41 MB/s |  60 MB     00:01
Red Hat CodeReady Linux Builder for RHEL 8 x86_64 (RPMs) from RHUI
33 MB/s | 8.9 MB     00:00
Red Hat Ansible Engine 2 for RHEL 8 x86_64 (RPMs) from RHUI
14 MB/s | 2.5 MB     00:00
Dependencies resolved.
================================================================================
================================================================================
========================
 Package                                          Architecture
Version                                  Repository
Size
================================================================================
================================================================================
========================
Installing:
 NetworkManager-cloud-setup                       x86_64
1:1.40.16-13.el8_9                       rhel-8-for-x86_64-appstream-rhui-rpms
198 k
Upgrading:
 NetworkManager                                   x86_64
1:1.40.16-13.el8_9                       rhel-8-for-x86_64-baseos-rhui-rpms
2.3 M
 NetworkManager-libnm                             x86_64
1:1.40.16-13.el8_9                       rhel-8-for-x86_64-baseos-rhui-rpms
1.9 M
 NetworkManager-team                              x86_64
1:1.40.16-13.el8_9                       rhel-8-for-x86_64-baseos-rhui-rpms
161 k
```

```
 NetworkManager-tui                                  x86_64
1:1.40.16-13.el8_9                        rhel-8-for-x86_64-baseos-rhui-rpms
356 k

Transaction Summary
================================================================================
================================================================================
========================
Install  1 Package
Upgrade  4 Packages

Total download size: 4.9 M
Is this ok [y/N]: y
Downloading Packages:
(1/5): NetworkManager-cloud-setup-1.40.16-13.el8_9.x86_64.rpm
2.0 MB/s | 198 kB     00:00
(2/5): NetworkManager-1.40.16-13.el8_9.x86_64.rpm
18 MB/s | 2.3 MB     00:00
(3/5): NetworkManager-team-1.40.16-13.el8_9.x86_64.rpm
4.1 MB/s | 161 kB     00:00
(4/5): NetworkManager-libnm-1.40.16-13.el8_9.x86_64.rpm
12 MB/s | 1.9 MB     00:00
(5/5): NetworkManager-tui-1.40.16-13.el8_9.x86_64.rpm
8.8 MB/s | 356 kB     00:00
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
------------------------
Total
28 MB/s | 4.9 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
1/1
  Running scriptlet: NetworkManager-libnm-1:1.40.16-13.el8_9.x86_64
1/1
  Upgrading        : NetworkManager-libnm-1:1.40.16-13.el8_9.x86_64
1/9
  Running scriptlet: NetworkManager-libnm-1:1.40.16-13.el8_9.x86_64
1/9
  Running scriptlet: NetworkManager-1:1.40.16-13.el8_9.x86_64
2/9
  Upgrading        : NetworkManager-1:1.40.16-13.el8_9.x86_64
2/9
  Running scriptlet: NetworkManager-1:1.40.16-13.el8_9.x86_64
2/9
  Installing       : NetworkManager-cloud-setup-1:1.40.16-13.el8_9.x86_64
3/9
  Running scriptlet: NetworkManager-cloud-setup-1:1.40.16-13.el8_9.x86_64
3/9
  Upgrading        : NetworkManager-team-1:1.40.16-13.el8_9.x86_64
4/9
  Upgrading        : NetworkManager-tui-1:1.40.16-13.el8_9.x86_64
```

```
5/9
  Cleanup          : NetworkManager-tui-1:1.40.0-1.el8.x86_64
6/9
  Cleanup          : NetworkManager-team-1:1.40.0-1.el8.x86_64
7/9
  Running scriptlet: NetworkManager-1:1.40.0-1.el8.x86_64
8/9
  Cleanup          : NetworkManager-1:1.40.0-1.el8.x86_64
8/9
  Running scriptlet: NetworkManager-1:1.40.0-1.el8.x86_64
8/9
  Cleanup          : NetworkManager-libnm-1:1.40.0-1.el8.x86_64
9/9
  Running scriptlet: NetworkManager-libnm-1:1.40.0-1.el8.x86_64
9/9
  Verifying        : NetworkManager-cloud-setup-1:1.40.16-13.el8_9.x86_64
1/9
  Verifying        : NetworkManager-1:1.40.16-13.el8_9.x86_64
2/9
  Verifying        : NetworkManager-1:1.40.0-1.el8.x86_64
3/9
  Verifying        : NetworkManager-libnm-1:1.40.16-13.el8_9.x86_64
4/9
  Verifying        : NetworkManager-libnm-1:1.40.0-1.el8.x86_64
5/9
  Verifying        : NetworkManager-team-1:1.40.16-13.el8_9.x86_64
6/9
  Verifying        : NetworkManager-team-1:1.40.0-1.el8.x86_64
7/9
  Verifying        : NetworkManager-tui-1:1.40.16-13.el8_9.x86_64
8/9
  Verifying        : NetworkManager-tui-1:1.40.0-1.el8.x86_64
9/9
Installed products updated.

Upgraded:
  NetworkManager-1:1.40.16-13.el8_9.x86_64   NetworkManager-libnm-1:1.40.16-
13.el8_9.x86_64   NetworkManager-team-1:1.40.16-13.el8_9.x86_64   NetworkManager-
tui-1:1.40.16-13.el8_9.x86_64
Installed:
  NetworkManager-cloud-setup-1:1.40.16-13.el8_9.x86_64

Complete!
```

3. To enable “nm-cloud-setup” to handle secondary IP configuration, create nm-cloud-setup.service and nm-cloud-setup.timer (adjust OnBootSec & OnUnitActiveSec parameters to suit your requirement) files

Bash

```
[root@rhel8 ~]# cat /etc/systemd/system/nm-cloud-setup.service
[Unit]
```

```
Description=Automatically configure NetworkManager in cloud
Documentation=man:nm-cloud-setup(8)
Before=network-online.target
After=NetworkManager.service

[Service]
Environment=NM_CLOUD_SETUP_AZURE=yes
Type=oneshot
ExecStart=/usr/libexec/nm-cloud-setup

[Install]
WantedBy=NetworkManager.service
```

Bash

```
[root@rhel8 ~]# cat /etc/systemd/system/nm-cloud-setup.timer
[Unit]
Description=Periodically run nm-cloud-setup

[Timer]
OnBootSec=1min
OnUnitActiveSec=1min

[Install]
WantedBy=timers.target
```

4. Enable services are reload daemon

Bash

```
systemctl enable --now nm-cloud-setup.service
systemctl start nm-cloud-setup.service
systemctl enable --now nm-cloud-setup.timer
systemctl daemon-reload
```

5. Reboot VM & verify. VM should now successfully identify respective primary and secondary IPs.

RHEL 8.5 & below

1. Open a terminal window.

2. Ensure you're the root user. If you aren't, enter the following command:

   Bash

   ```
   sudo -i
   ```

3. Enter your password and follow instructions as prompted. Once you're the root user, go to the network scripts folder with the following command:

```Bash
cd /etc/sysconfig/network-scripts
```

4. List the related ifcfg files using the following command:

```Bash
ls ifcfg-*
```

   You should see **ifcfg-eth0** as one of the files.

5. Create a new configuration file for each IP added to the system.

```Bash
touch ifcfg-eth0:0
```

6. Open the *ifcfg-eth0:0* file with the following command:

```Bash
vi ifcfg-eth0:0
```

7. Add content to the file, **eth0:0** in this case, with the following command. Replace `10.1.0.5` with your new private IP address and subnet mask.

```Bash
DEVICE=eth0:0
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.5
NETMASK=255.255.255.0
```

8. Save the file with the following command:

```Bash
:wq
```

9. Create a config file per IP address to add with their corresponding values:

Bash

```
touch ifcfg-eth0:1
```

Bash

```
vi ifcfg-eth0:1
```

Bash

```
DEVICE=eth0:1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.6
NETMASK=255.255.255.0
```

Bash

```
:wq
```

10. Restart the network services and make sure the changes are successful by running the following commands:

Bash

```
systemctl restart NetworkManager.service
ifconfig
```

You should see the IP address or addresses you added in the list returned.

Bash

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.4  netmask 255.255.255.0  broadcast 10.1.0.255
    inet6 fe80::6245:bdff:fe7d:704a  prefixlen 64  scopeid 0x20<link>
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
    RX packets 858  bytes 244215 (238.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1021  bytes 262077 (255.9 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.5  netmask 255.255.255.0  broadcast 10.1.0.255
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
```

```
eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.6  netmask 255.255.255.0  broadcast 10.1.0.255
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
```

Validation (Red Hat and others)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

Bash

```
ping -I 10.0.0.5 outlook.com
```

> ⓘ **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See the appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.0.0.5** with the private IP address that has a public IP address associated to it
  - **10.0.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Debian GNU/Linux

▼ Expand

We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.
2. Ensure you're the root user. If you aren't, enter the following command:

   Bash

   ```
   sudo -i
   ```

3. Update the configuration file of the network interface (assuming **'eth0'**).

   - Keep the existing line item for dhcp. The primary IP address remains configured as it was previously.
   - Add a configuration for each static IP address using the following commands:

     Bash

     ```
     cd /etc/network/interfaces.d/
     ls
     ```

     You should see a .cfg file.

4. Open the file. You should see the following lines at the end of the file:

   Bash

   ```
   auto eth0
   iface eth0 inet dhcp
   ```

5. Add the following lines after the lines that exist in the file. Replace `10.1.0.5` with your private IP address and subnet mask.

   Bash

   ```
   iface eth0 inet static
   address 10.1.0.5
   netmask 255.255.255.0
   ```

   Add the new IP addresses information in the configuration file:

   Bash

```
iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
iface eth0 inet static
address 10.1.0.6
netmask 255.255.255.0
```

6. Save the file by using the following command:

Bash

```
:wq
```

7. Restart networking services for the changes to take effect. For Debian 8 and above, use:

Bash

```
systemctl restart networking
```

For prior versions of Debian, you can use below commands:

Bash

```
service networking restart
```

8. Verify the IP address is added to the network interface with the following command:

Bash

```
ip addr list eth0
```

You should see the IP address you added as part of the list. Example:

Bash

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Debian GNU/Linux)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

```Bash
ping -I 10.1.0.5 outlook.com
```

> **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See appropriate documentation for your Linux distribution. The following method to accomplish this goal:

```Bash
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Next steps

- Learn more about public IP addresses in Azure.
- Learn more about private IP addresses in Azure.
- Learn how to Configure IP addresses for an Azure network interface.

Last updated on 10/08/2025

Assign multiple IP addresses to virtual machines using the Azure CLI

Summarize this article for me

An Azure Virtual Machine (VM) has one or more network interfaces (NIC) attached to it. Any NIC can have one or more static or dynamic public and private IP addresses assigned to it.

Assigning multiple IP addresses to a VM enables the following capabilities:

- Hosting multiple websites or services with different IP addresses and TLS/SSL certificates on a single server.
- Serve as a network virtual appliance, such as a firewall or load balancer.
- The ability to add any of the private IP addresses for any of the NICs to an Azure Load Balancer back-end pool. In the past, only the primary IP address for the primary NIC could be added to a back-end pool. For more information about load balancing multiple IP configurations, see Load balancing multiple IP configurations.

Every NIC attached to a VM has one or more IP configurations associated to it. Each configuration is assigned one static or dynamic private IP address. Each configuration may also have one public IP address resource associated to it. To learn more about IP addresses in Azure, see IP addresses in Azure.

> **Note**
>
> All IP configurations on a single NIC must be associated to the same subnet. If multiple IPs on different subnets are desired, multiple NICs on a VM can be used. To learn more about multiple NICs on a VM in Azure, see Create VM with Multiple NICs.

There's a limit to how many private IP addresses can be assigned to a NIC. There's also a limit to how many public IP addresses that can be used in an Azure subscription. See Azure limits for details.

This article explains how to add multiple IP addresses to a virtual machine using the Azure CLI.

Prerequisites

- An Azure account with an active subscription. Create an account for free.

- Use the Bash environment in Azure Cloud Shell. For more information, see Get started with Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Authenticate to Azure using Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use and manage extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.
- This tutorial requires version 2.0.28 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

> **Note**
>
> Though the steps in this article assigns all IP configurations to a single NIC, you can also assign multiple IP configurations to any NIC in a multi-NIC VM. To learn how to create a VM with multiple NICs, see Create a VM with multiple NICs.

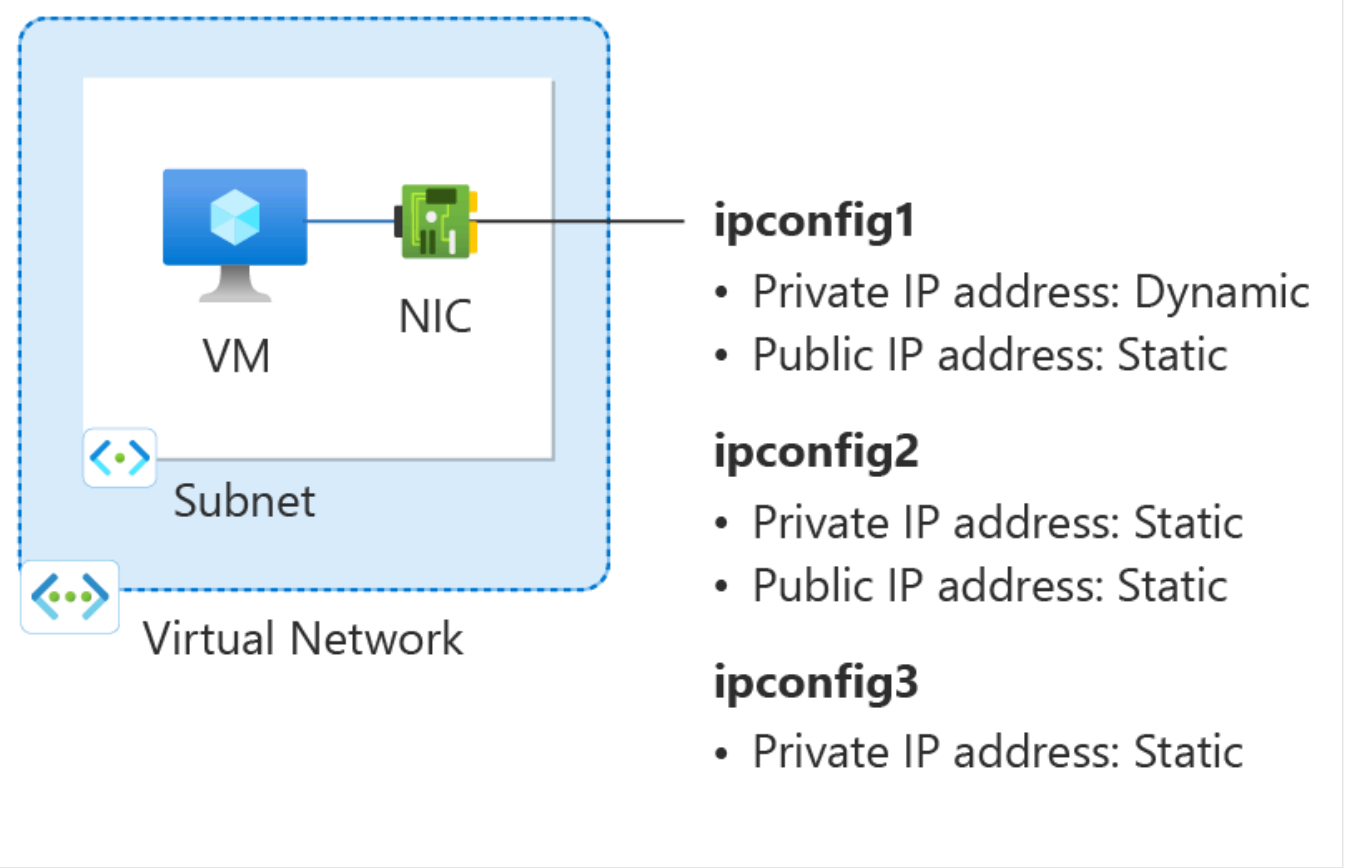


Figure: Diagram of network configuration resources created in this How-to article.

Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed.

Create a resource group with az group create named **myResourceGroup** in the **eastus2** location.

Azure CLI

```
az group create \
  --name myResourceGroup \
  --location eastus2
```

Create a virtual network

In this section, you create a virtual network for the virtual machine.

Use az network vnet create to create a virtual network.

Azure CLI

```
az network vnet create \
  --resource-group myResourceGroup \
  --location eastus2 \
  --name myVNet \
  --address-prefixes 10.1.0.0/16 \
  --subnet-name myBackendSubnet \
  --subnet-prefixes 10.1.0.0/24
```

Create public IP addresses

Use az network public-ip create to create two public IP addresses.

Azure CLI

```
az network public-ip create \
  --resource-group myResourceGroup \
  --name myPublicIP-1 \
  --sku Standard \
  --version IPv4 \
  --zone 1 2 3

az network public-ip create \
  --resource-group myResourceGroup \
  --name myPublicIP-2 \
  --sku Standard \
  --version IPv4 \
  --zone 1 2 3
```

Create a network security group

In this section, you create a network security group for the virtual machine and virtual network.

Use az network nsg create to create the network security group.

Azure CLI

```
az network nsg create \
  --resource-group myResourceGroup \
  --name myNSG
```

> **Note**
>
> The default rules of the network security group block all inbound access from the internet, including SSH. To connect to the virtual machine, use Azure Bastion. For more information,

see Quickstart: Deploy Azure Bastion with default settings.

Create a network interface

You use az network nic create to create the network interface for the virtual machine. The public IP addresses and the NSG created previously are associated with the NIC. The network interface is attached to the virtual network you created previously.

Azure CLI

```
az network nic create \
  --resource-group myResourceGroup \
  --name myNIC1 \
  --private-ip-address-version IPv4 \
  --vnet-name myVNet \
  --subnet myBackEndSubnet \
  --network-security-group myNSG \
  --public-ip-address myPublicIP-1
```

Create secondary private and public IP configuration

Use az network nic ip-config create to create the secondary private and public IP configuration for the NIC. Replace **10.1.0.5** with your secondary private IP address.

Azure CLI

```
az network nic ip-config create \
  --resource-group myResourceGroup \
  --name ipconfig2 \
  --nic-name myNIC1 \
  --private-ip-address 10.1.0.5 \
  --private-ip-address-version IPv4 \
  --vnet-name myVNet \
  --subnet myBackendSubnet \
  --public-ip-address myPublicIP-2
```

Create tertiary private IP configuration

Use az network nic ip-config create to create the tertiary private IP configuration for the NIC. Replace **10.1.0.6** with your secondary private IP address.

Azure CLI

```
az network nic ip-config create \
  --resource-group myResourceGroup \
  --name ipconfig3 \
  --nic-name myNIC1 \
  --private-ip-address 10.1.0.6 \
  --private-ip-address-version IPv4 \
  --vnet-name myVNet \
  --subnet myBackendSubnet
```

> **Note**
>
> When adding a static IP address, you must specify an unused, valid address on the subnet the NIC is connected to.

Create a virtual machine

Use az vm create to create the virtual machine.

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --nics myNIC1 \
  --image Ubuntu2204 \
  --admin-username azureuser \
  --authentication-type ssh \
  --generate-ssh-keys
```

Add IP addresses to a VM operating system

Connect and sign in to a VM you created with multiple private IP addresses. You must manually add all the private IP addresses, including the primary, that you added to the VM. Complete the following steps for your VM operating system.

Windows Server

▼ Expand

1. Open a command prompt or PowerShell.
2. Enter `ipconfig /all` at the command line. You see the **Primary** private IP address that was assigned through DHCP.

3. Enter `ncpa.cpl` at the command line to open the **Network Connections** configuration.
4. Open the **Properties** for the network adapter assigned the new IP addresses.
5. Double-click **Internet Protocol Version 4 (TCP/IPv4)**.
6. Select **Use the following IP address:**. Enter the following values.

Expand table

Setting	Value
IP address:	Enter the **Primary** private IP address.
Subnet mask:	Enter a subnet mask based on your IP address. For example, if the subnet is a **/24** subnet then the subnet mask is **255.255.255.0**.
Default gateway:	The first IP address in the subnet. If your subnet is **10.0.0.0/24**, then the gateway IP address is **10.0.0.1**.

7. Select **Use the following DNS server addresses:**. Enter the following values.

Expand table

Setting	Value
Preferred DNS server:	Enter your primary DNS server. Enter the IP address of **168.63.129.16** to use the default Azure provided DNS.

8. Select the **Advanced** button.
9. Select **Add**.
10. Enter the private **IP address** you added to the Azure network interface. Enter the corresponding **Subnet mask**. Select **Add**.
11. Repeat the previous steps to add any more private IP addresses that you added to the Azure network interface.

> **Important**
>
> You should never manually assign the public IP address assigned to an Azure virtual machine within the virtual machine's operating system. When you manually set the IP address within the operating system, ensure that it's the same address as the private IP

address assigned to the Azure network interface. Failure to assign the address correctly can cause loss of connectivity to the virtual machine. For more information, see Change IP address settings.

For more information about private IP addresses, see Private IP address.

12. Select **OK** to close the secondary IP address settings.
13. Select **OK** to close the adapter settings. Your RDP connection re-establishes.
14. Open a command prompt or PowerShell.
15. Enter `ipconfig /all` at the command line.
16. Verify the primary and secondary private IP addresses are present in the configuration.

PowerShell

```
PS C:\Users\azureuser> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : myVM
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Hyper-V Network Adapter
   Physical Address. . . . . . . . . : 00-0D-3A-E6-CE-A3
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::a8d1:11d5:3ab2:6a51%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.1.0.4(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.1.0.5(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   IPv4 Address. . . . . . . . . . . : 10.1.0.6(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.0.1
   DHCPv6 IAID . . . . . . . . . . . : 100666682
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-A8-26-B1-00-0D-3A-E6-CE-
A3
   DNS Servers . . . . . . . . . . . : 168.63.129.16
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

17. Ensure the primary private IP address used in windows is the same as the primary IP address of the Azure VM network interface. For more information, see No Internet access from Azure Windows VM that has multiple IP addresses.

Validation (Windows Server)

To validate connectivity to the internet from the secondary IP configuration via the public IP, use the following command. Replace 10.1.0.5 with the secondary private IP address you added to the Azure VM network interface.

PowerShell

```
ping -S 10.1.0.5 outlook.com
```

> **Note**
>
> For secondary IP configurations, you can ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

SUSE Linux Enterprise and openSUSE

▼ Expand

SUSE-based distributions use the `cloud-netconfig` plugin from the `cloud-netconfig-azure` package to manage the IP configuration. No manual steps are required on the part of the administrator. The first IP address of an interface set on the platform is assigned via DHCP. The cloud-netconfig plugin then probes the Azure Instance Metadata Service API continuously (once per minute) for more IP addresses assigned to the interface and adds/removes them as secondary IP addresses automatically.

This plugin should be installed and enabled on new images by default. Configuration steps for old workloads can be found here: https://www.suse.com/c/multi-nic-cloud-netconfig-ec2-azure/.

Ubuntu 14/16

▼ Expand

We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.

2. Ensure you're the root user. If you aren't, enter the following command:

```
Bash

sudo -i
```

3. Update the configuration file of the network interface (assuming **'eth0'**).

   - Keep the existing line item for dhcp. The primary IP address remains configured as it was previously.
   - Add a configuration for another static IP address with the following commands:

     ```
     Bash

     cd /etc/network/interfaces.d/
     ls
     ```

     You should see a .cfg file.

4. Open the file. You should see the following lines at the end of the file:

```
Bash

auto eth0
iface eth0 inet dhcp
```

5. Add the following lines after the lines that exist in the file. Replace `10.1.0.5` with your private IP address and subnet mask.

```
Bash

iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
```

   To add other private IP addresses, edit the file and add the new private IP addresses on subsequent lines:

```
Bash

iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
iface eth0 inet static
```

```
address 10.1.0.6
netmask 255.255.255.0
```

6. Save the file by using the following command:

```Bash
:wq
```

7. Reset the network interface with the following command:

```Bash
ifdown eth0 && ifup eth0
```

> **Important**
>
> Execute both ifdown and ifup in the same line if using a remote connection.

8. Verify the IP address is added to the network interface with the following command:

```Bash
ip addr list eth0
```

You should see the IP address you added as part of the list. Example:

```Bash
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Ubuntu 14/16)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

Bash

```
ping -I 10.1.0.5 outlook.com
```

> ⓘ **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Ubuntu 18.04+

▼ Expand

Starting on 18.04, `netplan` is used in Ubuntu for network management. We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.
2. Ensure you're the root user. If you are not, enter the following command:

   Bash

```
sudo -i
```

3. Create a file for the second interface and open it in a text editor:

```Bash
vi /etc/netplan/60-static.yaml
```

4. Add the following lines to the file, replacing `10.1.0.5/24` with your IP and subnet mask:

```Bash
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.1.0.5/24
```

   To add private IP addresses, edit the file and add the new private IP addresses on subsequent lines:

```Bash
network:
    version: 2
    ethernets:
        eth0:
            addresses:
                - 10.1.0.5/24
                - 10.1.0.6/24
```

5. Save the file by using the following command:

```Bash
:wq
```

6. Test the changes with netplan try to confirm syntax:

```Bash
netplan try
```

> ⓘ Note

`netplan try` will apply the changes temporarily and roll back the changes after 120 seconds. If there's a loss of connectivity, wait 2 minutes, and then reconnect. At that time, the changes will have been rolled back.

7. Assuming no issues with **`netplan try`**, apply the configuration changes:

```
Bash

netplan apply
```

8. Verify the IP address is added to the network interface with the following command:

```
Bash

ip addr list eth0
```

You should see the IP address you added as part of the list. Example:

```
Bash

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Ubuntu 18.04+)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

```
Bash

ping -I 10.1.0.5 outlook.com
```

ⓘ Note

For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when trying to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. Follow the appropriate documentation for your Linux distribution. The following method is one way to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure you replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Red Hat Enterprise Linux and others

▼ Expand

ⓘ Note

To configure the extra IP addresses in RHEL10.x, it's enough to restart NetworkManger with: `systemctl restart NetworkManger.service` or reboot the system. No other steps are required.

1. Open a terminal window.
2. Ensure you're the root user. If you aren't, enter the following command:

   Bash

   ```
   sudo -i
   ```

3. Enter your password and follow instructions as prompted. Once you're the root user, go to the network scripts folder with the following command:

```bash
cd /etc/sysconfig/network-scripts
```

4. List the related ifcfg files using the following command:

```bash
ls ifcfg-*
```

   You should see **ifcfg-eth0** as one of the files.

5. Create a new configuration file for each IP added to the system.

```bash
touch ifcfg-eth0:0
```

6. Open the *ifcfg-eth0:0* file with the following command:

```bash
vi ifcfg-eth0:0
```

7. Add content to the file, **eth0:0** in this case, with the following command. Replace `10.1.0.5` with your new private IP address and subnet mask.

```bash
DEVICE=eth0:0
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.5
NETMASK=255.255.255.0
```

8. Save the file with the following command:

```bash
:wq
```

9. Create a config file per IP address to add with their corresponding values:

```bash
```

```
touch ifcfg-eth0:1
```

Bash

```
vi ifcfg-eth0:1
```

Bash

```
DEVICE=eth0:1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.6
NETMASK=255.255.255.0
```

Bash

```
:wq
```

10. Restart the network services and make sure the changes are successful by running the following commands:

Bash

```
systemctl restart NetworkManager.service
ifconfig
```

You should see the IP address or addresses you added in the list returned.

Bash

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.4  netmask 255.255.255.0  broadcast 10.1.0.255
    inet6 fe80::6245:bdff:fe7d:704a  prefixlen 64  scopeid 0x20<link>
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
    RX packets 858  bytes 244215 (238.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1021  bytes 262077 (255.9 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.5  netmask 255.255.255.0  broadcast 10.1.0.255
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.1.0.6  netmask 255.255.255.0  broadcast 10.1.0.255
    ether 60:45:bd:7d:70:4a  txqueuelen 1000  (Ethernet)
```

Validation (Red Hat and others)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

Bash

```
ping -I 10.0.0.5 outlook.com
```

> ⓘ **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See the appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.0.0.5** with the private IP address that has a public IP address associated to it
  - **10.0.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Debian GNU/Linux

▼ Expand

We recommend looking at the latest documentation for your Linux distribution.

1. Open a terminal window.
2. Ensure you're the root user. If you aren't, enter the following command:

```bash
sudo -i
```

3. Update the configuration file of the network interface (assuming '**eth0**').

   - Keep the existing line item for dhcp. The primary IP address remains configured as it was previously.
   - Add a configuration for each static IP address using the following commands:

     ```bash
     cd /etc/network/interfaces.d/
     ls
     ```

     You should see a .cfg file.

4. Open the file. You should see the following lines at the end of the file:

```bash
auto eth0
iface eth0 inet dhcp
```

5. Add the following lines after the lines that exist in the file. Replace `10.1.0.5` with your private IP address and subnet mask.

```bash
iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
```

   Add the new IP addresses information in the configuration file:

```bash
iface eth0 inet static
address 10.1.0.5
netmask 255.255.255.0
iface eth0 inet static
address 10.1.0.6
netmask 255.255.255.0
```

6. Save the file by using the following command:

```
Bash
:wq
```

7. Restart networking services for the changes to take effect. For Debian 8 and above, use:

```
Bash
systemctl restart networking
```

   For prior versions of Debian, you can use below commands:

```
Bash
service networking restart
```

8. Verify the IP address is added to the network interface with the following command:

```
Bash
ip addr list eth0
```

   You should see the IP address you added as part of the list. Example:

```
Bash
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
link/ether 00:0d:3a:04:45:16 brd ff:ff:ff:ff:ff:ff
inet 10.1.0.5/24 brd 10.1.0.255 scope global eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.6/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet 10.1.0.4/24 brd 10.1.0.255 scope global secondary eth0
   valid_lft forever preferred_lft forever
inet6 fe80::20d:3aff:fe04:4516/64 scope link
   valid_lft forever preferred_lft forever
```

Validation (Debian GNU/Linux)

To ensure you're able to connect to the internet from your secondary IP configuration via the public IP associated with it, use the following command:

```
Bash
```

```
ping -I 10.1.0.5 outlook.com
```

> **Note**
>
> For secondary IP configurations, you can only ping to the Internet if the configuration has a public IP address associated with it. For primary IP configurations, a public IP address isn't required to ping to the Internet.

For Linux VMs, when attempting to validate outbound connectivity from a secondary NIC, you may need to add appropriate routes. See appropriate documentation for your Linux distribution. The following method to accomplish this goal:

Bash

```
echo 150 custom >> /etc/iproute2/rt_tables

ip rule add from 10.1.0.5 lookup custom
ip route add default via 10.1.0.1 dev eth2 table custom
```

- Ensure to replace:
  - **10.1.0.5** with the private IP address that has a public IP address associated to it
  - **10.1.0.1** to your default gateway
  - **eth2** to the name of your secondary NIC

Next steps

- Learn more about public IP addresses in Azure.
- Learn more about private IP addresses in Azure.
- Learn how to Configure IP addresses for an Azure network interface.

Last updated on 02/25/2026

Create and manage a Windows virtual machine that has multiple NICs

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

Virtual machines (VMs) in Azure can have multiple virtual network interface cards (NICs) attached to them. A common scenario is to have different subnets for front-end and back-end connectivity. You can associate multiple NICs on a VM to multiple subnets, but those subnets must all reside in the same virtual network (vNet). This article details how to create a VM that has multiple NICs attached to it. You also learn how to add or remove NICs from an existing VM. Different VM sizes support a varying number of NICs, so size your VM accordingly.

> **Note**
>
> If multiple subnets are not required for a scenario, it may be more straightforward to utilize multiple IP configurations on a single NIC. Instructions for this setup can be found here.

Prerequisites

In the following examples, replace example parameter names with your own values. Example parameter names include *myResourceGroup*, *myVnet*, and *myVM*.

Create a VM with multiple NICs

First, create a resource group. The following example creates a resource group named *myResourceGroup* in the *EastUs* location:

PowerShell

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create virtual network and subnets

A common scenario is for a virtual network to have two or more subnets. One subnet may be for front-end traffic, the other for back-end traffic. To connect to both subnets,

you then use multiple NICs on your VM.

1. Define two virtual network subnets with New-AzVirtualNetworkSubnetConfig. The following example defines the subnets for *mySubnetFrontEnd* and *mySubnetBackEnd*:

   PowerShell

   ```
   $mySubnetFrontEnd = New-AzVirtualNetworkSubnetConfig -Name
   "mySubnetFrontEnd" `
       -AddressPrefix "192.168.1.0/24"
   $mySubnetBackEnd = New-AzVirtualNetworkSubnetConfig -Name
   "mySubnetBackEnd" `
       -AddressPrefix "192.168.2.0/24"
   ```

2. Create your virtual network and subnets with New-AzVirtualNetwork. The following example creates a virtual network named *myVnet*:

   PowerShell

   ```
   $myVnet = New-AzVirtualNetwork -ResourceGroupName "myResourceGroup" `
       -Location "EastUs" `
       -Name "myVnet" `
       -AddressPrefix "192.168.0.0/16" `
       -Subnet $mySubnetFrontEnd,$mySubnetBackEnd
   ```

Create multiple NICs

Create two NICs with New-AzNetworkInterface. Attach one NIC to the front-end subnet and one NIC to the back-end subnet. The following example creates NICs named *myNic1* and *myNic2*:

PowerShell

```
$frontEnd = $myVnet.Subnets|?{$_.Name -eq 'mySubnetFrontEnd'}
$myNic1 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
    -Name "myNic1" `
    -Location "EastUs" `
    -SubnetId $frontEnd.Id

$backEnd = $myVnet.Subnets|?{$_.Name -eq 'mySubnetBackEnd'}
$myNic2 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
    -Name "myNic2" `
    -Location "EastUs" `
    -SubnetId $backEnd.Id
```

Typically you also create a network security group to filter network traffic to the VM and a load balancer to distribute traffic across multiple VMs.

Create the virtual machine

Now start to build your VM configuration. Each VM size has a limit for the total number of NICs that you can add to a VM. For more information, see Windows VM sizes.

1. Set your VM credentials to the `$cred` variable as follows:

   PowerShell

   ```
   $cred = Get-Credential
   ```

2. Define your VM with New-AzVMConfig. The following example defines a VM named *myVM* and uses a VM size that supports more than two NICs (*Standard_DS3_v2*):

   PowerShell

   ```
   $vmConfig = New-AzVMConfig -VMName "myVM" -VMSize "Standard_DS3_v2"
   ```

3. Create the rest of your VM configuration with Set-AzVMOperatingSystem and Set-AzVMSourceImage. The following example creates a Windows Server 2016 VM:

   PowerShell

   ```
   $vmConfig = Set-AzVMOperatingSystem -VM $vmConfig `
       -Windows `
       -ComputerName "myVM" `
       -Credential $cred `
       -ProvisionVMAgent `
       -EnableAutoUpdate
   $vmConfig = Set-AzVMSourceImage -VM $vmConfig `
       -PublisherName "MicrosoftWindowsServer" `
       -Offer "WindowsServer" `
       -Skus "2016-Datacenter" `
       -Version "latest"
   ```

4. Attach the two NICs that you previously created with Add-AzVMNetworkInterface:

   PowerShell

   ```
   $vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic1.Id -
   Primary
   ```

```
$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic2.Id
```

5. Create your VM with New-AzVM:

```
PowerShell

New-AzVM -VM $vmConfig -ResourceGroupName "myResourceGroup" -Location
"EastUs"
```

6. Add routes for secondary NICs to the OS by completing the steps in Configure the operating system for multiple NICs.

Add a NIC to an existing VM

To add a virtual NIC to an existing VM, you deallocate the VM, add the virtual NIC, then start the VM. Different VM sizes support a varying number of NICs, so size your VM accordingly. If needed, you can resize a VM.

1. Deallocate the VM with Stop-AzVM. The following example deallocates the VM named *myVM* in *myResourceGroup*:

```
PowerShell

Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

2. Get the existing configuration of the VM with Get-AzVm. The following example gets information for the VM named *myVM* in *myResourceGroup*:

```
PowerShell

$vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
```

3. The following example creates a virtual NIC with New-AzNetworkInterface named *myNic3* that is attached to *mySubnetBackEnd*. The virtual NIC is then attached to the VM named *myVM* in *myResourceGroup* with Add-AzVMNetworkInterface:

```
PowerShell

# Get info for the back end subnet
$myVnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName
"myResourceGroup"
$backEnd = $myVnet.Subnets|?{$_.Name -eq 'mySubnetBackEnd'}

# Create a virtual NIC
```

```
$myNic3 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
    -Name "myNic3" `
    -Location "EastUs" `
    -SubnetId $backEnd.Id

# Get the ID of the new virtual NIC and add to VM
$nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -
Name "MyNic3").Id
Add-AzVMNetworkInterface -VM $vm -Id $nicId | Update-AzVm -
ResourceGroupName "myResourceGroup"
```

Primary virtual NICs

One of the NICs on a multi-NIC VM needs to be primary. If one of the existing virtual NICs on the VM is already set as primary, you can skip this step. The following example assumes that two virtual NICs are now present on a VM and you wish to add the first NIC (`[0]`) as the primary:

PowerShell

```
# List existing NICs on the VM and find which one is primary
$vm.NetworkProfile.NetworkInterfaces

# Set NIC 0 to be primary
$vm.NetworkProfile.NetworkInterfaces[0].Primary = $true
$vm.NetworkProfile.NetworkInterfaces[1].Primary = $false

# Update the VM state in Azure
Update-AzVM -VM $vm -ResourceGroupName "myResourceGroup"
```

4. Start the VM with Start-AzVm:

PowerShell

```
Start-AzVM -ResourceGroupName "myResourceGroup" -Name "myVM"
```

5. Add routes for secondary NICs to the OS by completing the steps in Configure the operating system for multiple NICs.

Remove a NIC from an existing VM

To remove a virtual NIC from an existing VM, you deallocate the VM, remove the virtual NIC, then start the VM.

1. Deallocate the VM with Stop-AzVM. The following example deallocates the VM named *myVM* in *myResourceGroup*:

   PowerShell

   ```
   Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
   ```

2. Get the existing configuration of the VM with Get-AzVm. The following example gets information for the VM named *myVM* in *myResourceGroup*:

   PowerShell

   ```
   $vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
   ```

3. Get information about the NIC remove with Get-AzNetworkInterface. The following example gets information about *myNic3*:

   PowerShell

   ```
   # List existing NICs on the VM if you need to determine NIC name
   $vm.NetworkProfile.NetworkInterfaces

   $nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -
   Name "myNic3").Id
   ```

4. Remove the NIC with Remove-AzVMNetworkInterface and then update the VM with Update-AzVm. The following example removes *myNic3* as obtained by `$nicId` in the preceding step:

   PowerShell

   ```
   Remove-AzVMNetworkInterface -VM $vm -NetworkInterfaceIDs $nicId | `
       Update-AzVm -ResourceGroupName "myResourceGroup"
   ```

5. Start the VM with Start-AzVm:

   PowerShell

   ```
   Start-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
   ```

Create multiple NICs with templates

Azure Resource Manager templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. Resource Manager templates use declarative JSON files to define your environment. For more information, see overview of Azure Resource Manager. You can use *copy* to specify the number of instances to create:

```JSON
"copy": {
    "name": "multiplenics",
    "count": "[parameters('count')]"
}
```

For more information, see creating multiple instances by using *copy*.

You can also use `copyIndex()` to append a number to a resource name. You can then create *myNic1*, *MyNic2* and so on. The following code shows an example of appending the index value:

```JSON
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of creating multiple NICs by using Resource Manager templates.

Add routes for secondary NICs to the OS by completing the steps in Configure the operating system for multiple NICs.

Configure guest OS for multiple NICs

Azure assigns a default gateway to the first (primary) network interface attached to the virtual machine. Azure does not assign a default gateway to additional (secondary) network interfaces attached to a virtual machine. Therefore, you are unable to communicate with resources outside the subnet that a secondary network interface is in, by default. Secondary network interfaces can, however, communicate with resources outside their subnet, though the steps to enable communication are different for different operating systems.

1. From a Windows command prompt, run the `route print` command, which returns output similar to the following output for a virtual machine with two attached network interfaces:

```
===========================================================================
====
Interface List
3...00 0d 3a 10 92 ce ......Microsoft Hyper-V Network Adapter #3
7...00 0d 3a 10 9b 2a ......Microsoft Hyper-V Network Adapter #4
===========================================================================
====
```

In this example, **Microsoft Hyper-V Network Adapter #4** (interface 7) is the secondary network interface that doesn't have a default gateway assigned to it.

2. From a command prompt, run the `ipconfig` command to see which IP address is assigned to the secondary network interface. In this example, 192.168.2.4 is assigned to interface 7. No default gateway address is returned for the secondary network interface.

3. To route all traffic destined for addresses outside the subnet of the secondary network interface to the gateway for the subnet, run the following command:

```
route add -p 0.0.0.0 MASK 0.0.0.0 192.168.2.1 METRIC 5015 IF 7
```

The gateway address for the subnet is the first IP address (ending in .1) in the address range defined for the subnet. If you don't want to route all traffic outside the subnet, you could add individual routes to specific destinations, instead. For example, if you only wanted to route traffic from the secondary network interface to the 192.168.3.0 network, you enter the command:

```
route add -p 192.168.3.0 MASK 255.255.255.0 192.168.2.1 METRIC 5015 IF
7
```

4. To confirm successful communication with a resource on the 192.168.3.0 network, for example, enter the following command to ping 192.168.3.4 using interface 7 (192.168.2.4):

```
ping 192.168.3.4 -S 192.168.2.4
```

You may need to open ICMP through the Windows firewall of the device you're pinging with the following command:

```
netsh advfirewall firewall add rule name=Allow-ping protocol=icmpv4
dir=in action=allow
```

5. To confirm the added route is in the route table, enter the `route print` command, which returns output similar to the following text:

```
===========================================================================
====
Active Routes:
Network Destination        Netmask          Gateway       Interface
Metric
          0.0.0.0          0.0.0.0      192.168.1.1      192.168.1.4
15
          0.0.0.0          0.0.0.0      192.168.2.1      192.168.2.4
5015
```

The route listed with *192.168.1.1* under **Gateway**, is the route that is there by default for the primary network interface. The route with *192.168.2.1* under **Gateway**, is the route you added.

Next steps

Review Windows VM sizes when you're trying to create a VM that has multiple NICs. Pay attention to the maximum number of NICs that each VM size supports.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to create a Linux virtual machine in Azure with multiple network interface cards

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article details how to create a VM with multiple NICs with the Azure CLI.

Create supporting resources

Install the latest Azure CLI and log in to an Azure account using az login.

In the following examples, replace example parameter names with your own values. Example parameter names included *myResourceGroup*, *mystorageaccount*, and *myVM*.

First, create a resource group with az group create. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Create the virtual network with az network vnet create. The following example creates a virtual network named *myVnet* and subnet named *mySubnetFrontEnd*:

Azure CLI

```
az network vnet create \
    --resource-group myResourceGroup \
    --name myVnet \
    --address-prefix 10.0.0.0/16 \
    --subnet-name mySubnetFrontEnd \
    --subnet-prefix 10.0.1.0/24
```

Create a subnet for the back-end traffic with az network vnet subnet create. The following example creates a subnet named *mySubnetBackEnd*:

Azure CLI

```
az network vnet subnet create \
    --resource-group myResourceGroup \
```

```
    --vnet-name myVnet \
    --name mySubnetBackEnd \
    --address-prefix 10.0.2.0/24
```

Create a network security group with az network nsg create. The following example creates a network security group named *myNetworkSecurityGroup*:

Azure CLI

```
az network nsg create \
    --resource-group myResourceGroup \
    --name myNetworkSecurityGroup
```

Create and configure multiple NICs

Create two NICs with az network nic create. The following example creates two NICs, named *myNic1* and *myNic2*, connected the network security group, with one NIC connecting to each subnet:

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic1 \
    --vnet-name myVnet \
    --subnet mySubnetFrontEnd \
    --network-security-group myNetworkSecurityGroup
az network nic create \
    --resource-group myResourceGroup \
    --name myNic2 \
    --vnet-name myVnet \
    --subnet mySubnetBackEnd \
    --network-security-group myNetworkSecurityGroup
```

Create a VM and attach the NICs

When you create the VM, specify the NICs you created with `--nics`. You also need to take care when you select the VM size. There are limits for the total number of NICs that you can add to a VM. Read more about Linux VM sizes.

Create a VM with az vm create. The following example creates a VM named *myVM*:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image Ubuntu2204 \
    --size Standard_DS3_v2 \
    --admin-username azureuser \
    --generate-ssh-keys \
    --nics myNic1 myNic2
```

Add routing tables to the guest OS by completing the steps in Configure the guest OS for multiple NICs.

Add a NIC to a VM

The previous steps created a VM with multiple NICs. You can also add NICs to an existing VM with the Azure CLI. Different VM sizes support a varying number of NICs, so size your VM accordingly. If needed, you can resize a VM.

Create another NIC with az network nic create. The following example creates a NIC named *myNic3* connected to the back-end subnet and network security group created in the previous steps:

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic3 \
    --vnet-name myVnet \
    --subnet mySubnetBackEnd \
    --network-security-group myNetworkSecurityGroup
```

To add a NIC to an existing VM, first deallocate the VM with az vm deallocate. The following example deallocates the VM named *myVM*:

Azure CLI

```
az vm deallocate --resource-group myResourceGroup --name myVM
```

Add the NIC with az vm nic add. The following example adds *myNic3* to *myVM*:

Azure CLI

```
az vm nic add \
    --resource-group myResourceGroup \
```

```
    --vm-name myVM \
    --nics myNic3
```

Start the VM with az vm start:

Azure CLI

```
az vm start --resource-group myResourceGroup --name myVM
```

Add routing tables to the guest OS by completing the steps in Configure the guest OS for multiple NICs.

Remove a NIC from a VM

To remove a NIC from an existing VM, first deallocate the VM with az vm deallocate. The following example deallocates the VM named *myVM*:

Azure CLI

```
az vm deallocate --resource-group myResourceGroup --name myVM
```

Remove the NIC with az vm nic remove. The following example removes *myNic3* from *myVM*:

Azure CLI

```
az vm nic remove \
    --resource-group myResourceGroup \
    --vm-name myVM \
    --nics myNic3
```

Start the VM with az vm start:

Azure CLI

```
az vm start --resource-group myResourceGroup --name myVM
```

Create multiple NICs using Resource Manager templates

Azure Resource Manager templates use declarative JSON files to define your environment. You can read an overview of Azure Resource Manager. Resource Manager

templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. You use *copy* to specify the number of instances to create:

JSON

```
"copy": {
    "name": "multiplenics"
    "count": "[parameters('count')]"
}
```

Read more about creating multiple instances using *copy*.

You can also use a `copyIndex()` to then append a number to a resource name, which allows you to create `myNic1`, `myNic2`, etc. The following shows an example of appending the index value:

JSON

```
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of creating multiple NICs using Resource Manager templates.

Add routing tables to the guest OS by completing the steps in Configure the guest OS for multiple NICs.

Configure guest OS for multiple NICs

The previous steps created a virtual network and subnet, attached NICs, then created a VM. A public IP address and network security group rules that allow SSH traffic were not created. To configure the guest OS for multiple NICs, you need to allow remote connections and run commands locally on the VM.

To allow SSH traffic, create a network security group rule with az network nsg rule create as follows:

Azure CLI

```
az network nsg rule create \
    --resource-group myResourceGroup \
    --nsg-name myNetworkSecurityGroup \
    --name allow_ssh \
```

```
    --priority 101 \
    --destination-port-ranges 22
```

Create a public IP address with az network public-ip create and assign it to the first NIC with az network nic ip-config update:

Azure CLI

```
az network public-ip create --resource-group myResourceGroup --name
myPublicIP

az network nic ip-config update \
    --resource-group myResourceGroup \
    --nic-name myNic1 \
    --name ipconfig1 \
    --public-ip myPublicIP
```

To view the public IP address of the VM, use az vm show as follows::

Azure CLI

```
az vm show --resource-group myResourceGroup --name myVM -d --query publicIps
-o tsv
```

Now SSH to the public IP address of your VM. The default username provided in a previous step was *azureuser*. Provide your own username and public IP address:

Bash

```
ssh azureuser@137.117.58.232
```

To send to or from a secondary network interface, you have to manually add persistent routes to the operating system for each secondary network interface. In this article, *eth1* is the secondary interface. Instructions for adding persistent routes to the operating system vary by distro. See documentation for your distro for instructions.

When adding the route to the operating system, the gateway address is the first address of the subnet the network interface is in. For example, if the subnet has been assigned the range *10.0.2.0/24*, the gateway you specify for the route is *10.0.2.1* or if the subnet has been assigned the range *10.0.2.128/25*, the gateway you specify for the route is *10.0.2.129*. You can define a specific network for the route's destination, or specify a destination of *0.0.0.0*, if you want all traffic for the interface to go through the specified gateway. The gateway for each subnet is managed by the virtual network.

Once you've added the route for a secondary interface, verify that the route is in your route table with `route -n`. The following example output is for the route table that has the two network interfaces added to the VM in this article:

```
Bash

Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
0.0.0.0         10.0.1.1        0.0.0.0         UG    0      0        0 eth0
0.0.0.0         10.0.2.1        0.0.0.0         UG    0      0        0 eth1
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
168.63.129.16   10.0.1.1        255.255.255.255 UGH   0      0        0 eth0
169.254.169.254 10.0.1.1        255.255.255.255 UGH   0      0        0 eth0
```

Confirm that the route you added persists across reboots by checking your route table again after a reboot. To test connectivity, you can enter the following command, for example, where *eth1* is the name of a secondary network interface:

```
Bash

ping bing.com -c 4 -I eth1
```

Next steps

Review Linux VM sizes when trying to creating a VM with multiple NICs. Pay attention to the maximum number of NICs each VM size supports.

To further secure your VMs, use just in time VM access. This feature opens network security group rules for SSH traffic when needed, and for a defined period of time. For more information, see Manage virtual machine access using just in time.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create an Azure Virtual Machine with Accelerated Networking

Deploy to Azure

This article describes how to create a Linux or Windows virtual machine (VM) with Accelerated Networking (AccelNet) enabled by using the Azure CLI command-line interface.

Prerequisites

Portal

- An Azure account with an active subscription. You can create an account for free.

Create a resource group and virtual network

Portal

Create a resource group

1. Sign in to the Azure portal.
2. In the search box at the top of the portal, enter **Resource group**. Select **Resource groups** in the search results.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter or select the following information:

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Enter **test-rg**.
Region	Select **East US 2**.

5. Select **Review + create**.

6. Select **Create**.

Create a virtual network

1. In the search box at the top of the portal, enter **Virtual network**. Select **Virtual networks** in the search results.

2. Select **+ Create**.

3. On the **Basics** tab of **Create virtual network**, enter or select the following information:

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Name	Enter **vnet-1**.
Region	Select **East US 2**.

4. Select **Next** to proceed to the **Security** tab.

5. Select **Next** to proceed to the **IP Addresses** tab.

6. In the address space box in **Subnets**, select the **default** subnet.

7. In **Edit subnet**, enter or select the following information:

Expand table

Setting	Value
Name	Enter **subnet-1**.
Starting address	Enter **10.0.0.0**.
Subnet size	Leave the default of **/24 (256 addresses)**.

8. Select **Save**.

9. Select **Review + create** at the bottom of the screen, and when validation passes, select **Create**.

Create Azure Bastion

Azure Bastion uses your browser to connect to VMs in your virtual network over secure shell (SSH) or remote desktop protocol (RDP) by using their private IP addresses. The VMs don't need public IP addresses, client software, or special configuration. For more information about Azure Bastion, see Azure Bastion.

> **Note**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

1. In the search box at the top of the portal, enter **Bastion**. Select **Bastions** in the search results.
2. Select **+ Create**.
3. In the **Basics** tab of **Create a Bastion**, enter or select the following information:

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Name	Enter **bastion**.
Region	Select **East US 2**.
Tier	Select **Developer**.
Virtual network	Select **vnet-1**.

4. Select **Review + create**.
5. Select **Create**.

Create a network interface with Accelerated Networking

Portal

Accelerated networking is enabled in the portal during virtual machine creation. Create a virtual machine in the following section.

> **Note**
>
> - The Accelerated Networking setting in the portal shows the user-selected state. Accelerated Networking allows choosing Disabled in the portal even if the VM size requires Accelerated Networking. VM sizes that require Accelerated Networking enable Accelerated Networking at runtime regardless of the user setting in the portal. Accelerated Networking is a required feature for general purpose VM sizes of v5 or higher.

Create a VM and attach the NIC

Portal

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **+ Create** then **Azure virtual machine**.
3. In **Create a virtual machine** enter or select the following information in the **Basics** tab:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.

Setting	Value
Instance details	
Virtual machine name	Enter **vm-1**.
Region	Select **(US) East US 2**.
Availability options	Select **No infrastructure redundancy required**.
Security type	Select **Standard**.
Image	Select **Ubuntu Server 24.04 LTS - x64 Gen2**.
VM architecture	Leave the default of **x64**.
Size	Select a size.
Administrator account	
Authentication type	Select **SSH public key**.
Username	Enter a username.
SSH public key source	Select **Generate new key pair**.
Key pair name	Enter **vm-1-key**.
Inbound port rules	
Public inbound ports	Select **None**.

4. Select **Next: Disks** then **Next: Networking**.

5. In the Networking tab, enter or select the following information:

Expand table

Setting	Value
Network interface	
Virtual network	Select **vnet-1**.
Subnet	Select **subnet-1 (10.0.0.0/24)**.
Public IP	Select **None**.
Network interface (NIC) network security group	Select **Advanced**.
Configure network security group	Select **Create new**. In **Name** enter **nsg-1**.

Setting	Value
	Select **OK**.

6. Leave the rest of the options at the defaults and select **Review + create**.
7. Select **Create**.
8. When the **Generate new key pair** pop-up appears, select **Download private key and create resource**. The key file downloads as **vm-1-key.pem**. Make note of the download location for the `.pem` file. The private key is needed in later steps for connecting to the virtual machine with Azure Bastion.

Confirm that accelerated networking is enabled

Linux

1. In the Azure portal, search for and select *virtual machines*.
2. On the **Virtual machines** page, select your new VM.
3. On the VM's **Overview** page, select **Connect** then **Connect via Bastion**.
4. In the Bastion connection screen, change **Authentication Type** to **SSH Private Key from Local File**.
5. Enter the **Username** that you used when creating the virtual machine. In this example, the user is named **azureuser**, replace with the username you created.
6. In **Local File**, select the folder icon and browse to the private key file that was generated when you created the VM. The private key file is typically named `id_rsa` or `id_rsa.pem`.
7. Select **Connect**.
8. A new browser window opens with the Bastion connection to your VM.
9. From a shell on the remote VM, enter `uname -r` and confirm that the kernel version is one of the following versions, or greater:

   - **Ubuntu 16.04**: 4.11.0-1013.
   - **SLES SP3**: 4.4.92-6.18.
   - **RHEL**: 3.10.0-693, 2.6.32-573. RHEL 6.7-6.10 are supported if the Mellanox VF version 4.5+ is installed before Linux Integration Services 4.3+.

> **Note**
>
> Other kernel versions might be supported. For an updated list, see the compatibility tables for each distribution at Supported Linux and FreeBSD virtual machines for Hyper-V, and confirm that SR-IOV is supported. You can find more details in the release notes for Linux Integration Services for Hyper-V and Azure. *

10. Use the `lspci` command to confirm that the Mellanox VF device is exposed to the VM. The returned output should be similar to the following example:

    Output

    ```
    0000:00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host
    bridge (AGP disabled) (rev 03)
    0000:00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 01)
    0000:00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
    0000:00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 02)
    0000:00:08.0 VGA compatible controller: Microsoft Corporation Hyper-V virtual
    VGA
    0001:00:02.0 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
    [ConnectX-3/ConnectX-3 Pro Virtual Function]
    ```

11. Use the `ethtool -S eth0 | grep vf_` command to check for activity on the virtual function (VF). If accelerated networking is enabled and active, you receive output similar to the following example:

    Output

    ```
    vf_rx_packets: 992956
    vf_rx_bytes: 2749784180
    vf_tx_packets: 2656684
    vf_tx_bytes: 1099443970
    vf_tx_dropped: 0
    ```

12. Close the Bastion connection to the VM.

Windows

Once you create the VM in Azure, connect to the VM and confirm that the Ethernet controller is installed in Windows.

1. In the Azure portal, search for and select *virtual machines*.

2. On the **Virtual machines** page, select your new VM.

3. On the VM's **Overview** page, select **Connect** then **Connect via Bastion**.

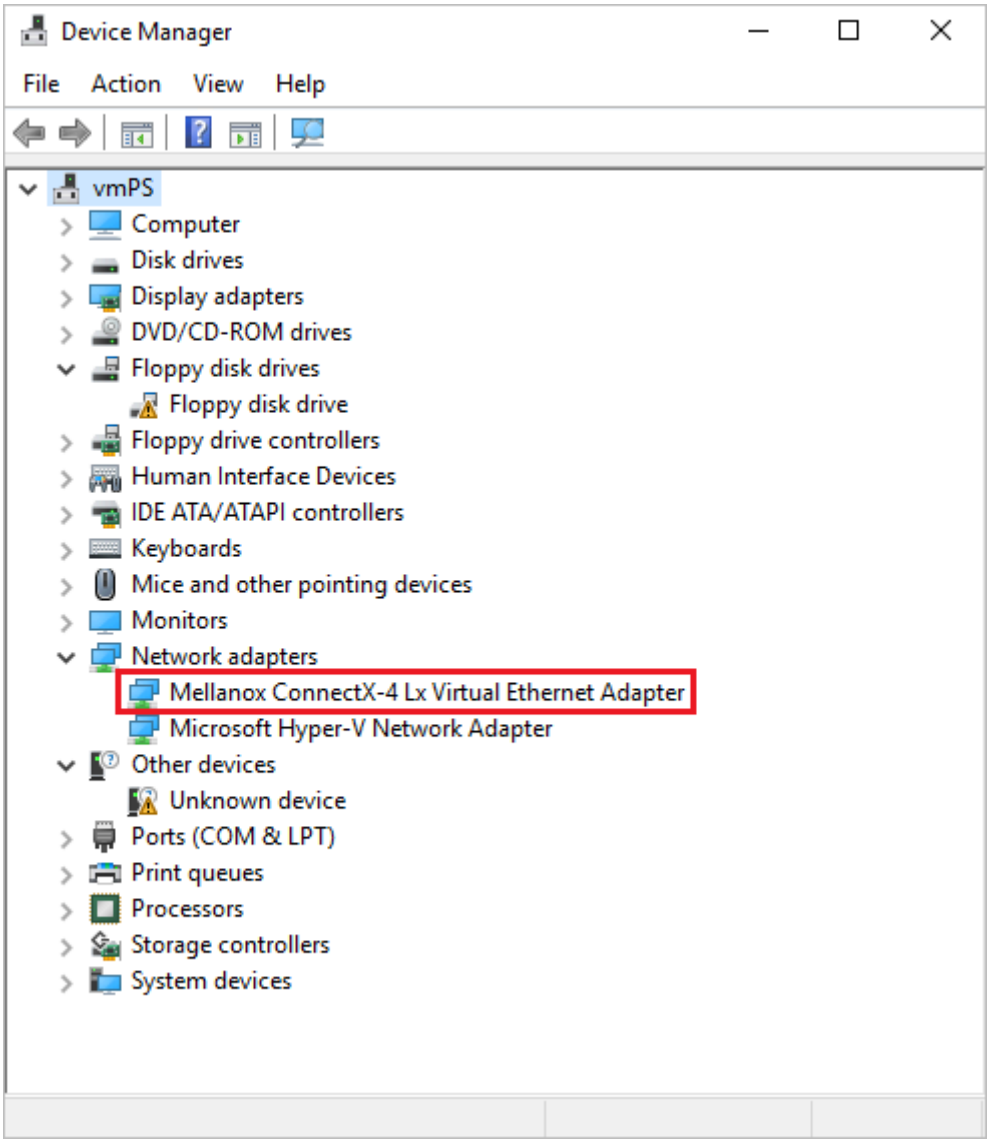
4. Enter the credentials you used when you created the VM, and then select **Connect**.

5. A new browser window opens with the Bastion connection to your VM.

6. On the remote VM, right-click **Start** and select **Device Manager**.

7. In the **Device Manager** window, expand the **Network adapters** node.

8. Confirm that the **Mellanox ConnectX-4 Lx Virtual Ethernet Adapter** appears, as shown in the following image:



The presence of the adapter confirms that Accelerated Networking is enabled for your VM.

9. Verify the packets are flowing over the VF interface from the output of the following command:

```
PowerShell
```

```
PS C:\ > Get-NetAdapter | Where-Object InterfaceDescription -like
"*Mellanox*Virtual*" | Get-NetAdapterStatistics

Name                            ReceivedBytes ReceivedUnicastPackets
SentBytes SentUnicastPackets
----                            ------------- ----------------------       --
------- ------------------
Ethernet 2                           492447549                 347643
7468446               34991
```

> **Note**
>
> If the Mellanox adapter fails to start, open an administrator command prompt on the remote VM and enter the following command:
>
> `netsh int tcp set global rss = enabled`

10. Close the Bastion connection to the VM.

Next steps

- How Accelerated Networking works in Linux and FreeBSD VMs
- Proximity placement groups

Last updated on 02/20/2026

Create an Azure Virtual Machine with Accelerated Networking

Deploy to Azure

This article describes how to create a Linux or Windows virtual machine (VM) with Accelerated Networking (AccelNet) enabled by using the Azure CLI command-line interface.

Prerequisites

Portal

- An Azure account with an active subscription. You can create an account for free.

Create a resource group and virtual network

Portal

Create a resource group

1. Sign in to the Azure portal.
2. In the search box at the top of the portal, enter **Resource group**. Select **Resource groups** in the search results.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter or select the following information:

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Enter **test-rg**.
Region	Select **East US 2**.

5. Select **Review + create**.

6. Select **Create**.

Create a virtual network

1. In the search box at the top of the portal, enter **Virtual network**. Select **Virtual networks** in the search results.

2. Select **+ Create**.

3. On the **Basics** tab of **Create virtual network**, enter or select the following information:

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Name	Enter **vnet-1**.
Region	Select **East US 2**.

4. Select **Next** to proceed to the **Security** tab.

5. Select **Next** to proceed to the **IP Addresses** tab.

6. In the address space box in **Subnets**, select the **default** subnet.

7. In **Edit subnet**, enter or select the following information:

Expand table

Setting	Value
Name	Enter **subnet-1**.
Starting address	Enter **10.0.0.0**.
Subnet size	Leave the default of **/24 (256 addresses)**.

8. Select **Save**.

9. Select **Review + create** at the bottom of the screen, and when validation passes, select **Create**.

Create Azure Bastion

Azure Bastion uses your browser to connect to VMs in your virtual network over secure shell (SSH) or remote desktop protocol (RDP) by using their private IP addresses. The VMs don't need public IP addresses, client software, or special configuration. For more information about Azure Bastion, see Azure Bastion.

> **Note**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

1. In the search box at the top of the portal, enter **Bastion**. Select **Bastions** in the search results.
2. Select **+ Create**.
3. In the **Basics** tab of **Create a Bastion**, enter or select the following information:

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Select **test-rg**.
Name	Enter **bastion**.
Region	Select **East US 2**.
Tier	Select **Developer**.
Virtual network	Select **vnet-1**.

4. Select **Review + create**.
5. Select **Create**.

Create a network interface with Accelerated Networking

Portal

Accelerated networking is enabled in the portal during virtual machine creation. Create a virtual machine in the following section.

> **Note**
>
> - The Accelerated Networking setting in the portal shows the user-selected state. Accelerated Networking allows choosing Disabled in the portal even if the VM size requires Accelerated Networking. VM sizes that require Accelerated Networking enable Accelerated Networking at runtime regardless of the user setting in the portal. Accelerated Networking is a required feature for general purpose VM sizes of v5 or higher.

Create a VM and attach the NIC

Portal

1. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines** in the search results.
2. Select **+ Create** then **Azure virtual machine**.
3. In **Create a virtual machine** enter or select the following information in the **Basics** tab:

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg**.

Setting	Value
Instance details	
Virtual machine name	Enter **vm-1**.
Region	Select **(US) East US 2**.
Availability options	Select **No infrastructure redundancy required**.
Security type	Select **Standard**.
Image	Select **Ubuntu Server 24.04 LTS - x64 Gen2**.
VM architecture	Leave the default of **x64**.
Size	Select a size.
Administrator account	
Authentication type	Select **SSH public key**.
Username	Enter a username.
SSH public key source	Select **Generate new key pair**.
Key pair name	Enter **vm-1-key**.
Inbound port rules	
Public inbound ports	Select **None**.

4. Select **Next: Disks** then **Next: Networking**.

5. In the Networking tab, enter or select the following information:

Expand table

Setting	Value
Network interface	
Virtual network	Select **vnet-1**.
Subnet	Select **subnet-1 (10.0.0.0/24)**.
Public IP	Select **None**.
Network interface (NIC) network security group	Select **Advanced**.
Configure network security group	Select **Create new**. In **Name** enter **nsg-1**.

Setting	Value
	Select **OK**.

6. Leave the rest of the options at the defaults and select **Review + create**.
7. Select **Create**.
8. When the **Generate new key pair** pop-up appears, select **Download private key and create resource**. The key file downloads as **vm-1-key.pem**. Make note of the download location for the `.pem` file. The private key is needed in later steps for connecting to the virtual machine with Azure Bastion.

Confirm that accelerated networking is enabled

Linux

1. In the Azure portal, search for and select *virtual machines*.
2. On the **Virtual machines** page, select your new VM.
3. On the VM's **Overview** page, select **Connect** then **Connect via Bastion**.
4. In the Bastion connection screen, change **Authentication Type** to **SSH Private Key from Local File**.
5. Enter the **Username** that you used when creating the virtual machine. In this example, the user is named **azureuser**, replace with the username you created.
6. In **Local File**, select the folder icon and browse to the private key file that was generated when you created the VM. The private key file is typically named `id_rsa` or `id_rsa.pem`.
7. Select **Connect**.
8. A new browser window opens with the Bastion connection to your VM.
9. From a shell on the remote VM, enter `uname -r` and confirm that the kernel version is one of the following versions, or greater:

   - **Ubuntu 16.04**: 4.11.0-1013.
   - **SLES SP3**: 4.4.92-6.18.
   - **RHEL**: 3.10.0-693, 2.6.32-573. RHEL 6.7-6.10 are supported if the Mellanox VF version 4.5+ is installed before Linux Integration Services 4.3+.

> **Note**
>
> Other kernel versions might be supported. For an updated list, see the compatibility tables for each distribution at Supported Linux and FreeBSD virtual machines for Hyper-V, and confirm that SR-IOV is supported. You can find more details in the release notes for Linux Integration Services for Hyper-V and Azure. *

10. Use the `lspci` command to confirm that the Mellanox VF device is exposed to the VM. The returned output should be similar to the following example:

    Output

    ```
    0000:00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host
    bridge (AGP disabled) (rev 03)
    0000:00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 01)
    0000:00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
    0000:00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 02)
    0000:00:08.0 VGA compatible controller: Microsoft Corporation Hyper-V virtual
    VGA
    0001:00:02.0 Ethernet controller: Mellanox Technologies MT27500/MT27520 Family
    [ConnectX-3/ConnectX-3 Pro Virtual Function]
    ```

11. Use the `ethtool -S eth0 | grep vf_` command to check for activity on the virtual function (VF). If accelerated networking is enabled and active, you receive output similar to the following example:

    Output

    ```
    vf_rx_packets: 992956
    vf_rx_bytes: 2749784180
    vf_tx_packets: 2656684
    vf_tx_bytes: 1099443970
    vf_tx_dropped: 0
    ```

12. Close the Bastion connection to the VM.

Windows

Once you create the VM in Azure, connect to the VM and confirm that the Ethernet controller is installed in Windows.

1. In the Azure portal, search for and select *virtual machines*.

2. On the **Virtual machines** page, select your new VM.

3. On the VM's **Overview** page, select **Connect** then **Connect via Bastion**.

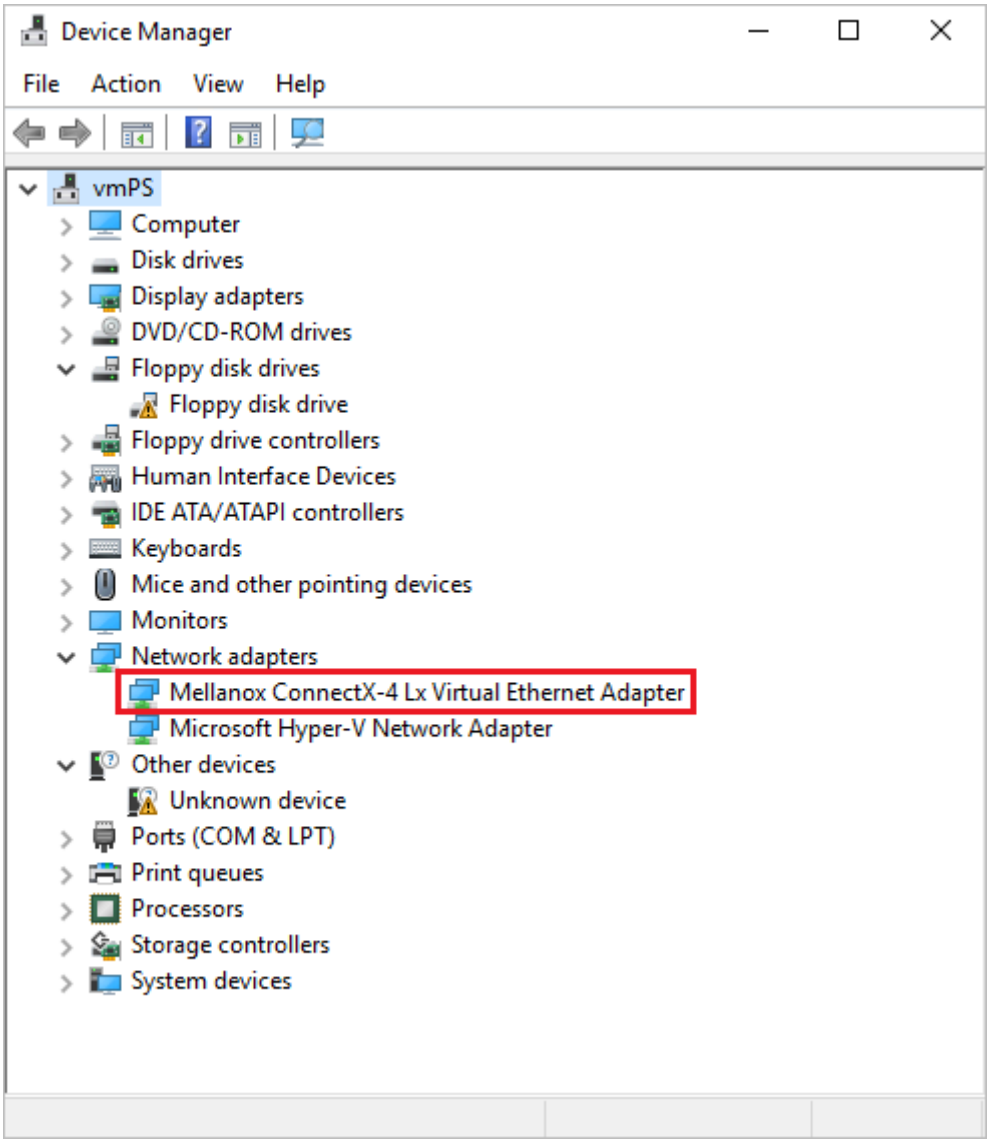
4. Enter the credentials you used when you created the VM, and then select **Connect**.

5. A new browser window opens with the Bastion connection to your VM.

6. On the remote VM, right-click **Start** and select **Device Manager**.

7. In the **Device Manager** window, expand the **Network adapters** node.

8. Confirm that the **Mellanox ConnectX-4 Lx Virtual Ethernet Adapter** appears, as shown in the following image:



The presence of the adapter confirms that Accelerated Networking is enabled for your VM.

9. Verify the packets are flowing over the VF interface from the output of the following command:

```
PowerShell
```

```
PS C:\ > Get-NetAdapter | Where-Object InterfaceDescription -like
"*Mellanox*Virtual*" | Get-NetAdapterStatistics

Name                            ReceivedBytes ReceivedUnicastPackets
SentBytes SentUnicastPackets
----                            ------------- ----------------------       --
------- ------------------
Ethernet 2                           492447549                 347643
7468446               34991
```

> **Note**
>
> If the Mellanox adapter fails to start, open an administrator command prompt on the remote VM and enter the following command:
>
> `netsh int tcp set global rss = enabled`

10. Close the Bastion connection to the VM.

Next steps

- How Accelerated Networking works in Linux and FreeBSD VMs
- Proximity placement groups

Last updated on 02/20/2026

Set up DPDK in a Linux virtual machine

Article • 10/02/2024

Data Plane Development Kit (DPDK) on Azure offers a faster user-space packet processing framework for performance-intensive applications. This framework bypasses the virtual machine's kernel network stack.

In typical packet processing that uses the kernel network stack, the process is interrupt-driven. When the network interface receives incoming packets, there's a kernel interrupt to process the packet and a context switch from the kernel space to the user space. DPDK eliminates context switching and the interrupt-driven method in favor of a user-space implementation that uses poll mode drivers for fast packet processing.

DPDK consists of sets of user-space libraries that provide access to lower-level resources. These resources can include hardware, logical cores, memory management, and poll mode drivers for network interface cards.

DPDK can run on Azure virtual machines that are supporting multiple operating system distributions. DPDK provides key performance differentiation in driving network function virtualization implementations. These implementations can take the form of network virtual appliances (NVAs), such as virtual routers, firewalls, VPNs, load balancers, evolved packet cores, and denial-of-service (DDoS) applications.

A list of setup instructions for DPDK on MANA VMs is available here: Microsoft Azure Network Adapter (MANA) and DPDK on Linux

Benefit

Higher packets per second (PPS): Bypassing the kernel and taking control of packets in the user space reduces the cycle count by eliminating context switches. It also improves the rate of packets that are processed per second in Azure Linux virtual machines.

Supported operating systems minimum versions

The following distributions from the Azure Marketplace are supported:

Expand table

Linux OS	Kernel version
Ubuntu 18.04	4.15.0-1014-azure+
SLES 15 SP1	4.12.14-8.19-azure+
RHEL 7.5	3.10.0-862.11.6.el7.x86_64+
Debian 10	4.19.0-1-cloud+

The noted versions are the minimum requirements. Newer versions are supported too.

A list of requirements for DPDK on MANA VMs is available here: Microsoft Azure Network Adapter (MANA) and DPDK on Linux

Custom kernel support

For any Linux kernel version that's not listed, see Patches for building an Azure-tuned Linux kernel. For more information, you can also contact aznetdpdk@microsoft.com.

Region support

All Azure regions support DPDK.

Prerequisites

Accelerated networking must be enabled on a Linux virtual machine. The virtual machine should have at least two network interfaces, with one interface for management. Enabling Accelerated networking on management interface isn't recommended. Learn how to create a Linux virtual machine with accelerated networking enabled.

In addition, DPDK uses RDMA verbs to create data queues on the Network Adapter. In the VM, ensure the correct RDMA kernel drivers are loaded. They can be mlx4_ib, mlx5_ib or mana_ib depending on VM sizes.

Install DPDK manually (recommended)

DPDK installation instructions for MANA VMs are available here: Microsoft Azure Network Adapter (MANA) and DPDK on Linux

Install build dependencies

RHEL

RHEL7.5

Bash

```
yum -y groupinstall "Infiniband Support"
sudo dracut --add-drivers "mlx4_en mlx4_ib mlx5_ib" -f
yum install -y gcc kernel-devel-`uname -r` numactl-devel.x86_64
librdmacm-devel libmnl-devel meson
```

Compile and install DPDK manually

1. Download the latest DPDK. Version 22.11 LTS or newer is recommended for Azure.
2. Build the default config with `meson builddir`.
3. Compile with `ninja -C builddir`.
4. Install with `DESTDIR=<output folder> ninja -C builddir install`.

Configure the runtime environment

After restarting, run the following commands once:

1. Hugepages

    - Configure hugepage by running the following command, once for each numa node:

    Bash

    ```
     echo 1024 | sudo tee
    /sys/devices/system/node/node*/hugepages/hugepages-2048kB/nr_hugepages
    ```

    - Create a directory for mounting with `mkdir /mnt/huge`.
    - Mount hugepages with `mount -t hugetlbfs nodev /mnt/huge`.
    - Check that hugepages are reserved with `grep Huge /proc/meminfo`.

- The example above is for 2M huge pages. 1G huge pages can also be used.

> ⓘ **Note**
>
> There is a way to modify the grub file so that hugepages are reserved on boot by following the instructions for the DPDK. The instructions are at the bottom of the page. When you're using an Azure Linux virtual machine, modify files under **/etc/config/grub.d** instead, to reserve hugepages across reboots.

2. MAC & IP addresses: Use `ifconfig -a` to view the MAC and IP address of the network interfaces. The *VF* network interface and *NETVSC* network interface have the same MAC address, but only the *NETVSC* network interface has an IP address. *VF* interfaces are running as subordinate interfaces of *NETVSC* interfaces.

3. PCI addresses

   - Use `ethtool -i <vf interface name>` to find out which PCI address to use for *VF*.

   - If *eth0* has accelerated networking enabled, make sure that testpmd doesn't accidentally take over the *VF* pci device for *eth0*. If the DPDK application accidentally takes over the management network interface and causes you to lose your SSH connection, use the serial console to stop the DPDK application. You can also use the serial console to stop or start the virtual machine.

4. Load *ibuverbs* on each reboot with `modprobe -a ib_uverbs`. For SLES 15 only, also load *mlx4_ib* with `modprobe -a mlx4_ib`.

Master PMD

DPDK applications must run over the master PMD that is exposed in Azure. If the application runs directly over the VF PMD, it doesn't receive all packets that are destined to the VM, since some packets show up over the synthetic interface. DPDK supports two types of master PMDs: NetVSC PMD and Failsafe PMD. A master PMD guarantees that the application receives all packets that are destined to it. It also makes sure that the application keeps running in DPDK mode over master PMD, even if the VF is revoked when the host is being serviced.

NetVSC PMD

NetVSC is the recommended PMD to run as a master PMD in Azure. It guarantees that the application receives all packets that are destined to it. It also makes sure that the application keeps running in DPDK mode, even if the VF is revoked when the host is being serviced. For more information about how to use and configure NetVSC PMD, see (https://doc.dpdk.org/guides/nics/netvsc.html).

Failsafe PMD

Note: running with failsafe PMD is not recommended in Azure. If your DPDK version is 22.11 LTS or newer, use NetVSC PMD is recommended.

As an alternative, you can run a DPDK application over the failsafe PMD. For more information about failsafe PMD see Fail-safe poll mode driver library .

Run testpmd

To run testpmd in root mode, use `sudo` before the *testpmd* command.

Basic: Sanity check, failsafe adapter initialization

1. Run the following commands to start a single port testpmd application:

```
Bash

testpmd -w <pci address from previous step> \
  -- -i \
  --port-topology=chained
```

2. Run the following commands to start a dual port testpmd application:

```
Bash

testpmd -w <pci address nic1> \
-w <pci address nic2> \
-- -i
```

After it's started, run `show port info all` to check port information. You should see one or two DPDK ports that are net_netvsc.

3. Use `start <port> /stop <port>` to start traffic.

The previous commands start *testpmd* in interactive mode, which is recommended for trying out testpmd commands.

Basic: Single sender/single receiver

The following commands periodically print the packets per second statistics:

1. On the TX side, run the following command:

```
Bash

testpmd \
  -l <core-list> \
  -n <num of mem channels> \
  -w <pci address of the device you plan to use> \
  -- --port-topology=chained \
  --nb-cores <number of cores to use for test pmd> \
  --forward-mode=txonly \
  --eth-peer=<port id>,<receiver peer MAC address> \
  --stats-period <display interval in seconds>
```

2. On the RX side, run the following command:

```
Bash

testpmd \
  -l <core-list> \
  -n <num of mem channels> \
  -w <pci address of the device you plan to use> \
  -- --port-topology=chained \
  --nb-cores <number of cores to use for test pmd> \
  --forward-mode=rxonly \
  --eth-peer=<port id>,<sender peer MAC address> \
  --stats-period <display interval in seconds>
```

When you're running the previous commands on a virtual machine, change *IP_SRC_ADDR* and *IP_DST_ADDR* in `app/test-pmd/txonly.c` to match the actual IP address of the virtual machines before you compile. Otherwise, the packets are dropped before reaching the receiver.

Advanced: Single sender/single forwarder

The following commands periodically print the packets per second statistics:

1. On the TX side, run the following command:

```bash
testpmd \
  -l <core-list> \
  -n <num of mem channels> \
  -w <pci address of the device you plan to use> \
  -- --port-topology=chained \
  --nb-cores <number of cores to use for test pmd> \
  --forward-mode=txonly \
  --eth-peer=<port id>,<receiver peer MAC address> \
  --stats-period <display interval in seconds>
```

2. On the FWD side, run the following command:

```bash
testpmd \
  -l <core-list> \
  -n <num of mem channels> \
  -w <pci address NIC1> \
  -w <pci address NIC2> \
  -- --nb-cores <number of cores to use for test pmd> \
  --forward-mode=io \
  --eth-peer=<recv port id>,<sender peer MAC address> \
  --stats-period <display interval in seconds>
```

When you're running the previous commands on a virtual machine, change *IP_SRC_ADDR* and *IP_DST_ADDR* in `app/test-pmd/txonly.c` to match the actual IP address of the virtual machines before you compile. Otherwise, the packets are dropped before reaching the forwarder. You can't have a third machine receive forwarded traffic, because the *testpmd* forwarder doesn't modify the layer-3 addresses, unless you make some code changes.

Install DPDK via system package (not recommended)

RHEL

```bash
sudo yum install -y dpdk
```

References

- EAL options
- Testpmd commands
- Packet dump commands

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

TCP/IP performance tuning for Azure VMs

Article • 04/21/2025

This article discusses common TCP/IP performance tuning techniques and some things to consider when you use them for virtual machines running on Azure. It provides a basic overview of the techniques and explored how virtual machines can be tuned.

Common TCP/IP tuning techniques

MTU, fragmentation, and large send offload

MTU

The maximum transmission unit (MTU) is the largest size frame (packet plus network access headers) specified in bytes that can be sent over a network interface. The MTU is a configurable setting. The default MTU used on Azure VMs, and the default setting on most network devices globally, is 1,500 bytes.

Fragmentation

Fragmentation occurs when a packet is sent that exceeds the MTU of a network interface. The TCP/IP stack breaks the packet into smaller pieces (fragments) that conform to the interface's MTU. Fragmentation occurs at the IP layer and is independent of the underlying protocol (such as TCP). When a 2,000-byte packet is sent over a network interface with an MTU of 1,500, the packet is broken down into one 1,500-byte packet and one 500-byte packet.

Network devices in the path between a source and destination can either drop packets that exceed the MTU or fragment the packet into smaller pieces.

The Don’t Fragment bit in an IP packet

The Don’t Fragment (DF) bit is a flag in the IP protocol header. The DF bit indicates that network devices on the path between the sender and receiver must not fragment the packet. This bit could be set for many reasons. (See the "Path MTU Discovery" section of this article for one example.) When a network device receives a packet with the Don’t Fragment bit set, and that packet exceeds the device's interface MTU, the standard behavior is for the device to drop the packet. The device sends an ICMP Fragmentation Needed message back to the original source of the packet.

Performance implications of fragmentation

Fragmentation can have negative performance implications. One of the main reasons for the effect on performance is the CPU/memory effect of the fragmentation and reassembly of packets. When a network device needs to fragment a packet, it has to allocate CPU/memory resources to perform fragmentation.

The same thing happens when the packet is reassembled. The network device must store all the fragments until they're received so it can reassemble them into the original packet.

Azure and fragmentation

Azure doesn't process fragmented packets with Accelerated Networking. When a VM receives a fragmented packet, the nonaccelerated path processes it. As a result, fragmented packets miss the benefits of Accelerated Networking, such as lower latency, reduced jitter, and higher packets per second. For this reason, we recommend avoiding fragmentation if possible.

Azure, by default, drops fragmented packets that arrive at the VM out of order, meaning the packets don't match the transmission sequence from the source endpoint. This issue can occur when packets travel over the internet or other large WANs.

Tune the MTU

You can improve internal Virtual Network throughput by increasing the MTU for your VM's traffic. However, if the VM communicates with destinations outside the Virtual Network that have a different MTU, fragmentation might occur and reduce performance.

For more information about setting the MTU on Azure VMs, see Configure Maximum Transmission Unit (MTU) for virtual machines in Azure.

Large send offload

Large send offload (LSO) can improve network performance by offloading the segmentation of packets to the ethernet adapter. When LSO is enabled, the TCP/IP stack creates a large TCP packet and sends it to the ethernet adapter for segmentation before forwarding it. The benefit of LSO frees the CPU from segmenting packets into sizes that conform to the MTU and offload that processing to the ethernet interface hardware. To learn more about the benefits of LSO, see Supporting large send offload.

When LSO is enabled, Azure customers might notice large frame sizes during packet captures. These large frame sizes might cause some customers to assume fragmentation is occurring or that a large MTU is being used when it’s not. With LSO, the ethernet adapter can advertise a

larger maximum segment size (MSS) to the TCP/IP stack to create a larger TCP packet. The ethernet adapter then breaks this entire nonsegmented frame into many smaller frames according to its MTU, which are visible in a packet capture performed on the VM.

TCP MSS window scaling and PMTUD

TCP maximum segment size

TCP maximum segment size (MSS) is a setting that limits the size of TCP segments, which avoids fragmentation of TCP packets. Operating systems typically use this formula to set MSS:

```
MSS = MTU - (IP header size + TCP header size)
```

The IP header and the TCP header are 20 bytes each, or 40 bytes total. An interface with an MTU of 1,500 has an MSS of 1,460. The MSS is configurable.

This setting is agreed to in the TCP three-way handshake when a TCP session is set up between a source and a destination. Both sides send an MSS value, and the lower of the two is used for the TCP connection.

Keep in mind that the MTUs of the source and destination aren't the only factors that determine the MSS value. Intermediary network devices, like VPN gateways, including Azure VPN Gateway, can adjust the MTU independently of the source and destination to ensure optimal network performance.

Path MTU Discovery

MSS is negotiated, but it might not indicate the actual MSS that can be used. Other network devices in the path between the source and the destination might have a lower MTU value than the source and destination. In this case, the device whose MTU is smaller than the packet drops the packet. The device sends back an ICMP Fragmentation Needed (Type 3, Code 4) message that contains its MTU. This ICMP message allows the source host to reduce its Path MTU appropriately. The process is called Path MTU Discovery (PMTUD).

The PMTUD process reduces network performance due to its inefficiency. When a network path's MTU is exceeded, packets must be retransmitted with a lower MSS. If a network firewall blocks the ICMP Fragmentation Needed message, the sender remains unaware of the need to lower the MSS and repeatedly retransmits the packet. For this reason, we advise against increasing the Azure VM MTU.

VPN and MTU

If you use VMs that perform encapsulation (like IPsec VPNs), there are some other considerations regarding packet size and MTU. VPNs add more headers to packets. The added headers increase the packet size and require a smaller MSS.

For Azure, we recommend that you set TCP MSS clamping to 1,350 bytes and tunnel interface MTU to 1,400. For more information, see the VPN devices and IPsec/IKE parameters page.

Latency, round-trip time, and TCP window scaling

Latency and round-trip time

The speed of light determines network latency over a fiber optic network. The round-trip time (RTT) between two network devices governs TCP network throughput.

Expand table

Route	Distance	One-way time	RTT
New York to San Francisco	4,148 km	21 ms	42 ms
New York to London	5,585 km	28 ms	56 ms
New York to Sydney	15,993 km	80 ms	160 ms

This table shows the straight-line distance between two locations. In networks, the distance is typically longer than the straight-line distance. Here's a simple formula to calculate minimum RTT as governed by the speed of light:

```
minimum RTT = 2 * (Distance in kilometers / Speed of propagation)
```

You can use 200 for the speed of propagation. The speed of propagation is the distance, in kilometers, that light travels in 1 millisecond.

Let's take New York to San Francisco as an example. The straight-line distance is 4,148 km. Entering that value into the equation results in the following equation:

```
Minimum RTT = 2 * (4,148 / 200)
```

The output of the equation is in milliseconds.

If you want to get the best network performance, the logical option is to select destinations with the shortest distance between them. You should also design your virtual network to optimize the path of traffic and reduce latency. For more information, see the "Network design considerations" section of this article.

Latency and round-trip time effects on TCP

Round-trip time has a direct effect on maximum TCP throughput. In TCP protocol, *window size* is the maximum amount of traffic that can be sent over a TCP connection before the sender needs to receive acknowledgment from the receiver. If the TCP MSS is set to 1,460 and the TCP window size is set to 65,535, the sender can send 45 packets before acknowledgment from the receiver. If the sender doesn't get acknowledgment, it retransmits the data. Here's the formula:

```
TCP window size / TCP MSS = packets sent
```

In this example, 65,535 / 1,460 is rounded up to 45.

This "waiting for acknowledgment" state, a mechanism to ensure reliable delivery of data, is what causes RTT to affect TCP throughput. The longer the sender waits for acknowledgment, the longer it needs to wait before sending more data.

Here's the formula for calculating the maximum throughput of a single TCP connection:

```
Window size / (RTT latency in milliseconds / 1,000) = maximum bytes/second
```

This table shows the maximum megabytes/per second throughput of a single TCP connection. (For readability, megabytes is used for the unit of measure.)

Expand table

TCP window size (bytes)	RTT latency (ms)	Maximum megabyte/second throughput	Maximum megabit/second throughput
65,535	1	65.54	524.29
65,535	30	2.18	17.48
65,535	60	1.09	8.74
65,535	90	0.73	5.83
65,535	120	0.55	4.37

If packets are lost, the maximum throughput of a TCP connection is reduced while the sender retransmits data it sent.

TCP window scaling

TCP window scaling is a technique that dynamically increases the TCP window size to allow more data to be sent before an acknowledgment is required. In the previous example, 45 packets would be sent before an acknowledgment was required. If you increase the number of

packets that can be sent before an acknowledgment is needed, you're reducing the number of times a sender is waiting for acknowledgment.

This table illustrates those relationships:

Expand table

TCP window size (bytes)	RTT latency (ms)	Maximum megabyte/second throughput	Maximum megabit/second throughput
65,535	30	2.18	17.48
131,070	30	4.37	34.95
262,140	30	8.74	69.91
524,280	30	17.48	139.81

But the TCP header value for TCP window size is only 2 bytes long, which means the maximum value for a receive window is 65,535. To increase the maximum window size, a TCP window scale factor was introduced.

The scale factor is also a setting that you can configure in an operating system. Here's the formula for calculating the TCP window size by using scale factors:

```
TCP window size = TCP window size in bytes * (2^scale factor)
```

Here's the calculation for a window scale factor of 3 and a window size of 65,535:

```
65,535 * (2^3) = 524,280 bytes
```

A scale factor of 14 results in a TCP window size of 14 (the maximum offset allowed). The TCP window size is 1,073,725,440 bytes (8.5 gigabits).

Support for TCP window scaling

Windows can set different scaling factors for different connection types. (Classes of connections include datacenter, internet, and so on.) You use the `Get-NetTCPConnection` PowerShell command to view the window scaling connection type:

PowerShell

```
Get-NetTCPConnection
```

You can use the `Get-NetTCPSetting` PowerShell command to view the values of each class:

PowerShell

```
Get-NetTCPSetting
```

You can set the initial TCP window size and TCP scaling factor in Windows by using the `Set-NetTCPSetting` PowerShell command. For more information, see Set-NetTCPSetting.

PowerShell

```
Set-NetTCPSetting
```

The following values are the effective TCP settings for `AutoTuningLevel`:

Expand table

AutoTuningLevel	Scaling factor	Scaling multiplier	Formula to calculate maximum window size
Disabled	None	None	Window size
Restricted	4	2^4	Window size * (2^4)
Highly restricted	2	2^2	Window size * (2^2)
Normal	8	2^8	Window size * (2^8)
Experimental	14	2^14	Window size * (2^14)

These settings are the most likely to affect TCP performance, but keep in mind that many other factors across the internet, outside the control of Azure, can also affect TCP performance.

Accelerated networking and receive side scaling

Accelerated networking

Virtual machine network functions historically are CPU intensive on both the guest VM and the hypervisor/host. The host CPU processes in software all packets that transit through the host, including all virtual network encapsulation and decapsulation. The more traffic that goes through the host, the higher the CPU load. If the host CPU is busy with other operations that affects network throughput and latency. Azure addresses this issue with accelerated networking.

Accelerated networking provides consistent ultralow network latency via the in-house programmable hardware of Azure and technologies like SR-IOV. Accelerated networking

moves much of the Azure software-defined networking stack off the CPUs and into FPGA-based SmartNICs. This change enables end-user applications to reclaim compute cycles that put less load on the VM decreasing jitter and inconsistency in latency. In other words, performance can be more deterministic.

Accelerated networking improves performance by allowing the guest VM to bypass the host and establish a datapath directly with a host's SmartNIC. Here are some benefits of accelerated networking:

- **Lower latency / higher packets per second (pps)**: Removing the virtual switch from the datapath eliminates the time packets spend in the host for policy processing and increases the number of packets that can be processed in the VM.
- **Reduced jitter**: Virtual switch processing depends on the amount of policy that needs to be applied and the workload of the CPU that's doing the processing. Offloading the policy enforcement to the hardware removes that variability by delivering packets directly to the VM, eliminating the host-to-VM communication and all software interrupts and context switches.
- **Decreased CPU utilization**: Bypassing the virtual switch in the host leads to less CPU utilization for processing network traffic.

To use accelerated networking, you need to explicitly enable it on each applicable VM. See Create a Linux virtual machine with Accelerated Networking for instructions.

Receive side scaling

Receive side scaling (RSS) is a network driver technology that distributes the receiving of network traffic more efficiently by distributing receive processing across multiple CPUs in a multiprocessor system. In simple terms, RSS allows a system to process more received traffic because it uses all available CPUs instead of just one. For a more technical discussion of RSS, see Introduction to receive side scaling.

To get the best performance when accelerated networking is enabled on a VM, you need to enable RSS. RSS can also provide benefits on VMs that don't use accelerated networking. For an overview of how to determine if RSS is enabled and how to enable it, see Optimize network throughput for Azure virtual machines.

TCP TIME_WAIT and TIME_WAIT assassination

TCP TIME_WAIT is another common setting that impacts network and application performance. Busy VMs frequently open and close many sockets, either as clients or servers. During normal TCP operations, a socket often enters the TIME_WAIT state and remains there for an extended

period. This state ensures the delivery of any remaining data on the socket before it closes. As a result, TCP/IP stacks typically prevent socket reuse by dropping the client's TCP SYN packet silently.

You can configure how long a socket remains in the TIME_WAIT state. The duration can range from 30 seconds to 240 seconds. Sockets are a finite resource, and their availability is configurable. Typically, about 30,000 sockets are available for use at any given time. If the system consumes all available sockets or if clients and servers use mismatched TIME_WAIT settings, the VM might attempt to reuse a socket still in the TIME_WAIT state. In such cases, new connections fail because the TCP SYN packets are silently dropped.

The value for port range for outbound sockets is configurable within the TCP/IP stack of an operating system. The same thing is true for TCP TIME_WAIT settings and socket reuse. Changing these numbers can potentially improve scalability. But, depending on the situation, these changes could cause interoperability issues. You should be careful if you change these values.

You can use TIME_WAIT assassination to address this scaling limitation. TIME_WAIT assassination allows a socket to be reused in certain situations, like when the sequence number in the IP packet of the new connection exceeds the sequence number of the last packet from the previous connection. In this case, the operating system allows the new connection to be established (it accepts the new SYN/ACK) and force close the previous connection that was in a TIME_WAIT state. This capability is supported on Windows VMs in Azure. To learn about support in other VMs, check with the OS vendor.

To learn about configuring TCP TIME_WAIT settings and source port range, see Settings that can be modified to improve network performance.

Virtual network factors that can affect performance

VM maximum outbound throughput

Azure provides various VM sizes and types, each with a different mix of performance capabilities. One of these capabilities is network throughput (or bandwidth), which is measured in megabits per second (Mbps). Because virtual machines are hosted on shared hardware, the network capacity needs to be shared fairly among the virtual machines using the same hardware. Larger virtual machines are allocated more bandwidth than smaller virtual machines.

The network bandwidth allocated to each virtual machine is measured on egress (outbound) traffic from the virtual machine. All network traffic leaving the virtual machine is counted

toward the allocated limit, regardless of destination. If a virtual machine has a 1,000-Mbps limit, that limit applies whether the outbound traffic is destined for another virtual machine in the same virtual network or one outside of Azure.

Ingress isn't measured or limited directly. But there are other factors, like CPU and storage limits, that can affect a virtual machine's ability to process incoming data.

Accelerated networking is designed to improve network performance, including latency, throughput, and CPU utilization. Accelerated networking can improve a virtual machine's throughput, but it can do that only up to the virtual machine's allocated bandwidth.

Azure virtual machines have at least one network interface attached to them. They might have several. The bandwidth allocated to a virtual machine is the sum of all outbound traffic across all network interfaces attached to the machine. In other words, the bandwidth is allocated on a per-virtual machine basis, regardless of how many network interfaces are attached to the machine.

Expected outbound throughput and the number of network interfaces supported by each VM size are detailed in Sizes for Windows virtual machines in Azure. To see maximum throughput, select a type, like **General purpose**, and then find the section about the size series on the resulting page (for example, "Dv2-series"). For each series, there's a table that provides networking specifications in the last column, which is titled "Max NICs / Expected network bandwidth (Mbps)."

The throughput limit applies to the virtual machine. Throughput isn't affected by these factors:

- **Number of network interfaces**: The bandwidth limit applies to the sum of all outbound traffic from the virtual machine.
- **Accelerated networking**: Though this feature can be helpful in achieving the published limit, it doesn't change the limit.
- **Traffic destination**: All destinations count toward the outbound limit.
- **Protocol**: All outbound traffic over all protocols counts towards the limit.

For more information, see Virtual machine network bandwidth.

Internet performance considerations

As discussed throughout this article, factors on the internet and outside the control of Azure can affect network performance. Here are some of those factors:

- **Latency**: The round-trip time between two endpoints is affected by issues on intermediate networks, by traffic that doesn't take the "shortest" distance path, and by

suboptimal peering paths.

- **Packet loss**: Packet loss is caused by network congestion, physical path issues, and underperforming network devices.
- **MTU size/Fragmentation**: Fragmentation along the path can lead to delays in data arrival or in packets arriving out of order, which can affect the delivery of packets.

Traceroute is a good tool for measuring network performance characteristics (like packet loss and latency) along every network path between a source device and a destination device.

Network design considerations

Along with the considerations discussed earlier in this article, the topology of a virtual network can affect the network's performance. For example, a hub-and-spoke design that backhauls traffic globally to a single-hub virtual network introduces network latency, which affects overall network performance.

The number of network devices that network traffic passes through can also affect overall latency. In a hub-and-spoke design, if traffic passes through a spoke network virtual appliance and a hub virtual appliance before transiting to the internet, the network virtual appliances introduce some latency.

Azure regions, virtual networks, and latency

Azure regions are made up of multiple datacenters that exist within a general geographic area. These datacenters might not be physically next to each other. In some cases, they're separated by as much as 10 kilometers. The virtual network is a logical overlay on top of the Azure physical datacenter network. A virtual network doesn't imply any specific network topology within the datacenter.

For example, two VMs that are in the same virtual network and subnet might be in different racks, rows, or even datacenters. They can be separated by feet of fiber optic cable or by kilometers of fiber optic cable. This variation could introduce variable latency (a few milliseconds difference) between different VMs.

The geographic placement of VMs, and the potential resulting latency between two VMs, is influenced by the configuration of availability sets, proximity placement groups, and availability zones. But the distance between datacenters in a region is region-specific and primarily influenced by datacenter topology in the region.

Source NAT port exhaustion

A deployment in Azure can communicate with endpoints outside of Azure on the public internet and/or in the public IP space. When an instance initiates an outbound connection, Azure dynamically maps the private IP address to a public IP address. After Azure creates this mapping, return traffic for the outbound originated flow can also reach the private IP address where the flow originated.

For every outbound connection, the Azure Load Balancer needs to maintain this mapping for some period of time. With the multitenant nature of Azure, maintaining this mapping for every outbound flow for every VM can be resource intensive. So there are limits that are set and based on the configuration of the Azure Virtual Network. Or, to say that more precisely, an Azure VM can only make some outbound connections at a given time. When these limits are reached, the VM won't make more outbound connections.

But this behavior is configurable. For more information about SNAT and SNAT port exhaustion, see this article.

Measure network performance on Azure

Many of the performance maximums in this article are related to the network latency / round-trip time (RTT) between two VMs. This section provides some suggestions for how to test latency/RTT and how to test TCP performance and VM network performance. You can tune and performance test the TCP/IP and network values discussed earlier by using the techniques described in this section. Enter latency, MTU, MSS, and window size values into the calculations provided earlier and compare theoretical maximums to actual values observed during testing.

Measure round-trip time and packet loss

TCP performance relies heavily on RTT and packet Loss. The PING utility available in Windows and Linux provides the easiest way to measure RTT and packet loss. The output of PING shows the minimum/maximum/average latency between a source and destination. It shows packet loss. PING uses the ICMP protocol by default. You can use PsPing to test TCP RTT. For more information, see PsPing.

ICMP and TCP pings don't measure the accelerated networking datapath. To measure the datapath, read about Latte and SockPerf in this article.

Measure actual bandwidth of a virtual machine

To accurately measure the bandwidth of Azure VMs, follow this guidance.

For more information on testing other scenarios, see these articles:

- Troubleshooting Expressroute network performance
- How to validate VPN throughput to a virtual network

Detect inefficient TCP behaviors

In packet captures, Azure customers might see TCP packets with TCP flags (SACK, DUP ACK, RETRANSMIT, and FAST RETRANSMIT) that could indicate network performance problems. These packets specifically indicate network inefficiencies that result from packet loss. But packet loss isn't necessarily caused by Azure performance problems. Performance issues could be the result of application, operating system, or other problems that might not be directly related to the Azure platform.

Also, keep in mind that some retransmission and duplicate ACKs are normal on a network. TCP protocols were built to be reliable. Evidence of these TCP packets in a packet capture doesn't necessarily indicate a systemic network problem, unless they're excessive.

Still, these packet types are indications that TCP throughput isn't achieving its maximum performance, for reasons discussed in other sections of this article.

Next steps

Now that you've learned about TCP/IP performance tuning for Azure VMs, consider exploring other factors for planning virtual networks. You can also learn more about connecting and configuring virtual networks.

Virtual machine network bandwidth

Azure virtual machine network throughput determines how much bandwidth your applications can use for network communication. Azure offers various virtual machine sizes and types, each with different network performance capabilities measured in megabits per second (Mbps). Understanding how bandwidth allocation works helps you optimize application performance and choose the right VM size for your workload requirements.

Each virtual machine size has a different mix of performance capabilities. One capability is network throughput (or bandwidth), measured in megabits per second (Mbps). Because virtual machines are hosted on shared hardware, the network capacity must be shared fairly among the virtual machines sharing the same hardware. Larger virtual machines are allocated relatively more bandwidth than smaller virtual machines.

The network bandwidth allocated to each virtual machine is measured on egress (outbound) traffic from the virtual machine. All network traffic leaving the virtual machine is counted toward the allocated limit, regardless of destination. For example, if a virtual machine has a 1,000-Mbps limit, that limit applies whether the outbound traffic is destined for another virtual machine in the same virtual network, or outside of Azure.

Ingress isn't measured or limited directly. However, there are other factors, such as CPU and storage limits, which can affect a virtual machine's ability to process incoming data.

Accelerated networking is a feature designed to improve network performance, including latency, throughput, and CPU utilization. While accelerated networking can improve a virtual machine's throughput, it can do so only up to the virtual machine's allocated bandwidth. To learn more about Accelerated networking, see Accelerated networking for Windows or Linux virtual machines.

Azure virtual machines must have one, but might have several, network interfaces attached to them. Bandwidth allocated to a virtual machine is the sum of all outbound traffic across all network interfaces attached to a virtual machine. In other words, the allocated bandwidth is per virtual machine, regardless of how many network interfaces are attached to the virtual machine. To learn how many network interfaces different Azure VM sizes support, see Azure Windows and Linux VM sizes.

Expected network throughput

Expected outbound throughput and the number of network interfaces supported by each VM size is detailed in Azure Windows and Linux VM sizes. Select a type, such as General purpose, then select a size and series on the resulting page, such as the Dv2-series. Each series has a table with networking specifications in the last column titled,

Max NICs / Expected network performance (Mbps).

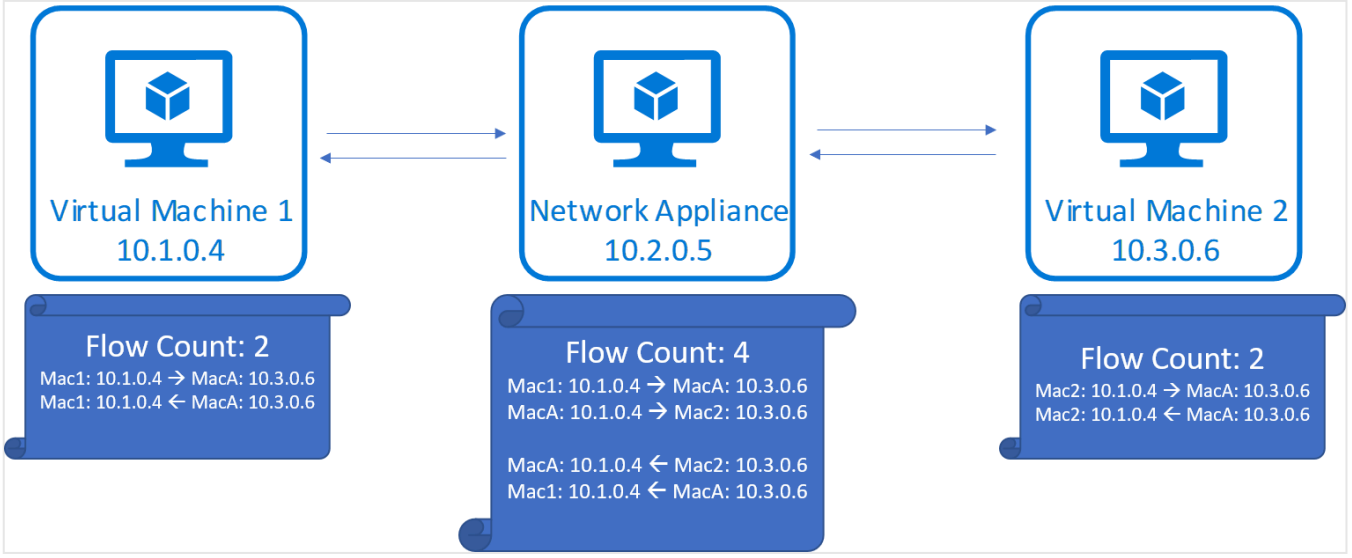
The throughput limit applies to the virtual machine. Throughput is unaffected by the following factors:

- **Number of network interfaces**: The bandwidth limit is cumulative of all outbound traffic from the virtual machine.
- **Accelerated networking**: Though the feature can be helpful in achieving the published limit, it doesn't change the limit.
- **Traffic destination**: All destinations count toward the outbound limit.
- **Protocol**: All outbound traffic over all protocols counts towards the limit.

Network flow limits

The number of network connections on a virtual machine at any moment can affect its network performance. The Azure networking stack uses data structures called **flows** to track each direction of a TCP/UDP connection. For a typical TCP/UDP connection, it creates two flows: one for inbound traffic and another for outbound traffic. A Five-tuple, consisting of protocol, local IP address, remote IP address, local port, and remote port, identifies each flow.

Data transfer between endpoints requires creation of several flows in addition to flows that perform the data transfer. Some examples are flows created for DNS resolution and flows created for load balancer health probes. Network virtual appliances (NVAs) such as gateways, proxies, firewalls, see flows created for connections terminated at the appliance and originated by the appliance.



Flow limits and active connections recommendations

Today, the Azure networking stack supports at least 500K total connections (500k inbound + 500k outbound flows) for all VM sizes. For the smallest sizes (2-7 vCPU), we recommend that your workload utilizes 100K or fewer total connections. Recommended connection limits vary based on the VM vCPU count and are shared below.

Azure Boost VM Sizes with MANA

Expand table

VM Size(#vCPUs)	Recommended Connection Limit
2-7	100,000
8-15	500,000
16-31	700,000
32-63	800,000
64+	2,000,000

Other VM Sizes

Expand table

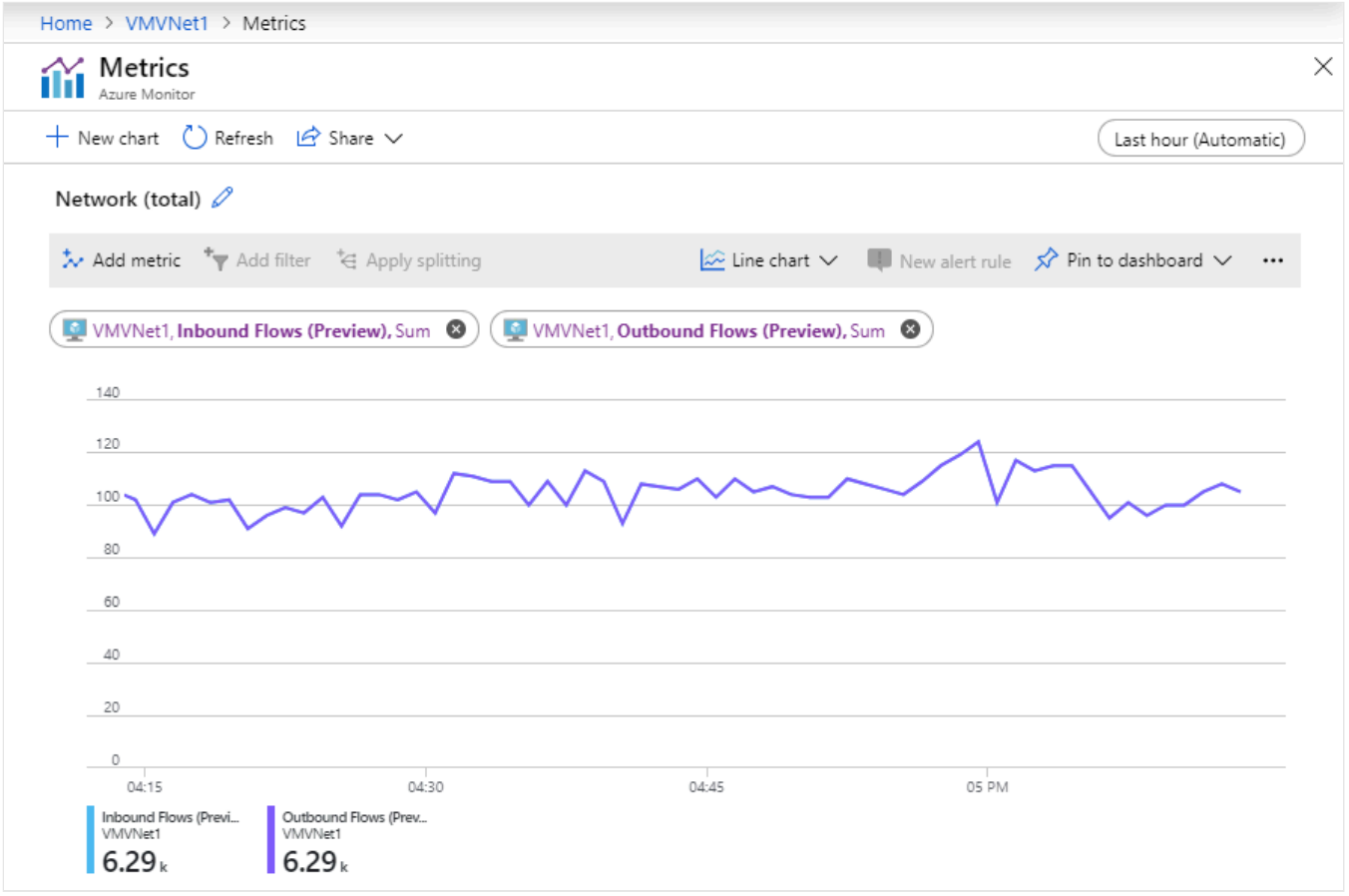
VM Size(#vCPUs)	Recommended Connection Limit
2-7	100,000
8-15	500,000
16-31	700,000
32-63	800,000
64+	1,000,000

Network Optimized VM sizes have improved network connection performance that differs from the limits above. For information on Network Optimized VM connection limits see the following article

Above the recommended limit, connections may be dropped or encounter reduced performance. Connection establishment and termination rates can also affect network performance as connection establishment and termination shares CPU with packet processing routines. We recommend that you benchmark workloads against expected traffic patterns and scale out workloads appropriately to match your performance needs. Microsoft has released a tool to make this easier, see NCPS Tool for more details.

Note, NVAs such as gateways, proxies, firewalls, and other applications that forward traffic should use half the recommended connection limits, as forwarding traffic consumes twice the number of flows when compared to typical client-server communication.

Metrics are available in Azure Monitor to track the number of network flows and the flow creation rate on your VM or Virtual Machine Scale Sets instances. It's possible that the number of flows tracked by your VM guest OS is different than the number of flows tracked by the Azure network stack for various reasons. To ensure your network connections aren't dropped, use the Inbound and Outbound Flows metric.



Next steps

- Optimize network throughput for a virtual machine operating system
- Test network throughput for a virtual machine.

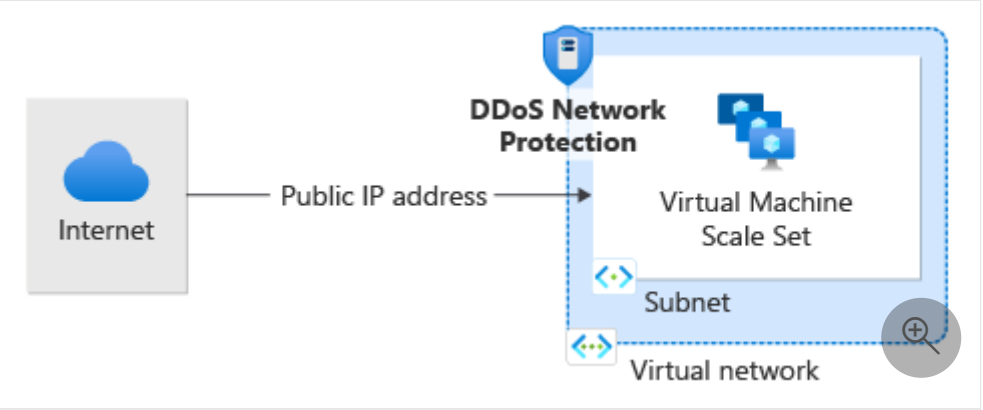
Last updated on 12/12/2025

Quickstart: Create and configure Azure DDoS Network Protection using the Azure portal

Summarize this article for me

In this Quickstart, you create a DDoS protection plan and link it to a virtual network.

A DDoS protection plan defines a set of virtual networks that have DDoS Network Protection enabled, across subscriptions. You can configure one DDoS protection plan for your organization and link virtual networks from multiple subscriptions under a single Microsoft Entra tenant to the same plan.



Prerequisites

- If you don't have an Azure subscription, create a free account before you begin.
- Sign in to the Azure portal. Ensure that your account is assigned to the network contributor role or to a custom role that is assigned the appropriate actions listed in the how-to guide on Permissions.

Create a DDoS protection plan

1. Select **Create a resource** in the upper left corner of the Azure portal.
2. Search the term *DDoS*. When **DDoS protection plan** appears in the search results, select it.
3. Select **Create**.
4. Enter or select the following values.

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Select **Create new** and enter **MyResourceGroup**.
Name	Enter **MyDdosProtectionPlan**.
Region	Enter **East US**.

5. Select **Review + create** then **Create**

> **Note**
>
> Although DDoS Protection Plan resources needs to be associated with a region, users can enable DDoS protection on Virtual Networks in different regions and across multiple subscriptions under a single Microsoft Entra tenant.

Enable DDoS protection for a virtual network

New virtual network

1. Select **Create a resource** in the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual network**.
3. Enter or select the following values then select **Next**.

Expand table

Setting	Value
Subscription	Select your subscription.
Resource group	Select **Use existing**, and then select **MyResourceGroup**
Name	Enter **MyVnet**.
Region	Enter **East US**.

4. In the *Security* pane, select **Enable** on the **Azure DDoS Network Protection** radio.

5. Select **MyDdosProtectionPlan** from the **DDoS protection plan** pane or create a new plan. The plan you select can be in the same, or different subscription than the virtual network, but both subscriptions must be associated to the same Microsoft Entra tenant.

6. Select **Next**. In the **IP addresses** pane, update the IP address space with the following values.

Expand table

Setting	Value
IPv4 address space	Enter **10.1.0.0/16**.
Subnet name	Under **Subnets**, select the **default** link and update the name to **mySubnet**.
Starting address	Enter **10.1.0.0**.
Size	Select **/24**.

7. Select **Save** > **Review + create** > **Create**.

> **Note**
>
> You cannot move a virtual network to another resource group or subscription when DDoS Protection is enabled for the virtual network. If you need to move a virtual network with DDoS Protection enabled, disable DDoS Protection first, move the virtual network, and then enable DDoS Protection. After the move, the auto-tuned policy thresholds for all the protected public IP addresses in the virtual network are reset.

Configure an Azure DDoS Protection Plan using Azure Firewall Manager

Azure Firewall Manager is a platform to manage and protect your network resources at scale. You can associate your virtual networks with a DDoS protection plan within Azure Firewall Manager. See Configure an Azure DDoS Protection Plan using Azure Firewall Manager.

Enable DDoS protection for all virtual networks

This built-in policy detects virtual networks within a defined scope that don't have DDoS Network Protection enabled. It can then optionally create a remediation task to enable protection for the virtual network. See Azure Policy built-in definitions for Azure DDoS Network Protection for full list of built-in policies.

View protected resources

First, check the details of your DDoS protection plan:

1. Search for **DDoS protection plans** in the **Search resources, services, and docs box** at the top of the Azure portal. When it appears, select it.
2. Select your DDoS protection plan from the list.
3. Under **Settings**, select **Protected resources**.
4. In the **Protected resources** page, you can view the resources that are protected by this DDoS protection plan.

Disable for a virtual network:

You can disable DDoS protection for a virtual network while keeping it enabled on other virtual networks. To disable DDoS protection for a virtual network, follow these steps.

1. Search for **Virtual Network** in the **Search resources, services, and docs box** at the top of the Azure portal. When it appears, select it.
2. Under **Settings**, select **DDoS Protection**.
3. Select **Disable** for **DDoS Network Protection**.

> **Note**
>
> Disabling DDoS protection for a virtual network won't delete the protection plan. You'll still incur costs if you only disable DDoS protection without deleting the plan. To avoid unnecessary charges, you need to delete the DDoS protection plan resource. See Clean up resources.

Clean up resources

You can keep your resources for the next tutorial. If no longer needed, delete the *MyResourceGroup* used in this example. When you delete the resource group, you also delete the DDoS protection plan and all its related resources. If you don't intend to use this DDoS protection plan, you should remove resources to avoid unnecessary charges.

> ⚠ **Warning**
>
> This action is irreversible.

1. In the Azure portal, search for and select **Resource groups**, or select **Resource groups** from the Azure portal menu.
2. Filter or scroll down to find the *MyResourceGroup* resource group.
3. Select the resource group, then select **Delete resource group**.
4. Type the resource group name to verify, and then select **Delete**.

> ⓘ **Note**
>
> To delete a DDoS protection plan, first dissociate all virtual networks from it.

Next steps

To learn how to configure metrics alerts through Azure Monitor, continue to the tutorials.

Configure Azure DDoS Protection metric alerts through portal

Last updated on 01/26/2026

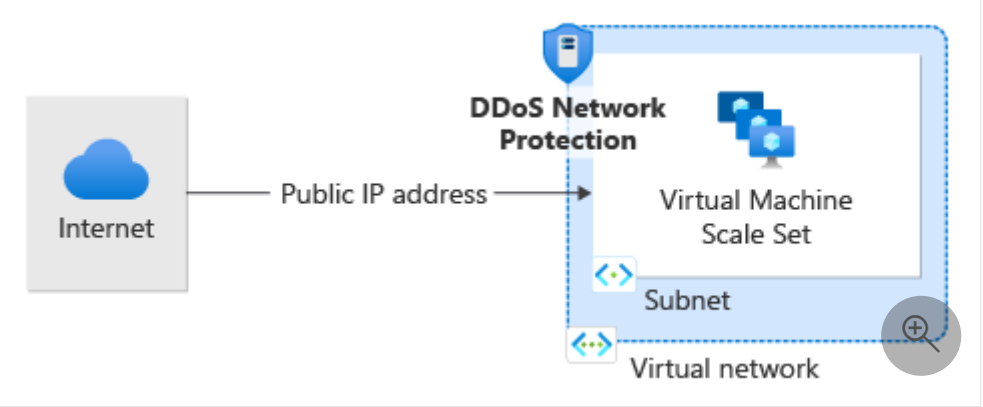
QuickStart: Create and configure Azure DDoS Network Protection using Azure PowerShell

Summarize this article for me

Get started with Azure DDoS Network Protection by using Azure PowerShell.

A DDoS protection plan defines a set of virtual networks that have DDoS Network Protection enabled, across subscriptions. You can configure one DDoS protection plan for your organization and link virtual networks from multiple subscriptions to the same plan.

In this QuickStart, you'll create a DDoS protection plan and link it to a virtual network.



Prerequisites

- An Azure account with an active subscription. Create an account for free.
- Azure PowerShell installed locally or Azure Cloud Shell.

> **Note**
>
> We recommend that you use the Azure Az PowerShell module to interact with Azure. To get started, see Install Azure PowerShell. To learn how to migrate to the Az PowerShell module, see Migrate Azure PowerShell from AzureRM to Az.

Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services.

You can use the Cloud Shell preinstalled commands to run the code in this article, without having to install anything on your local environment.

To start Azure Cloud Shell:

Expand table

Option	Example/Link
Select **Try It** in the upper-right corner of a code or command block. Selecting **Try It** doesn't automatically copy the code or command to Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To use Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block (or command block) to copy the code or command.
3. Paste the code or command into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux, or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code or command.

Create a DDoS Protection plan

In Azure, you allocate related resources to a resource group. You can either use an existing resource group or create a new one.

To create a resource group, use New-AzResourceGroup. In this example, we'll name our resource group *MyResourceGroup* and use the *East US* location:

Azure PowerShell

```
New-AzResourceGroup -Name MyResourceGroup -Location "East US"
```

Now create a DDoS protection plan named *MyDdosProtectionPlan*:

Azure PowerShell

```
New-AzDdosProtectionPlan -ResourceGroupName MyResourceGroup -Name
MyDdosProtectionPlan -Location "East US"
```

Enable DDoS for a virtual network

You can enable DDoS protection when creating a new virtual network or for an existing virtual network. Select the tab for your scenario.

New virtual network

In this example, we'll name our virtual network *MyVnet*:

Azure PowerShell

```
#Gets the DDoS protection plan ID
$ddosProtectionPlanID = Get-AzDdosProtectionPlan -ResourceGroupName
MyResourceGroup -Name MyDdosProtectionPlan

#Creates the virtual network
New-AzVirtualNetwork -Name MyVnet -ResourceGroupName MyResourceGroup -Location
"East US" -AddressPrefix 10.0.0.0/16 -DdosProtectionPlan
$ddosProtectionPlanID.Id -EnableDdosProtection
```

Disable DDoS for a virtual network

To disable DDoS protection for a virtual network:

Azure PowerShell

```
# Gets the most updated version of the virtual network
$vnet = Get-AzVirtualNetwork -Name MyVnet -ResourceGroupName MyResourceGroup
$vnet.DdosProtectionPlan = $null
$vnet.EnableDdosProtection = $false
$vnet | Set-AzVirtualNetwork
```

Validate and test

Check the details of your DDoS protection plan and verify that the command returns the correct details of your DDoS protection plan.

Azure PowerShell

```
Get-AzDdosProtectionPlan -ResourceGroupName MyResourceGroup -Name
MyDdosProtectionPlan
```

Check the details of your virtual network and verify the DDoS protection plan is enabled.

Azure PowerShell

```
Get-AzVirtualNetwork -Name MyVnet -ResourceGroupName MyResourceGroup
```

Clean up resources

You can keep your resources for the next tutorial. If no longer needed, delete the *MyResourceGroup* resource group. When you delete the resource group, you also delete the DDoS protection plan and all its related resources.

Azure PowerShell

```
Remove-AzResourceGroup -Name MyResourceGroup
```

> **Note**
>
> To delete a DDoS protection plan, first dissociate all virtual networks from it.

Next steps

To learn how to view and configure telemetry for your DDoS protection plan, continue to the tutorials.

View and configure DDoS protection telemetry

Last updated on 03/05/2026

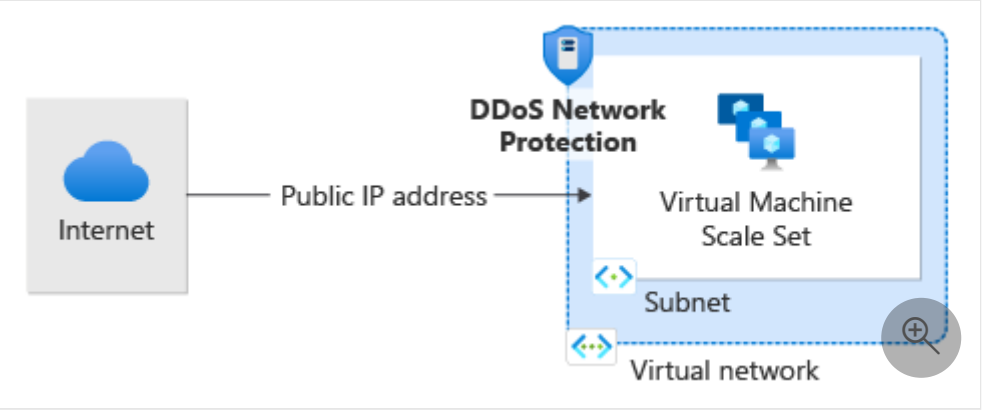
QuickStart: Create and configure Azure DDoS Network Protection using Azure CLI

Summarize this article for me

Get started with Azure DDoS Network Protection by using Azure CLI.

A DDoS protection plan defines a set of virtual networks that have DDoS Network Protection enabled, across subscriptions. You can configure one DDoS protection plan for your organization and link virtual networks from multiple subscriptions to the same plan.

In this QuickStart, you'll create a DDoS protection plan and link it to a virtual network.



Prerequisites

- An Azure account with an active subscription. Create an account for free.
- Azure CLI installed locally or Azure Cloud Shell

Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article, without having to install anything on your local environment.

To start Azure Cloud Shell:

Expand table

Option	Example/Link
Select **Try It** in the upper-right corner of a code or command block. Selecting **Try It** doesn't automatically copy the code or command to	

Option	Example/Link
Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To use Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block (or command block) to copy the code or command.
3. Paste the code or command into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux, or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code or command.

If you choose to install and use the CLI locally, this quickstart requires Azure CLI version 2.0.56 or later. To find the version, run `az --version`. If you need to install or upgrade, see Install the Azure CLI.

Create a DDoS Protection plan

In Azure, you allocate related resources to a resource group. You can either use an existing resource group or create a new one.

To create a resource group, use az group create. In this example, we'll name our resource group *MyResourceGroup* and use the *East US* location:

Azure CLI

```
az group create \
    --name MyResourceGroup \
    --location eastus
```

Now create a DDoS protection plan named *MyDdosProtectionPlan*:

Azure CLI

```
az network ddos-protection create \
    --resource-group MyResourceGroup \
```

```
    --name MyDdosProtectionPlan
```

Enable DDoS protection for a virtual network

You can enable DDoS protection when creating a new virtual network or for an existing virtual network. Select the tab for your scenario.

New virtual network

In this example, we'll name our virtual network *MyVnet*:

Azure CLI

```
az network vnet create \
    --resource-group MyResourceGroup \
    --name MyVnet \
    --location eastus \
    --ddos-protection-plan MyDdosProtectionPlan \
    --ddos-protection true
```

> **Note**
>
> You cannot move a virtual network to another resource group or subscription when DDoS Protection is enabled for the virtual network. If you need to move a virtual network with DDoS Protection enabled, disable DDoS Protection first, move the virtual network, and then enable DDoS Protection. After the move, the auto-tuned policy thresholds for all the protected public IP addresses in the virtual network are reset.

Disable DDoS protection for a virtual network

Update a given virtual network to disable DDoS protection:

Azure CLI

```
az network vnet update \
    --resource-group MyResourceGroup \
    --name MyVnet \
    --ddos-protection-plan MyDdosProtectionPlan \
    --ddos-protection false
```

Validate and test

First, check the details of your DDoS protection plan:

Azure CLI

```
az network ddos-protection show \
    --resource-group MyResourceGroup \
    --name MyDdosProtectionPlan
```

Verify that the command returns the correct details of your DDoS protection plan.

Clean up resources

You can keep your resources for the next tutorial. If no longer needed, delete the *MyResourceGroup* resource group. When you delete the resource group, you also delete the DDoS protection plan and all its related resources.

To delete the resource group, use az group delete:

Azure CLI

```
az group delete \
--name MyResourceGroup
```

> **Note**
>
> To delete a DDoS protection plan, first dissociate all virtual networks from it.

Next steps

To learn how to view and configure telemetry for your DDoS protection plan, continue to the tutorials.

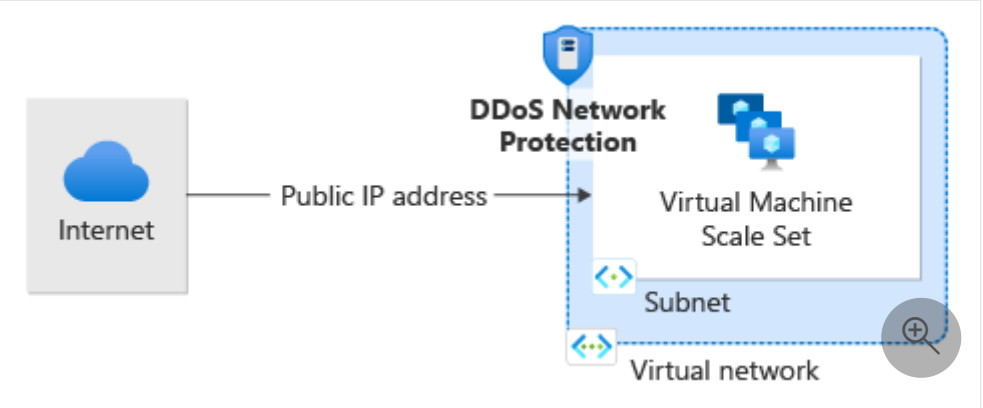
View and configure DDoS protection telemetry

Last updated on 01/26/2026

QuickStart: Create and configure Azure DDoS Network Protection using ARM template

Summarize this article for me

This QuickStart describes how to use an Azure Resource Manager template (ARM template) to create a distributed denial of service (DDoS) protection plan and virtual network, then enables the protection plan for the virtual network. An Azure DDoS Network Protection plan defines a set of virtual networks that have DDoS protection enabled across subscriptions. You can configure one DDoS protection plan for your organization and link virtual networks from multiple subscriptions to the same plan.



An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template opens in the Azure portal.

Deploy to Azure

Prerequisites

If you don't have an Azure subscription, create a free account before you begin.

Review the template

The template used in this quickstart is from Azure Quickstart Templates.

JSON


```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "metadata": {
    "_generator": {
      "name": "bicep",
      "version": "0.5.6.12127",
      "templateHash": "14909118711877377105"
    }
  },
  "parameters": {
    "ddosProtectionPlanName": {
      "type": "string",
      "metadata": {
        "description": "Specify a DDoS protection plan name."
      }
    },
    "virtualNetworkName": {
      "type": "string",
      "metadata": {
        "description": "Specify a DDoS virtual network name."
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Specify a location for the resources."
      }
    },
    "vnetAddressPrefix": {
      "type": "string",
      "defaultValue": "172.17.0.0/16",
      "metadata": {
        "description": "Specify the virtual network address prefix"
      }
    },
    "subnetPrefix": {
      "type": "string",
      "defaultValue": "172.17.0.0/24",
      "metadata": {
        "description": "Specify the virtual network subnet prefix"
      }
    },
    "ddosProtectionPlanEnabled": {
      "type": "bool",
      "defaultValue": true,
      "metadata": {
        "description": "Enable DDoS protection plan."
      }
    }
```

```
  },
  "resources": [
    {
      "type": "Microsoft.Network/ddosProtectionPlans",
      "apiVersion": "2021-05-01",
      "name": "[parameters('ddosProtectionPlanName')]",
      "location": "[parameters('location')]"
    },
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2021-05-01",
      "name": "[parameters('virtualNetworkName')]",
      "location": "[parameters('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[parameters('vnetAddressPrefix')]"
          ]
        },
        "subnets": [
          {
            "name": "default",
            "properties": {
              "addressPrefix": "[parameters('subnetPrefix')]"
            }
          }
        ],
        "enableDdosProtection": "[parameters('ddosProtectionPlanEnabled')]",
        "ddosProtectionPlan": {
          "id": "[resourceId('Microsoft.Network/ddosProtectionPlans',
parameters('ddosProtectionPlanName'))]"
        }
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/ddosProtectionPlans',
parameters('ddosProtectionPlanName'))]"
      ]
    }
  ]
}
```

The template defines two resources:

- Microsoft.Network/ddosProtectionPlans
- Microsoft.Network/virtualNetworks

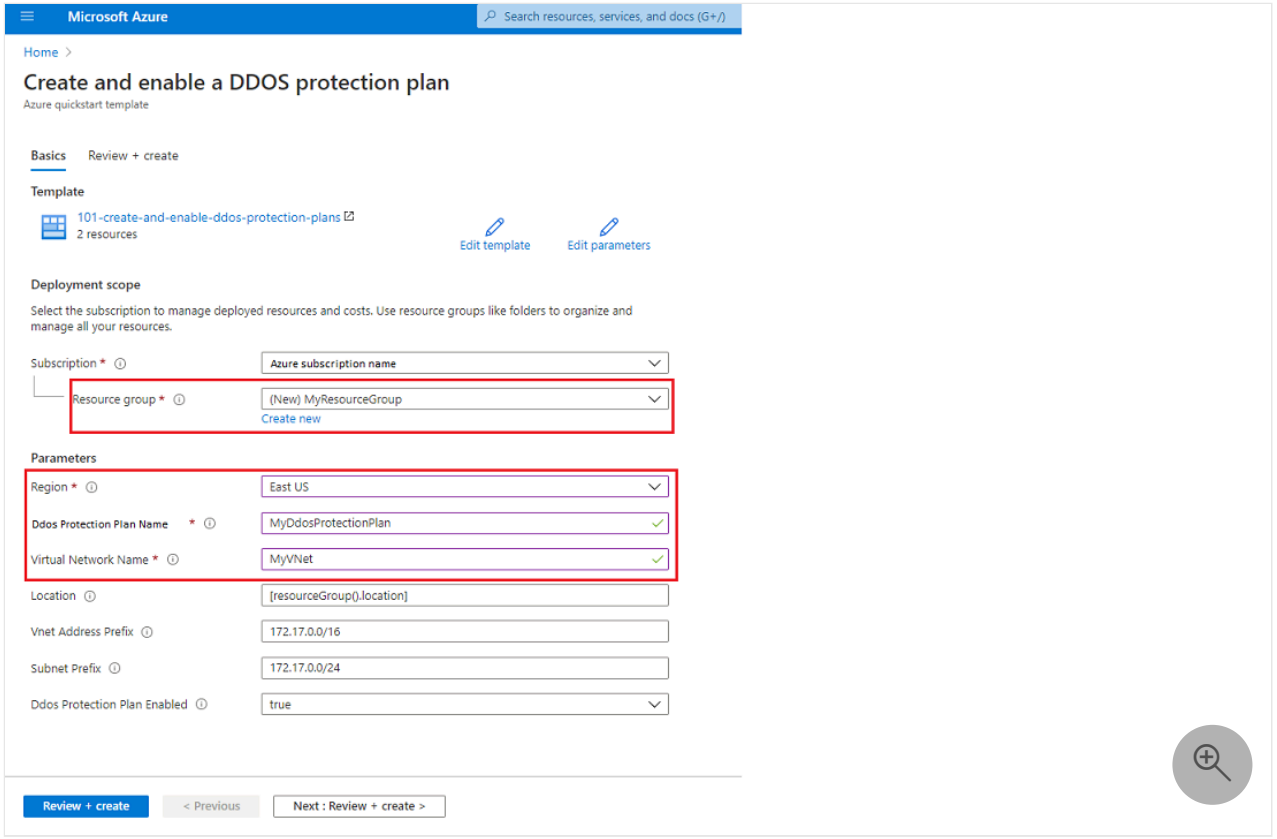
Deploy the template

In this example, the template creates a new resource group, a DDoS protection plan, and a virtual network.

1. To sign in to Azure and open the template, select the **Deploy to Azure** button.

   Deploy to Azure

2. Enter the values to create a new resource group, DDoS protection plan, and virtual network name.



   - **Subscription**: Name of the Azure subscription where the resources will be deployed.
   - **Resource group**: Select an existing resource group or create a new resource group.
   - **Region**: The region where the resource group is deployed, such as East US.
   - **Ddos Protection Plan Name**: The name of for the new DDoS protection plan.
   - **Virtual Network Name**: Creates a name for the new virtual network.
   - **Location**: Function that uses the same region as the resource group for resource deployment.
   - **Vnet Address Prefix**: Use the default value or enter your virtual network address.
   - **Subnet Prefix**: Use the default value or enter your virtual network subnet.
   - **Ddos Protection Plan Enabled**: Default is `true` to enable the DDoS protection plan.

3. Select **Review + create**.

4. Verify that template validation passed and select **Create** to begin the deployment.

Review deployed resources

To copy the Azure CLI or Azure PowerShell command, select the **Copy** button. The **Try it** button opens Azure Cloud Shell to run the command.

CLI

Azure CLI

```azurecli
az network ddos-protection show \
  --resource-group MyResourceGroup \
  --name MyDdosProtectionPlan
```

The output shows the new resources.

CLI

Output


```
{
  "etag": "W/\"abcdefgh-1111-2222-bbbb-987654321098\"",
  "id": "/subscriptions/aaaa0a0a-bb1b-cc2c-dd3d-
eeeeee4e4e4e/resourceGroups/MyResourceGroup/providers/Microsoft.Network/ddosPro
tectionPlans/MyDdosProtectionPlan",
  "location": "eastus",
  "name": "MyDdosProtectionPlan",
  "provisioningState": "Succeeded",
  "resourceGroup": "MyResourceGroup",
  "resourceGuid": null,
  "tags": null,
  "type": "Microsoft.Network/ddosProtectionPlans",
  "virtualNetworks": [
    {
      "id": "/subscriptions/aaaa0a0a-bb1b-cc2c-dd3d-
eeeeee4e4e4e/resourceGroups/MyResourceGroup/providers/Microsoft.Network/virtual
Networks/MyVNet",
      "resourceGroup": "MyResourceGroup"
    }
  ]
}
```


Clean up resources

When you're finished, you can delete the resources. The command deletes the resource group and all the resources it contains.

CLI

Azure CLI

```
az group delete --name MyResourceGroup
```

> **Note**
>
> To delete a DDoS protection plan, first dissociate all virtual networks from it.

Next steps

To learn how to view and configure telemetry for your DDoS protection plan, continue to the tutorials.

View and configure DDoS protection telemetry

Last updated on 03/05/2026

Tutorial: Migrate a virtual machine public IP address to Azure NAT Gateway

In this tutorial, you learn how to migrate your virtual machine's public IP address to a NAT gateway. You learn how to remove the IP address from the virtual machine. You reuse the IP address from the virtual machine for the NAT gateway.

Azure NAT Gateway is the recommended method for outbound connectivity. Azure NAT Gateway is a fully managed and highly resilient Network Address Translation (NAT) service. A NAT gateway doesn't have the same limitations of Source Network Address Translation (SNAT) port exhaustion as default outbound access. A NAT gateway replaces the need for a virtual machine to have a public IP address to have outbound connectivity.

For more information about Azure NAT Gateway, see What is Azure NAT Gateway?

In this tutorial, you learn how to:

- ✓ Remove the public IP address from the virtual machine.
- ✓ Associate the public IP address from the virtual machine with a NAT gateway.

Prerequisites

- An Azure account with an active subscription. Create an account for free.
- An Azure Virtual Machine with a public IP address assigned to its network interface. For more information on creating a virtual machine with a public IP, see Quickstart: Create a Windows virtual machine in the Azure portal.
  - For the purposes of this article, the example virtual machine is named **vm-1**. The example public IP address is named **public-ip**.

> **Note**
>
> Removal of the public IP address prevents direct connections to the virtual machine from the internet. RDP or SSH access won't function to the virtual machine after you complete this migration. To securely manage virtual machines in your subscription, use Azure Bastion. For more information on Azure Bastion, see What is Azure Bastion?

Remove public IP from virtual machine

In this section, you learn how to remove the public IP address from the virtual machine.

1. Sign in to the Azure portal.
2. In the search box at the top of the portal, enter **Virtual machine**. Select **Virtual machines**.
3. In **Virtual machines**, select **vm-1** or your virtual machine.
4. In the **Overview** of **vm-1**, select **Public IP address**.
5. In **public-ip**, select the **Overview** page in the left-hand column.
6. In **Overview**, select **Dissociate**.
7. Select **Yes** in **Dissociate public IP address**.

(Optional) Upgrade IP address

The NAT gateway resource requires a standard public IP address. In this section, you upgrade the IP you removed from the virtual machine in the previous section. If the IP address you removed is already a standard public IP, you can proceed to the next section.

1. In the search box at the top of the portal, enter **Public IP**. Select **Public IP addresses**.
2. In **Public IP addresses**, select **public-ip** or your basic IP address.
3. In the **Overview** of **public-ip**, select the IP address upgrade banner.
4. In **Upgrade to Standard SKU**, select the box next to **I acknowledge**. Select the **Upgrade** button.
5. When the upgrade is complete, proceed to the next section.

Create NAT gateway

In this section, you create a NAT gateway with the IP address you previously removed from the virtual machine. You assign the NAT gateway to your precreated subnet within your virtual network. The subnet name for this example is **default**.

1. In the search box at the top of the Azure portal, enter **NAT gateway**. Select **NAT gateways** in the search results.
2. Select **Create**.
3. Enter or select the following information in the **Basics** tab of **Create network address translation (NAT) gateway**.

Expand table

Setting	Value
Project details	
Subscription	Select your subscription.
Resource group	Select **test-rg** or your resource group.
Instance details	
NAT gateway name	Enter **nat-gateway**.
Region	Select your region. This example uses **East US 2**.
SKU	Select **Standard**.
TCP idle timeout (minutes)	Leave the default of **4**.

4. Select **Next**.
5. In the **Outbound IP** tab, select **+ Add public IP addresses or prefixes**.
6. In **Add public IP addresses or prefixes**, select **Public IP addresses**. Select the public IP address you removed from the virtual machine in the previous steps. In this example, it's **public-ip**.
7. Select **Next**.
8. In the **Networking** tab, in **Virtual network**, select your virtual network. In this example, it's **vnet-1**.
9. Leave the checkbox for **Default to all subnets** unchecked.
10. In **Select specific subnets**, select your subnet. In this example, it's **subnet-1**.
11. Select **Review + create**, then select **Create**.

Next step

In this article, you learned how to:

- Remove a public IP address from a virtual machine.
- Create a NAT gateway and use the public IP address from the virtual machine for the NAT gateway resource.

NAT gateway provides connectivity benefits and allows any virtual machine created within this subnet to have outbound connectivity without requiring a public IP address. For more information about NAT gateway and its connectivity benefits, see the Design virtual networks with NAT gateway documentation.

Advance to the next article to learn how to migrate default outbound access to Azure NAT Gateway:

Migrate outbound access to NAT gateway

Last updated on 11/18/2025

Create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI

Summarize this article for me

In this article, you create a virtual machine (VM) with a static public IP address. A public IP address enables you to communicate to a VM from the internet. Assign a static public IP address, rather than a dynamic address, to ensure the address never changes.

Public IP addresses have a nominal charge. There's a limit to the number of public IP addresses that you can use per subscription.

You can download the list of ranges (prefixes) for the Azure Public, US government, China, and Germany clouds.

Prerequisites

Azure portal

- An Azure account with an active subscription. You can create an account for free.

Create a virtual machine with a static public IP address

In this section, you create a virtual machine with a static public IP address using the Azure portal, Azure PowerShell, or Azure CLI. Along with the virtual machine, you create a public IP address and the other required resources.

Azure portal

Sign in to Azure

Sign in to the Azure portal.

Create a resource group

1. In the search box at the top of the portal, enter *Resource group*.
2. In the search results, select **Resource groups**.
3. Select **+ Create**.
4. In the **Basics** tab of **Create a resource group**, enter or select the following:

Expand table

Setting	Value
Subscription	Select your Azure subscription
Resource Group	Enter *myResourceGroup*.
Region	Select **East US**.

5. Select **Review + create**, then select **Create**.

Create a virtual machine

1. In the search box at the top of the portal, enter *Virtual machine*.
2. In the search results, select **Virtual machines**.
3. Select **+ Create**, then select **Azure virtual machine**.
4. In **Basics** tab of **Create a virtual machine**, enter or select the following:

Expand table

Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select **myResourceGroup**.
Instance details	
Virtual machine name	Enter *myVM*.
Region	Select **East US**.
Availability Options	Select **No infrastructure redundancy required**.
Security type	Select **Standard**.

Setting	Value
Image	Select **Windows Server 2019 Datacenter - x64 Gen2**.
Size	Choose VM size or take default setting.
Administrator account	
Username	Enter a username.
Password	Enter a password.
Confirm password	Reenter password.
Public inbound ports	Select **None**.

> **Note**
>
> All public inbound ports are closed for this virtual machine. To manage your virtual machines, deploy Azure Bastion. For more information, see Quickstart: Deploy Azure Bastion from the Azure portal.

5. Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.

6. In the **Networking** tab, enter or select the following:

Expand table

Setting	Value
Network interface	
Virtual network	Accept the default network name.
Subnet	Accept the default subnet configuration.
Public IP	Select **Create new**. In **Create public IP address**, enter *myPublicIP* in **Name**. **SKU**: select **Standard**. **Assignment**: select **Static**. Select **OK**.

> **Note**

The SKU of the virtual machine's public IP address must match the public IP SKU of Azure public load balancer when added to the backend pool of the load balancer. For details, see Azure Load Balancer.

7. Select **Review + create**.

8. Review the settings, and then select **Create**.

> ⚠ **Warning**
>
> Do not modify the IP address settings within the virtual machine's operating system. The operating system is unaware of Azure public IP addresses. Though you can add private IP address settings to the operating system, we recommend not doing so unless necessary. For more information, see Add a private IP address to an operating system.

> ⓘ **Note**
>
> Azure provides a default outbound access IP for VMs that either aren't assigned a public IP address or are in the backend pool of an internal basic Azure load balancer. The default outbound access IP mechanism provides an outbound IP address that isn't configurable.
>
> The default outbound access IP is disabled when one of the following events happens:
>
> - A public IP address is assigned to the VM.
> - The VM is placed in the backend pool of a standard load balancer, with or without outbound rules.
> - An Azure NAT Gateway resource is assigned to the subnet of the VM.
>
> VMs that you create by using virtual machine scale sets in flexible orchestration mode don't have default outbound access.
>
> For more information about outbound connections in Azure, see Default outbound access in Azure and Use Source Network Address Translation (SNAT) for outbound connections.

Clean up resources

When resources are no longer needed, delete all resources created in this article to avoid incurring charges.

Azure portal

Use the Azure portal to delete the resource group and all of the resources it contains:

1. Enter *myResourceGroup* in the search box at the top of the portal. When you see **myResourceGroup** in the search results, select it.
2. Select **Delete resource group**.
3. Enter *myResourceGroup* for **TYPE THE RESOURCE GROUP NAME:** and select **Delete**.

Next steps

In this article, you learned how to create a VM with a static public IP.

- Learn how to Configure IP addresses for an Azure network interface.
- Learn how to Assign multiple IP addresses to virtual machines using the Azure portal.
- Learn more about public IP addresses in Azure.

Last updated on 02/25/2026

Add Custom Domain to Azure VM or resource

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

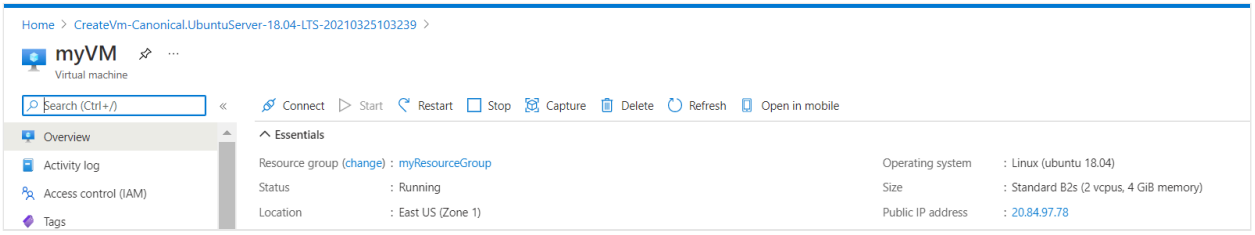
In Azure there are multiple ways to connect a custom domain to your VM or resource. For any resource with a public IP (Virtual Machine, Load Balancer, Application Gateway) the most straight-forward way is to create an A record set in your corresponding domain registrar.

Prerequisites

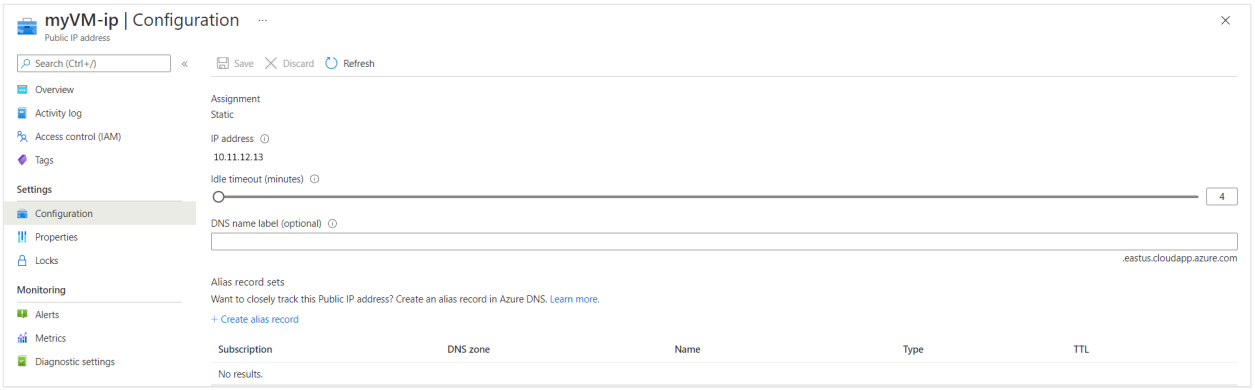
- You need a VM with a web server running. You can use the Quickstart to create a VM and add NGINX.
- The VM must be accessible to the web (open port 80, or 443). For a more secure deployment place your VM behind a load balancer or Application Gateway first. For more information, see Quickstart: Load Balancer.
- Have an existing domain and access to DNS settings. For more information, see Buy a custom domain for Azure App Service.

Add custom domain to VM public IP address

When you create a virtual machine in the Azure portal, a public IP resource for the virtual machine is automatically created. Your public IP address is shown in VM overview page.



If you select the IP address you can see more information on it. Check to make sure your **IP Assignment** is set to **Static**. A static IP address will not change if the VM or resource reboots or shuts down.



If your IP Address is not static, you will need to create an FQDN.

1. Select your VM in the portal.
2. In the left menu, select **Properties**
3. Under **Public IP address\DNS name label**, select your IP address. The **Configuration** page will open.
4. Under **DNS name label**, enter the prefix you want to use.
5. Select **Save** at the top of the page.
6. Select **Overview** in the left menu to return to the VM overview blade.
7. Verify that the *DNS name* appears correctly.

Open a browser and enter your IP address or FQDN and verify that it shows the web content running on your VM.

After verifying your static IP or FQDN, go to your domain provider and navigate to DNS settings. Add an *A record* pointing to your Public IP Address or FQDN. For example, the procedure for the GoDaddy domain registrar is as follows:

1. Sign in and select the custom domain you want to use.
2. In the **Domains** section, select **Manage All**, then select **DNS | Manage Zones**.
3. For **Domain Name**, enter your custom domain, then select **Search**.
4. From the DNS Management page, select Add, then select A in the Type list.
5. Complete the fields of the A entry:

    - Type: Leave **A** selected.
    - Host: Enter @
    - Points to: Enter the Public IP Address or FQDN of your VM.
    - TTL: Leave one hour selected.

6. Select **Save**.

The A record entry is added to the DNS records table.

After the record is created it usually takes about an hour for DNS propagate, but it can sometimes take up to 48 hours.

Next steps

Overview of TLS termination and end to end TLS with Application Gateway.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to create a Linux virtual machine in Azure with multiple network interface cards

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article details how to create a VM with multiple NICs with the Azure CLI.

Create supporting resources

Install the latest Azure CLI and log in to an Azure account using az login.

In the following examples, replace example parameter names with your own values. Example parameter names included *myResourceGroup*, *mystorageaccount*, and *myVM*.

First, create a resource group with az group create. The following example creates a resource group named *myResourceGroup* in the *eastus* location:

Azure CLI

```
az group create --name myResourceGroup --location eastus
```

Create the virtual network with az network vnet create. The following example creates a virtual network named *myVnet* and subnet named *mySubnetFrontEnd*:

Azure CLI

```
az network vnet create \
    --resource-group myResourceGroup \
    --name myVnet \
    --address-prefix 10.0.0.0/16 \
    --subnet-name mySubnetFrontEnd \
    --subnet-prefix 10.0.1.0/24
```

Create a subnet for the back-end traffic with az network vnet subnet create. The following example creates a subnet named *mySubnetBackEnd*:

Azure CLI

```
az network vnet subnet create \
    --resource-group myResourceGroup \
```

```
    --vnet-name myVnet \
    --name mySubnetBackEnd \
    --address-prefix 10.0.2.0/24
```

Create a network security group with az network nsg create. The following example creates a network security group named *myNetworkSecurityGroup*:

Azure CLI

```
az network nsg create \
    --resource-group myResourceGroup \
    --name myNetworkSecurityGroup
```

Create and configure multiple NICs

Create two NICs with az network nic create. The following example creates two NICs, named *myNic1* and *myNic2*, connected the network security group, with one NIC connecting to each subnet:

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic1 \
    --vnet-name myVnet \
    --subnet mySubnetFrontEnd \
    --network-security-group myNetworkSecurityGroup
az network nic create \
    --resource-group myResourceGroup \
    --name myNic2 \
    --vnet-name myVnet \
    --subnet mySubnetBackEnd \
    --network-security-group myNetworkSecurityGroup
```

Create a VM and attach the NICs

When you create the VM, specify the NICs you created with `--nics`. You also need to take care when you select the VM size. There are limits for the total number of NICs that you can add to a VM. Read more about Linux VM sizes.

Create a VM with az vm create. The following example creates a VM named *myVM*:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --image Ubuntu2204 \
    --size Standard_DS3_v2 \
    --admin-username azureuser \
    --generate-ssh-keys \
    --nics myNic1 myNic2
```

Add routing tables to the guest OS by completing the steps in Configure the guest OS for multiple NICs.

Add a NIC to a VM

The previous steps created a VM with multiple NICs. You can also add NICs to an existing VM with the Azure CLI. Different VM sizes support a varying number of NICs, so size your VM accordingly. If needed, you can resize a VM.

Create another NIC with az network nic create. The following example creates a NIC named *myNic3* connected to the back-end subnet and network security group created in the previous steps:

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic3 \
    --vnet-name myVnet \
    --subnet mySubnetBackEnd \
    --network-security-group myNetworkSecurityGroup
```

To add a NIC to an existing VM, first deallocate the VM with az vm deallocate. The following example deallocates the VM named *myVM*:

Azure CLI

```
az vm deallocate --resource-group myResourceGroup --name myVM
```

Add the NIC with az vm nic add. The following example adds *myNic3* to *myVM*:

Azure CLI

```
az vm nic add \
    --resource-group myResourceGroup \
```

```
    --vm-name myVM \
    --nics myNic3
```

Start the VM with az vm start:

Azure CLI

```
az vm start --resource-group myResourceGroup --name myVM
```

Add routing tables to the guest OS by completing the steps in Configure the guest OS for multiple NICs.

Remove a NIC from a VM

To remove a NIC from an existing VM, first deallocate the VM with az vm deallocate. The following example deallocates the VM named *myVM*:

Azure CLI

```
az vm deallocate --resource-group myResourceGroup --name myVM
```

Remove the NIC with az vm nic remove. The following example removes *myNic3* from *myVM*:

Azure CLI

```
az vm nic remove \
    --resource-group myResourceGroup \
    --vm-name myVM \
    --nics myNic3
```

Start the VM with az vm start:

Azure CLI

```
az vm start --resource-group myResourceGroup --name myVM
```

Create multiple NICs using Resource Manager templates

Azure Resource Manager templates use declarative JSON files to define your environment. You can read an overview of Azure Resource Manager. Resource Manager

templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. You use *copy* to specify the number of instances to create:

JSON

```
"copy": {
    "name": "multiplenics"
    "count": "[parameters('count')]"
}
```

Read more about creating multiple instances using *copy*.

You can also use a `copyIndex()` to then append a number to a resource name, which allows you to create `myNic1`, `myNic2`, etc. The following shows an example of appending the index value:

JSON

```
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of creating multiple NICs using Resource Manager templates.

Add routing tables to the guest OS by completing the steps in Configure the guest OS for multiple NICs.

Configure guest OS for multiple NICs

The previous steps created a virtual network and subnet, attached NICs, then created a VM. A public IP address and network security group rules that allow SSH traffic were not created. To configure the guest OS for multiple NICs, you need to allow remote connections and run commands locally on the VM.

To allow SSH traffic, create a network security group rule with az network nsg rule create as follows:

Azure CLI

```
az network nsg rule create \
    --resource-group myResourceGroup \
    --nsg-name myNetworkSecurityGroup \
    --name allow_ssh \
```

```
    --priority 101 \
    --destination-port-ranges 22
```

Create a public IP address with az network public-ip create and assign it to the first NIC with az network nic ip-config update:

Azure CLI

```
az network public-ip create --resource-group myResourceGroup --name
myPublicIP

az network nic ip-config update \
    --resource-group myResourceGroup \
    --nic-name myNic1 \
    --name ipconfig1 \
    --public-ip myPublicIP
```

To view the public IP address of the VM, use az vm show as follows::

Azure CLI

```
az vm show --resource-group myResourceGroup --name myVM -d --query publicIps
-o tsv
```

Now SSH to the public IP address of your VM. The default username provided in a previous step was *azureuser*. Provide your own username and public IP address:

Bash

```
ssh azureuser@137.117.58.232
```

To send to or from a secondary network interface, you have to manually add persistent routes to the operating system for each secondary network interface. In this article, *eth1* is the secondary interface. Instructions for adding persistent routes to the operating system vary by distro. See documentation for your distro for instructions.

When adding the route to the operating system, the gateway address is the first address of the subnet the network interface is in. For example, if the subnet has been assigned the range *10.0.2.0/24*, the gateway you specify for the route is *10.0.2.1* or if the subnet has been assigned the range *10.0.2.128/25*, the gateway you specify for the route is *10.0.2.129*. You can define a specific network for the route's destination, or specify a destination of *0.0.0.0*, if you want all traffic for the interface to go through the specified gateway. The gateway for each subnet is managed by the virtual network.

Once you've added the route for a secondary interface, verify that the route is in your route table with `route -n`. The following example output is for the route table that has the two network interfaces added to the VM in this article:

```
Bash

Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
0.0.0.0         10.0.1.1        0.0.0.0         UG    0      0        0 eth0
0.0.0.0         10.0.2.1        0.0.0.0         UG    0      0        0 eth1
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
168.63.129.16   10.0.1.1        255.255.255.255 UGH   0      0        0 eth0
169.254.169.254 10.0.1.1        255.255.255.255 UGH   0      0        0 eth0
```

Confirm that the route you added persists across reboots by checking your route table again after a reboot. To test connectivity, you can enter the following command, for example, where *eth1* is the name of a secondary network interface:

```
Bash

ping bing.com -c 4 -I eth1
```

Next steps

Review Linux VM sizes when trying to creating a VM with multiple NICs. Pay attention to the maximum number of NICs each VM size supports.

To further secure your VMs, use just in time VM access. This feature opens network security group rules for SSH traffic when needed, and for a defined period of time. For more information, see Manage virtual machine access using just in time.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create and manage a Windows virtual machine that has multiple NICs

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

Virtual machines (VMs) in Azure can have multiple virtual network interface cards (NICs) attached to them. A common scenario is to have different subnets for front-end and back-end connectivity. You can associate multiple NICs on a VM to multiple subnets, but those subnets must all reside in the same virtual network (vNet). This article details how to create a VM that has multiple NICs attached to it. You also learn how to add or remove NICs from an existing VM. Different VM sizes support a varying number of NICs, so size your VM accordingly.

> **Note**
>
> If multiple subnets are not required for a scenario, it may be more straightforward to utilize multiple IP configurations on a single NIC. Instructions for this setup can be found here.

Prerequisites

In the following examples, replace example parameter names with your own values. Example parameter names include *myResourceGroup*, *myVnet*, and *myVM*.

Create a VM with multiple NICs

First, create a resource group. The following example creates a resource group named *myResourceGroup* in the *EastUs* location:

PowerShell

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

Create virtual network and subnets

A common scenario is for a virtual network to have two or more subnets. One subnet may be for front-end traffic, the other for back-end traffic. To connect to both subnets,

you then use multiple NICs on your VM.

1. Define two virtual network subnets with New-AzVirtualNetworkSubnetConfig. The following example defines the subnets for *mySubnetFrontEnd* and *mySubnetBackEnd*:

   PowerShell

   ```
   $mySubnetFrontEnd = New-AzVirtualNetworkSubnetConfig -Name
   "mySubnetFrontEnd" `
       -AddressPrefix "192.168.1.0/24"
   $mySubnetBackEnd = New-AzVirtualNetworkSubnetConfig -Name
   "mySubnetBackEnd" `
       -AddressPrefix "192.168.2.0/24"
   ```

2. Create your virtual network and subnets with New-AzVirtualNetwork. The following example creates a virtual network named *myVnet*:

   PowerShell

   ```
   $myVnet = New-AzVirtualNetwork -ResourceGroupName "myResourceGroup" `
       -Location "EastUs" `
       -Name "myVnet" `
       -AddressPrefix "192.168.0.0/16" `
       -Subnet $mySubnetFrontEnd,$mySubnetBackEnd
   ```

Create multiple NICs

Create two NICs with New-AzNetworkInterface. Attach one NIC to the front-end subnet and one NIC to the back-end subnet. The following example creates NICs named *myNic1* and *myNic2*:

PowerShell

```
$frontEnd = $myVnet.Subnets|?{$_.Name -eq 'mySubnetFrontEnd'}
$myNic1 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
    -Name "myNic1" `
    -Location "EastUs" `
    -SubnetId $frontEnd.Id

$backEnd = $myVnet.Subnets|?{$_.Name -eq 'mySubnetBackEnd'}
$myNic2 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
    -Name "myNic2" `
    -Location "EastUs" `
    -SubnetId $backEnd.Id
```

Typically you also create a network security group to filter network traffic to the VM and a load balancer to distribute traffic across multiple VMs.

Create the virtual machine

Now start to build your VM configuration. Each VM size has a limit for the total number of NICs that you can add to a VM. For more information, see Windows VM sizes.

1. Set your VM credentials to the `$cred` variable as follows:

   PowerShell

   ```
   $cred = Get-Credential
   ```

2. Define your VM with New-AzVMConfig. The following example defines a VM named *myVM* and uses a VM size that supports more than two NICs (*Standard_DS3_v2*):

   PowerShell

   ```
   $vmConfig = New-AzVMConfig -VMName "myVM" -VMSize "Standard_DS3_v2"
   ```

3. Create the rest of your VM configuration with Set-AzVMOperatingSystem and Set-AzVMSourceImage. The following example creates a Windows Server 2016 VM:

   PowerShell

   ```
   $vmConfig = Set-AzVMOperatingSystem -VM $vmConfig `
       -Windows `
       -ComputerName "myVM" `
       -Credential $cred `
       -ProvisionVMAgent `
       -EnableAutoUpdate
   $vmConfig = Set-AzVMSourceImage -VM $vmConfig `
       -PublisherName "MicrosoftWindowsServer" `
       -Offer "WindowsServer" `
       -Skus "2016-Datacenter" `
       -Version "latest"
   ```

4. Attach the two NICs that you previously created with Add-AzVMNetworkInterface:

   PowerShell

   ```
   $vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic1.Id -
   Primary
   ```

```
$vmConfig = Add-AzVMNetworkInterface -VM $vmConfig -Id $myNic2.Id
```

5. Create your VM with New-AzVM:

```
PowerShell

New-AzVM -VM $vmConfig -ResourceGroupName "myResourceGroup" -Location
"EastUs"
```

6. Add routes for secondary NICs to the OS by completing the steps in Configure the operating system for multiple NICs.

Add a NIC to an existing VM

To add a virtual NIC to an existing VM, you deallocate the VM, add the virtual NIC, then start the VM. Different VM sizes support a varying number of NICs, so size your VM accordingly. If needed, you can resize a VM.

1. Deallocate the VM with Stop-AzVM. The following example deallocates the VM named *myVM* in *myResourceGroup*:

```
PowerShell

Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
```

2. Get the existing configuration of the VM with Get-AzVm. The following example gets information for the VM named *myVM* in *myResourceGroup*:

```
PowerShell

$vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
```

3. The following example creates a virtual NIC with New-AzNetworkInterface named *myNic3* that is attached to *mySubnetBackEnd*. The virtual NIC is then attached to the VM named *myVM* in *myResourceGroup* with Add-AzVMNetworkInterface:

```
PowerShell

# Get info for the back end subnet
$myVnet = Get-AzVirtualNetwork -Name "myVnet" -ResourceGroupName
"myResourceGroup"
$backEnd = $myVnet.Subnets|?{$_.Name -eq 'mySubnetBackEnd'}

# Create a virtual NIC
```

```
$myNic3 = New-AzNetworkInterface -ResourceGroupName "myResourceGroup" `
    -Name "myNic3" `
    -Location "EastUs" `
    -SubnetId $backEnd.Id

# Get the ID of the new virtual NIC and add to VM
$nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -
Name "MyNic3").Id
Add-AzVMNetworkInterface -VM $vm -Id $nicId | Update-AzVm -
ResourceGroupName "myResourceGroup"
```

Primary virtual NICs

One of the NICs on a multi-NIC VM needs to be primary. If one of the existing virtual NICs on the VM is already set as primary, you can skip this step. The following example assumes that two virtual NICs are now present on a VM and you wish to add the first NIC (`[0]`) as the primary:

PowerShell

```
# List existing NICs on the VM and find which one is primary
$vm.NetworkProfile.NetworkInterfaces

# Set NIC 0 to be primary
$vm.NetworkProfile.NetworkInterfaces[0].Primary = $true
$vm.NetworkProfile.NetworkInterfaces[1].Primary = $false

# Update the VM state in Azure
Update-AzVM -VM $vm -ResourceGroupName "myResourceGroup"
```

4. Start the VM with Start-AzVm:

PowerShell

```
Start-AzVM -ResourceGroupName "myResourceGroup" -Name "myVM"
```

5. Add routes for secondary NICs to the OS by completing the steps in Configure the operating system for multiple NICs.

Remove a NIC from an existing VM

To remove a virtual NIC from an existing VM, you deallocate the VM, remove the virtual NIC, then start the VM.

1. Deallocate the VM with Stop-AzVM. The following example deallocates the VM named *myVM* in *myResourceGroup*:

   PowerShell

   ```
   Stop-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
   ```

2. Get the existing configuration of the VM with Get-AzVm. The following example gets information for the VM named *myVM* in *myResourceGroup*:

   PowerShell

   ```
   $vm = Get-AzVm -Name "myVM" -ResourceGroupName "myResourceGroup"
   ```

3. Get information about the NIC remove with Get-AzNetworkInterface. The following example gets information about *myNic3*:

   PowerShell

   ```
   # List existing NICs on the VM if you need to determine NIC name
   $vm.NetworkProfile.NetworkInterfaces

   $nicId = (Get-AzNetworkInterface -ResourceGroupName "myResourceGroup" -
   Name "myNic3").Id
   ```

4. Remove the NIC with Remove-AzVMNetworkInterface and then update the VM with Update-AzVm. The following example removes *myNic3* as obtained by `$nicId` in the preceding step:

   PowerShell

   ```
   Remove-AzVMNetworkInterface -VM $vm -NetworkInterfaceIDs $nicId | `
       Update-AzVm -ResourceGroupName "myResourceGroup"
   ```

5. Start the VM with Start-AzVm:

   PowerShell

   ```
   Start-AzVM -Name "myVM" -ResourceGroupName "myResourceGroup"
   ```

Create multiple NICs with templates

Azure Resource Manager templates provide a way to create multiple instances of a resource during deployment, such as creating multiple NICs. Resource Manager templates use declarative JSON files to define your environment. For more information, see overview of Azure Resource Manager. You can use *copy* to specify the number of instances to create:

```JSON
"copy": {
    "name": "multiplenics",
    "count": "[parameters('count')]"
}
```

For more information, see creating multiple instances by using *copy*.

You can also use `copyIndex()` to append a number to a resource name. You can then create *myNic1*, *MyNic2* and so on. The following code shows an example of appending the index value:

```JSON
"name": "[concat('myNic', copyIndex())]",
```

You can read a complete example of creating multiple NICs by using Resource Manager templates.

Add routes for secondary NICs to the OS by completing the steps in Configure the operating system for multiple NICs.

Configure guest OS for multiple NICs

Azure assigns a default gateway to the first (primary) network interface attached to the virtual machine. Azure does not assign a default gateway to additional (secondary) network interfaces attached to a virtual machine. Therefore, you are unable to communicate with resources outside the subnet that a secondary network interface is in, by default. Secondary network interfaces can, however, communicate with resources outside their subnet, though the steps to enable communication are different for different operating systems.

1. From a Windows command prompt, run the `route print` command, which returns output similar to the following output for a virtual machine with two attached network interfaces:

```
===========================================================================
====
Interface List
3...00 0d 3a 10 92 ce ......Microsoft Hyper-V Network Adapter #3
7...00 0d 3a 10 9b 2a ......Microsoft Hyper-V Network Adapter #4
===========================================================================
====
```

In this example, **Microsoft Hyper-V Network Adapter #4** (interface 7) is the secondary network interface that doesn't have a default gateway assigned to it.

2. From a command prompt, run the `ipconfig` command to see which IP address is assigned to the secondary network interface. In this example, 192.168.2.4 is assigned to interface 7. No default gateway address is returned for the secondary network interface.

3. To route all traffic destined for addresses outside the subnet of the secondary network interface to the gateway for the subnet, run the following command:

```
route add -p 0.0.0.0 MASK 0.0.0.0 192.168.2.1 METRIC 5015 IF 7
```

The gateway address for the subnet is the first IP address (ending in .1) in the address range defined for the subnet. If you don't want to route all traffic outside the subnet, you could add individual routes to specific destinations, instead. For example, if you only wanted to route traffic from the secondary network interface to the 192.168.3.0 network, you enter the command:

```
route add -p 192.168.3.0 MASK 255.255.255.0 192.168.2.1 METRIC 5015 IF
7
```

4. To confirm successful communication with a resource on the 192.168.3.0 network, for example, enter the following command to ping 192.168.3.4 using interface 7 (192.168.2.4):

```
ping 192.168.3.4 -S 192.168.2.4
```

You may need to open ICMP through the Windows firewall of the device you're pinging with the following command:

```
netsh advfirewall firewall add rule name=Allow-ping protocol=icmpv4
dir=in action=allow
```

5. To confirm the added route is in the route table, enter the `route print` command, which returns output similar to the following text:

```
===========================================================================
====
Active Routes:
Network Destination        Netmask          Gateway       Interface
Metric
          0.0.0.0          0.0.0.0      192.168.1.1      192.168.1.4
15
          0.0.0.0          0.0.0.0      192.168.2.1      192.168.2.4
5015
```

The route listed with *192.168.1.1* under **Gateway**, is the route that is there by default for the primary network interface. The route with *192.168.2.1* under **Gateway**, is the route you added.

Next steps

Review Windows VM sizes when you're trying to create a VM that has multiple NICs. Pay attention to the maximum number of NICs that each VM size supports.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create a fully qualified domain name for a VM in the Azure portal

Article • 08/22/2024

Applies to: Applies to: ✔ Linux VMs ✔ Windows VMs

When you create a virtual machine (VM) in the Azure portal, a public IP resource for the virtual machine is automatically created. You use this public IP address to remotely access the VM. Although the portal doesn't create a fully qualified domain name, or FQDN, you can add one once the VM is created. This article demonstrates the steps to create a DNS name or FQDN. If you create a VM without a public IP address, you can't create an FQDN.

Create an FQDN

This article assumes that you've already created a VM. If needed, you can create a Linux or Windows VM in the portal. Follow these steps once your VM is up and running:

1. Select your VM in the portal.
2. In the left menu, select **Properties**
3. Under **Public IP address\DNS name label**, select your IP address.
4. Under **DNS name label**, enter the prefix you want to use.
5. Select **Save** at the top of the page.
6. Select **Overview** in the left menu to return to the VM overview blade.
7. Verify that the **DNS name** appears correctly.

Next steps

You can also manage DNS using Azure DNS zones.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

DNS Name Resolution options for Linux virtual machines in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

Azure provides DNS name resolution by default for all virtual machines that are in a single virtual network. You can implement your own DNS name resolution solution by configuring your own DNS services on your virtual machines that Azure hosts. The following scenarios should help you choose the one that works for your situation.

- Name resolution that Azure provides
- Name resolution using your own DNS server

The type of name resolution that you use depends on how your virtual machines and role instances need to communicate with each other.

The following table illustrates scenarios and corresponding name resolution solutions:

Expand table

Scenario	Solution	Suffix
Name resolution between role instances or virtual machines in the same virtual network	Name resolution that Azure provides	hostname or fully-qualified domain name (FQDN)
Name resolution between role instances or virtual machines in different virtual networks	Customer-managed DNS servers that forward queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server.	FQDN only
Resolution of on-premises computers and service names from role instances or virtual machines in Azure	Customer-managed DNS servers (for example, on-premises domain controller, local read-only domain controller, or a DNS secondary synced by using zone transfers). See Name resolution using your own DNS server.	FQDN only
Resolution of Azure hostnames from on-premises computers	Forward queries to a customer-managed DNS proxy server in the corresponding virtual network. The proxy server forwards queries to Azure for resolution. See Name resolution using your own DNS server.	FQDN only

Scenario	Solution	Suffix
Reverse DNS for internal IPs	Name resolution using your own DNS server	n/a

Name resolution that Azure provides

Along with resolution of public DNS names, Azure provides internal name resolution for virtual machines and role instances that are in the same virtual network. In virtual networks that are based on Azure Resource Manager, the DNS suffix is consistent across the virtual network; the FQDN isn't needed. DNS names can be assigned to both network interface cards (NICs) and virtual machines. Although the name resolution that Azure provides does not require any configuration, it isn't the appropriate choice for all deployment scenarios, as seen on the preceding table.

Features and considerations

Features:

- No configuration is required to use name resolution that Azure provides.
- The name resolution service that Azure provides is highly available. You don't need to create and manage clusters of your own DNS servers.
- The name resolution service that Azure provides can be used along with your own DNS servers to resolve both on-premises and Azure hostnames.
- Name resolution is provided between virtual machines in virtual networks without need for the FQDN.
- You can use hostnames that best describe your deployments rather than working with auto-generated names.

Considerations:

- The DNS suffix that Azure creates can't be modified.
- You can't manually register your own records.
- WINS and NetBIOS aren't supported.
- Hostnames must be DNS-compatible. Names must use only 0-9, a-z, and '-', and they can't start or end with a '-'. See RFC 3696 Section 2.
- DNS query traffic is throttled for each virtual machine. Throttling shouldn't impact most applications. If request throttling is observed, ensure that client-side caching is enabled. For more information, see Getting the most from name resolution that Azure provides.

Getting the most from name resolution that Azure provides

Client-side caching:

Some DNS queries aren't sent across the network. Client-side caching helps reduce latency and improve resilience to network inconsistencies by resolving recurring DNS queries from a local cache. DNS records contain a Time-To-Live (TTL), which enables the cache to store the record for as long as possible without impacting record freshness. As a result, client-side caching is suitable for most situations.

Some Linux distributions don't include caching by default. We recommend that you add a cache to each Linux virtual machine after you check that there isn't a local cache already.

Several different DNS caching packages, such as dnsmasq, are available. Here are the steps to install dnsmasq on the most common distributions:

Ubuntu

1. Install the dnsmasq package:

```Bash
sudo apt-get install dnsmasq
```

2. Enable the dnsmasq service:

```Bash
sudo systemctl enable dnsmasq.service
```

3. Start the dnsmasq service:

```Bash
sudo systemctl start dnsmasq.service
```

Client-side retries

DNS is primarily a UDP protocol. Because the UDP protocol doesn't guarantee message delivery, the DNS protocol itself handles retry logic. Each DNS client (operating system) can exhibit different retry logic depending on the creator's preference:

- Windows operating systems retry after one second and then again after another two, four, and another four seconds.
- The default Linux setup retries after five seconds. You should change this to retry five times at one-second intervals.

To check the current settings on a Linux virtual machine, 'cat /etc/resolv.conf', and look at the 'options' line, for example:

```Bash
sudo cat /etc/resolv.conf
```

```config-conf
options timeout:1 attempts:5
```

The `/etc/resolv.conf` file is auto-generated and shouldn't be edited. The specific steps that add the 'options' line vary by distribution:

Ubuntu (uses resolvconf)

1. Add the options line to `/etc/resolvconf/resolv.conf.d/head` file.
2. Run `sudo resolvconf -u` to update.

SUSE (uses netconf)

1. Add `timeout:1 attempts:5` to the `NETCONFIG_DNS_RESOLVER_OPTIONS=""` parameter in `/etc/sysconfig/network/config`.
2. Run `sudo netconfig update` to update.

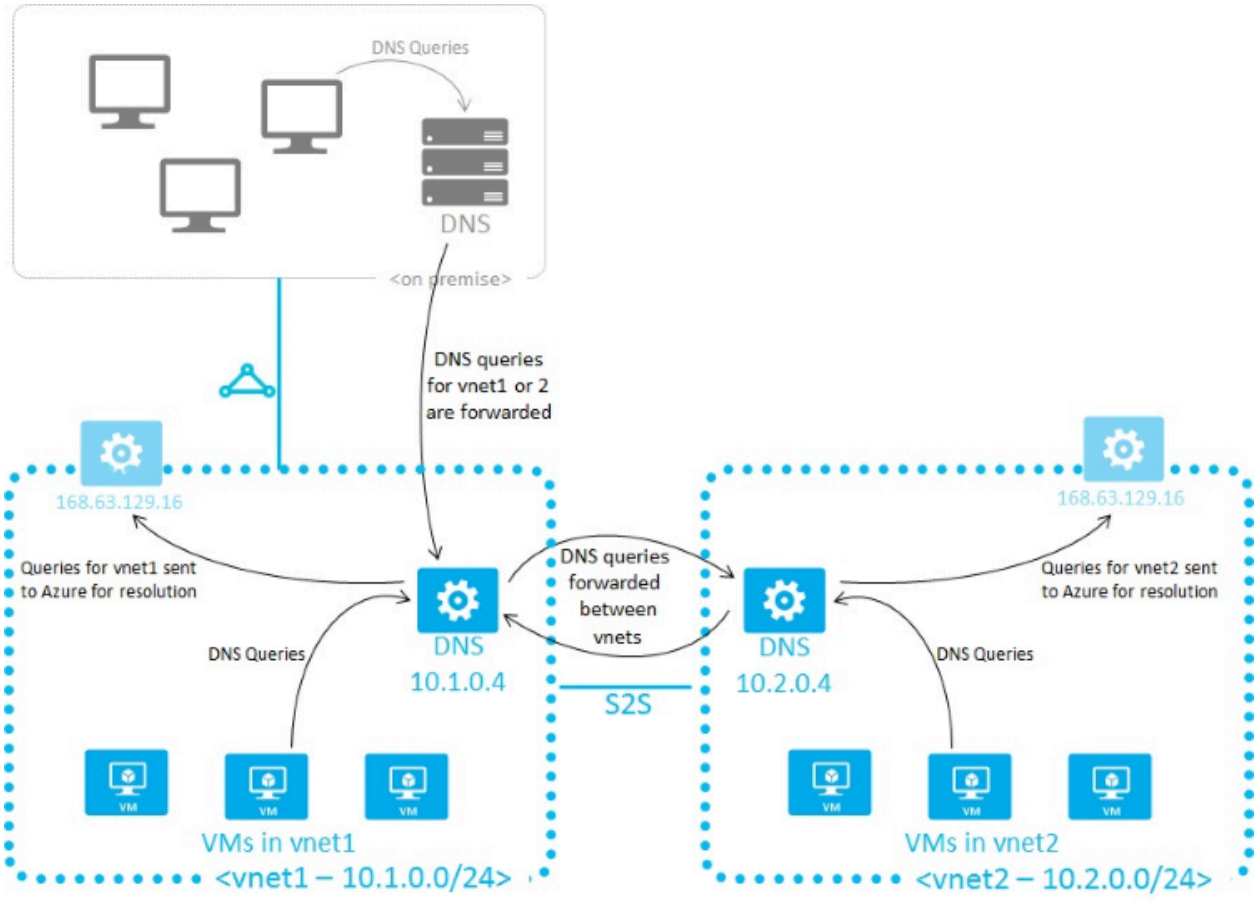
Name resolution using your own DNS server

Your name resolution needs may go beyond the features that Azure provides. For example, you might require DNS resolution between virtual networks. To cover this scenario, you can use your own DNS servers.

DNS servers within a virtual network can forward DNS queries to recursive resolvers of Azure to resolve hostnames that are in the same virtual network. For example, a DNS server that runs in Azure can respond to DNS queries for its own DNS zone files and forward all other queries to Azure. This functionality enables virtual machines to see both your entries in your zone files and hostnames that Azure provides (via the

forwarder). Access to the recursive resolvers of Azure is provided via the virtual IP 168.63.129.16.

DNS forwarding also enables DNS resolution between virtual networks and enables your on-premises machines to resolve hostnames that Azure provides. To resolve a virtual machine's hostname, the DNS server virtual machine must reside in the same virtual network and be configured to forward hostname queries to Azure. Because the DNS suffix is different in each virtual network, you can use conditional forwarding rules to send DNS queries to the correct virtual network for resolution. The following image shows two virtual networks and an on-premises network doing DNS resolution between virtual networks by using this method:



When you use name resolution that Azure provides, the internal DNS suffix is provided to each virtual machine by using DHCP. When you use your own name resolution solution, this suffix isn't supplied to virtual machines because the suffix interferes with other DNS architectures. To refer to machines by FQDN or to configure the suffix on your virtual machines, you can use PowerShell or the API to determine the suffix:

- For virtual networks that are managed by Azure Resource Manager, the suffix is available via the network interface card resource. You can also run the `azure network public-ip show <resource group> <pip name>` command to display the details of your public IP, which includes the FQDN of the NIC.

If forwarding queries to Azure doesn't suit your needs, you need to provide your own DNS solution. Your DNS solution needs to:

- Provide appropriate hostname resolution, for example via DDNS. If you use DDNS, you might need to disable DNS record scavenging. DHCP leases of Azure are long and scavenging may remove DNS records prematurely.
- Provide appropriate recursive resolution to allow resolution of external domain names.
- Be accessible (TCP and UDP on port 53) from the clients it serves and be able to access the Internet.
- Be secured against access from the Internet to mitigate threats posed by external agents.

> **Note**
>
> For best performance, when you use virtual machines in Azure DNS servers, disable IPv6 and assign an **Instance-Level Public IP** to each DNS server virtual machine.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual network interface cards and use internal DNS for VM name resolution on Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This article shows you how to set static internal DNS names for Linux VMs using virtual network interface cards (vNics) and DNS label names with the Azure CLI. Static DNS names are used for permanent infrastructure services like a Jenkins build server, which is used for this document, or a Git server.

The requirements are:

- an Azure account
- SSH public and private key files

Quick commands

If you need to quickly accomplish the task, the following section details the commands needed. More detailed information and context for each step can be found in the rest of the document, starting here. To perform these steps, you need the latest Azure CLI installed and logged in to an Azure account using az login.

Pre-Requirements: Resource Group, virtual network and subnet, Network Security Group with SSH inbound.

Create a virtual network interface card with a static internal DNS name

Create the vNic with az network nic create. The `--internal-dns-name` CLI flag is for setting the DNS label, which provides the static DNS name for the virtual network interface card (vNic). The following example creates a vNic named `myNic`, connects it to the `myVnet` virtual network, and creates an internal DNS name record called `jenkins`:

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic \
```

```
    --vnet-name myVnet \
    --subnet mySubnet \
    --internal-dns-name jenkins
```

Deploy a VM and connect the vNic

Create a VM with az vm create. The `--nics` flag connects the vNic to the VM during the deployment to Azure. The following example creates a VM named `myVM` with Azure Managed Disks and attaches the vNic named `myNic` from the preceding step:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --nics myNic \
    --image Ubuntu2204 \
    --admin-username azureuser \
    --ssh-key-value ~/.ssh/id_rsa.pub
```

Detailed walkthrough

A full continuous integration and continuous deployment (CiCd) infrastructure on Azure requires certain servers to be static or long-lived servers. It's recommended that Azure assets like the virtual networks and Network Security Groups are static and long lived resources that are rarely deployed. Once a virtual network has been deployed, it can be reused in new deployments without any adverse affects to the infrastructure. You can later add a Git repository server or a Jenkins automation server delivers CiCd to this virtual network for your development or test environments.

Internal DNS names are only resolvable inside an Azure virtual network. Because the DNS names are internal, they aren't resolvable to the outside internet, providing extra security to the infrastructure.

In the following examples, replace example parameter names with your own values. Example parameter names include `myResourceGroup`, `myNic`, and `myVM`.

Create the resource group

First, create the resource group with az group create. The following example creates a resource group named `myResourceGroup` in the `westus` location:

Azure CLI

```
az group create --name myResourceGroup --location westus
```

Create the virtual network

The next step is to build a virtual network to launch the VMs into. The virtual network contains one subnet for this walkthrough. For more information on Azure virtual networks, see Create a virtual network.

Create the virtual network with az network vnet create. The following example creates a virtual network named `myVnet` and subnet named `mySubnet`:

Azure CLI

```
az network vnet create \
    --resource-group myResourceGroup \
    --name myVnet \
    --address-prefix 192.168.0.0/16 \
    --subnet-name mySubnet \
    --subnet-prefix 192.168.1.0/24
```

Create the Network Security Group

Azure Network Security Groups are equivalent to a firewall at the network layer. For more information about Network Security Groups, see How to create NSGs in the Azure CLI.

Create the network security group with az network nsg create. The following example creates a network security group named `myNetworkSecurityGroup`:

Azure CLI

```
az network nsg create \
    --resource-group myResourceGroup \
    --name myNetworkSecurityGroup
```

Add an inbound rule to allow SSH

Add an inbound rule for the network security group with az network nsg rule create. The following example creates a rule named `myRuleAllowSSH`:

Azure CLI

```
az network nsg rule create \
    --resource-group myResourceGroup \
    --nsg-name myNetworkSecurityGroup \
    --name myRuleAllowSSH \
    --protocol tcp \
    --direction inbound \
    --priority 1000 \
    --source-address-prefix '*' \
    --source-port-range '*' \
    --destination-address-prefix '*' \
    --destination-port-range 22 \
    --access allow
```

Associate the subnet with the Network Security Group

To associate the subnet with the Network Security Group, use az network vnet subnet update. The following example associates the subnet name `mySubnet` with the Network Security Group named `myNetworkSecurityGroup`:

Azure CLI

```
az network vnet subnet update \
    --resource-group myResourceGroup \
    --vnet-name myVnet \
    --name mySubnet \
    --network-security-group myNetworkSecurityGroup
```

Create the virtual network interface card and static DNS names

To use DNS names for VM name resolution, you need to create virtual network interface cards (vNics) that include a DNS label. vNics are important as you can reuse them by connecting them to different VMs over the infrastructure lifecycle. This approach keeps the vNic as a static resource while the VMs can be temporary. By using DNS labeling on the vNic, we're able to enable simple name resolution from other VMs in the VNet. Using resolvable names enables other VMs to access the automation server by the DNS name `Jenkins` or the Git server as `gitrepo`.

Create the vNic with az network nic create. The following example creates a vNic named `myNic`, connects it to the `myVnet` virtual network named `myVnet`, and creates an internal

DNS name record called `jenkins`:

Azure CLI

```
az network nic create \
    --resource-group myResourceGroup \
    --name myNic \
    --vnet-name myVnet \
    --subnet mySubnet \
    --internal-dns-name jenkins
```

Deploy the VM into the virtual network infrastructure

We now have a virtual network and subnet, a Network Security Group acting as a firewall to protect our subnet by blocking all inbound traffic except port 22 for SSH, and a vNic. You can now deploy a VM inside this existing network infrastructure.

Create a VM with az vm create. The following example creates a VM named `myVM` with Azure Managed Disks and attaches the vNic named `myNic` from the preceding step:

Azure CLI

```
az vm create \
    --resource-group myResourceGroup \
    --name myVM \
    --nics myNic \
    --image Ubuntu2204 \
    --admin-username azureuser \
    --ssh-key-value ~/.ssh/id_rsa.pub
```

By using the CLI flags to call out existing resources, we instruct Azure to deploy the VM inside the existing network. To reiterate, once a VNet and subnet have been deployed, they can be left as static or permanent resources inside your Azure region.

Next steps

- Create your own custom environment for a Linux VM using Azure CLI commands directly
- Create a Linux VM on Azure using templates

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Virtual Machines security overview

This article provides an overview of core Azure security features for virtual machines.

Azure Virtual Machines lets you deploy a wide range of computing solutions in an agile way. The service supports Microsoft Windows, Linux, Microsoft SQL Server, Oracle, IBM, SAP, and Azure BizTalk Services. You can deploy any workload and any language on nearly any operating system.

Azure VMs provide the flexibility of virtualization without buying and maintaining physical hardware. You can build and deploy applications with the assurance that your data is protected in highly secure datacenters.

By using Azure, you can build security-enhanced, compliant solutions that:

- Protect virtual machines from viruses and malware
- Encrypt sensitive data
- Secure network traffic
- Identify and detect threats
- Meet compliance requirements

Trusted launch

Trusted launch is the default for newly created Generation 2 Azure VMs and Virtual Machine Scale Sets. Trusted launch protects against advanced and persistent attack techniques including boot kits, rootkits, and kernel-level malware.

Trusted launch provides:

- **Secure Boot**: Protects against installation of malware-based rootkits and boot kits by ensuring only signed operating systems and drivers can boot
- **vTPM (virtual Trusted Platform Module)**: A dedicated secure vault for keys and measurements that enables attestation and boot integrity verification
- **Boot Integrity Monitoring**: Uses attestation through Microsoft Defender for Cloud to verify boot chain integrity and alert on failures

You can enable trusted launch on existing VMs and Virtual Machine Scale Sets. For more information, see Trusted launch for Azure virtual machines.

Confidential computing

Azure confidential computing protects data while in use through hardware-based trusted execution environments. Confidential VMs use AMD SEV-SNP technology to create a

hardware-enforced boundary between your application and the virtualization stack.

Confidential VMs provide:

- **Hardware-based isolation**: Between virtual machines, hypervisor, and host management code.
- **Confidential OS disk encryption**: Binds disk encryption keys to the VM's TPM, making disk content accessible only to the VM.
- **Secure key release**: Cryptographic binding between platform attestation and VM encryption keys.
- **Attestation**: Customizable policies to ensure host compliance before deployment.

For more information, see Azure confidential VMs.

Virtual machine disk encryption

> **Important**
>
> Azure Disk Encryption is scheduled for retirement on **September 15, 2028**. Until that date, you can continue to use Azure Disk Encryption without disruption. On September 15, 2028, ADE-enabled workloads will continue to run, but encrypted disks will fail to unlock after VM reboots, resulting in service disruption.
>
> Use encryption at host for new VMs. All ADE-enabled VMs (including backups) must migrate to encryption at host before the retirement date to avoid service disruption. See Migrate from Azure Disk Encryption to encryption at host for details.

Azure provides several encryption options for managed disks:

- **Server-side encryption (SSE)**: Also called encryption at rest or Azure Storage encryption, SSE is always enabled and automatically encrypts data on Azure managed disks (OS and data disks) when persisting to storage clusters. Data is encrypted transparently using 256-bit AES encryption, which is FIPS 140-2 compliant. SSE doesn't affect disk performance and has no extra cost. However, SSE doesn't encrypt temp disks or disk caches.

- **Encryption at host**: Enhances SSE by encrypting temp disks and disk caches at rest, with data flowing encrypted to storage clusters. This encryption provides end-to-end encryption for your VM data. Encryption at host doesn't use your VM's CPU and doesn't affect performance. Temp disks and ephemeral OS disks are encrypted with platform-managed keys. OS and data disk caches are encrypted with either customer-managed or platform-managed keys, depending on the disk encryption type.

- **Confidential disk encryption**: For confidential VMs, this encryption binds disk encryption keys to the VM's TPM, making protected disk content accessible only to the VM.

By default, managed disks use platform-managed keys with no additional configuration required. For customer-managed keys, you can use Azure Key Vault, Azure Key Vault Managed HSM, or Azure Cloud HSM to control and manage your own encryption keys.

For more information, see Overview of managed disk encryption options and Server-side encryption of Azure Disk Storage.

Key management

Azure Key Vault provides secure storage for cryptographic keys, secrets, and certificates. Azure offers multiple key management solutions with different security levels:

- **Azure Key Vault Standard**: FIPS 140-2 Level 1 validated, multitenant service with software-protected keys.
- **Azure Key Vault Premium**: FIPS 140-3 Level 3 validated, multitenant service with HSM-protected keys.
- **Azure Key Vault Managed HSM**: FIPS 140-3 Level 3 validated, single-tenant service that provides full customer control over the HSM and key sovereignty.

For virtual machines, these services can store:

- **Disk encryption keys**: Customer-managed keys for encrypting managed disks through disk encryption sets.
- **SQL Server encryption keys**: Keys for transparent data encryption on SQL Server VMs.
- **Application secrets**: Keys and secrets used by applications running on your VMs.

You manage permissions and access to protected items through Microsoft Entra ID. You can audit key usage and retain full control over your encryption keys.

For more information, see What is Azure Key Vault?, What is Azure Key Vault Managed HSM?, and Key management in Azure.

Virtual machine backup

Azure Backup is a scalable solution that protects your VM data with zero capital investment and minimal operating costs. Application errors can corrupt your data, and human errors can introduce bugs. With Azure Backup, your virtual machines running Windows and Linux are protected.

Azure Backup provides:

- **Independent and isolated backups**: Backups are stored in a Recovery Services vault, which provides built-in management of recovery points and guards against accidental destruction of data
- **Application-consistent recovery points**: Captures the state of the application data at the time of backup for running VMs
- **No explicit outbound connectivity required**: All communication and data transfer happens on the Azure backbone network
- **Soft delete protection**: Deleted backup data is retained for 14 additional days, allowing recovery with no data loss
- **Cross Region Restore**: Restore Azure VMs in a secondary Azure paired region for disaster recovery scenarios

For more information, see What is Azure Backup? and Azure Backup service FAQ.

Azure Site Recovery

Azure Site Recovery helps orchestrate replication, failover, and recovery of workloads and apps so they're available from a secondary location if your primary location goes down.

Site Recovery:

- **Simplifies BCDR strategy**: Makes it easy to handle replication, failover, and recovery of multiple business workloads and apps from a single location
- **Provides flexible replication**: Replicate workloads running on Hyper-V VMs, VMware VMs, and Windows/Linux physical servers
- **Supports failover and recovery**: Provides test failovers for disaster recovery drills without affecting production environments
- **Eliminates secondary datacenters**: Replicate to Azure, eliminating the cost and complexity of maintaining a secondary site

For more information, see What is Azure Site Recovery?, How does Azure Site Recovery work?, and What workloads are protected by Azure Site Recovery?.

Virtual networking

Virtual machines require network connectivity. Azure requires virtual machines to be connected to an Azure virtual network.

An Azure virtual network is a logical construct built on top of the physical Azure network fabric. Each logical Azure virtual network is isolated from all other Azure virtual networks. This isolation helps ensure that network traffic in your deployments is not accessible to other Microsoft Azure customers.

For more information, see Azure network security overview and Virtual Network overview.

Security policy management

Microsoft Defender for Cloud helps you prevent, detect, and respond to threats. Defender for Cloud gives you increased visibility into, and control over, the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions.

Defender for Cloud helps you optimize and monitor VM security by:

- Providing security recommendations for virtual machines
- Monitoring the state of your virtual machines
- Providing Microsoft Defender for Servers with advanced threat protection

Microsoft Defender for Servers includes:

- Microsoft Defender for Endpoint integration for endpoint detection and response
- Vulnerability assessment for identifying security weaknesses
- Just-in-time VM access to reduce attack surface
- File integrity monitoring to detect changes to critical files
- Adaptive application controls for approved applications

For more information, see Introduction to Microsoft Defender for Cloud, Microsoft Defender for Servers, and Microsoft Defender for Cloud frequently asked questions.

Compliance

Azure Virtual Machines is certified for FISMA, FedRAMP, HIPAA, PCI DSS Level 1, and other key compliance programs. This certification makes it easier for your Azure applications to meet compliance requirements and for your business to address domestic and international regulatory requirements.

For more information, see Microsoft Trust Center: Compliance and Azure compliance documentation.

Next steps

Learn about security best practices for VMs and operating systems.

Last updated on 01/16/2026

Apply Zero Trust principles to virtual machines in Azure

Article • 05/20/2025

Summary: To apply Zero Trust principles to Azure virtual machines, you must configure logical isolation with dedicated resource groups, leverage Role Based Access Control (RBAC), secure virtual machine boot components, enable customer-managed keys and double encryption, control installed applications, configure secure access and maintenance of virtual machines, and enable advanced threat detection and protection.

This article provides steps to apply the principles of Zero Trust to virtual machines in Azure:

Expand table

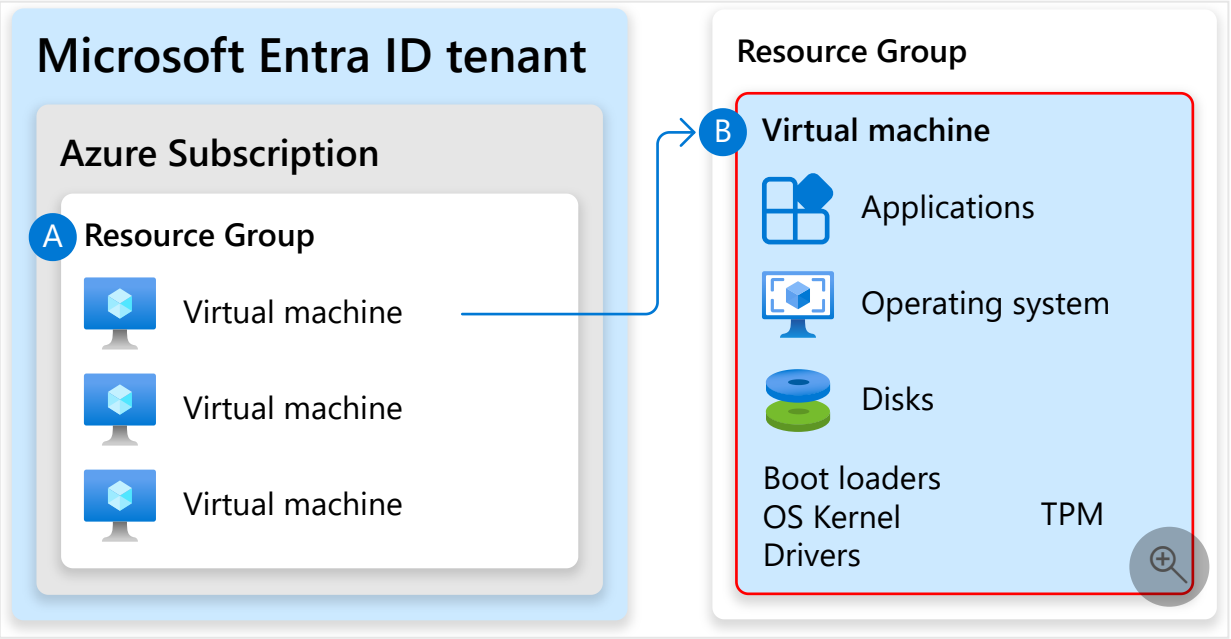
Zero Trust principle	Definition	Met by
Verify explicitly	Always authenticate and authorize based on all available data points.	Use secure access.
Use least privileged access	Limit user access with Just-In-Time and Just-Enough-Access (JIT/JEA), risk-based adaptive policies, and data protection.	Leverage Role Based Access Control (RBAC) and control the applications running on virtual machines.
Assume breach	Minimize blast radius and segment access. Verify end-to-end encryption and use analytics to get visibility, drive threat detection, and improve defenses.	Isolate virtual machines with resource groups, secure their components, use double encryption, and enable advanced threat detection and protection.

This article is part of a series of articles that demonstrate how to apply the principles of Zero Trust across an environment in Azure that includes a spoke virtual network (VNet) hosting a virtual machine-based workload. For an overview, see Apply Zero Trust principles to Azure infrastructure.

Logical architecture for virtual machines

Zero Trust principles for virtual machines are applied across the logical architecture, from the tenant and directory level down to the data and application layer within each virtual machine.

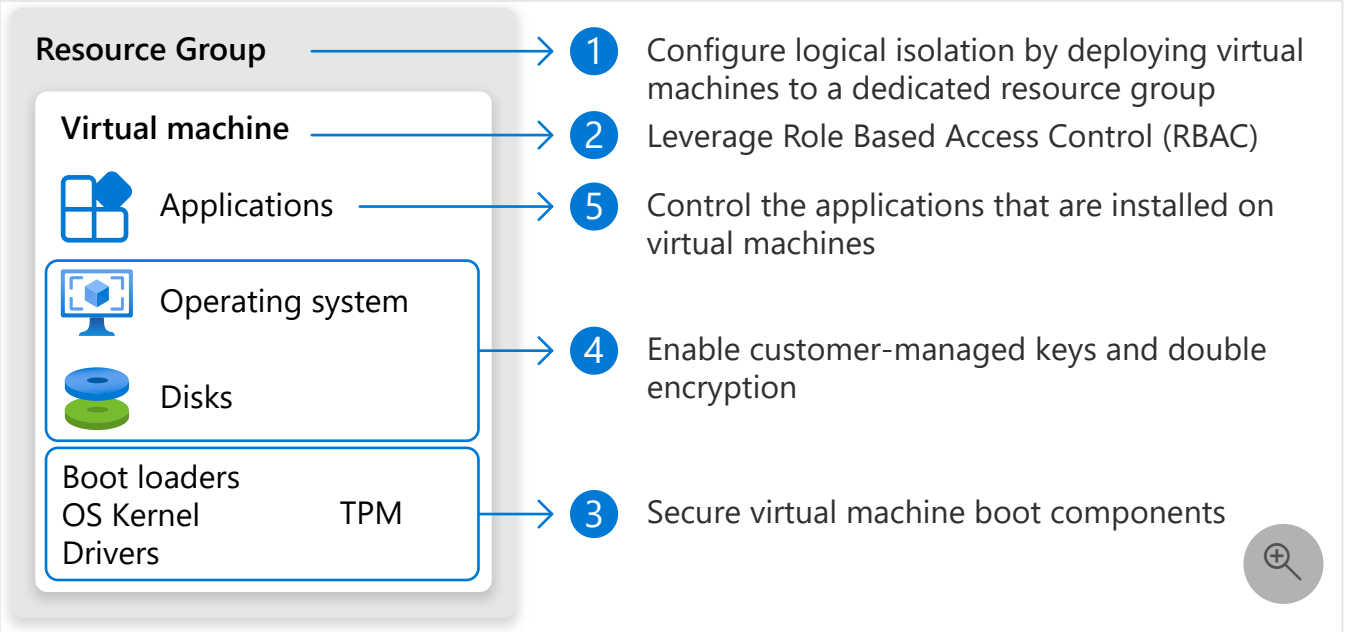
The following diagram shows the logical architecture components.



In this diagram:

- **A** is a set of virtual machines isolated within a dedicated resource group that resides within an Azure subscription.
- **B** is the logical architecture for a single virtual machine with the following components called out: applications, operating system, disks, boot loaders, OS Kernel, drivers, and the Trusted Platform Module (TPM) component.

This article walks through the steps to apply the principles of Zero Trust across this logical architecture, using these steps.



Expand table

Step	Task	Zero Trust principles applied
1	Configure logical isolation by deploying virtual machines to a dedicated resource group.	Assume breach
2	Leverage Role Based Access Control (RBAC).	Verify explicitly Use least privileged access
3	Secure virtual machine boot components including boot loaders, OS kernels, and drivers. Securely protect keys, certificates, and secrets in the Trusted Platform Module (TPM).	Assume breach
4	Enable customer-managed keys and double encryption.	Assume breach
5	Control the applications that are installed on virtual machines.	Use least privileged access
6	Configure secure access (not shown on the logical architecture figure).	Verify explicitly Use least privileged access Assume breach
7	Set up secure maintenance of virtual machines (not shown on the logical architecture figure).	Assume breach
8	Enable advanced threat detection and protection (not shown on the logical architecture figure).	Assume breach

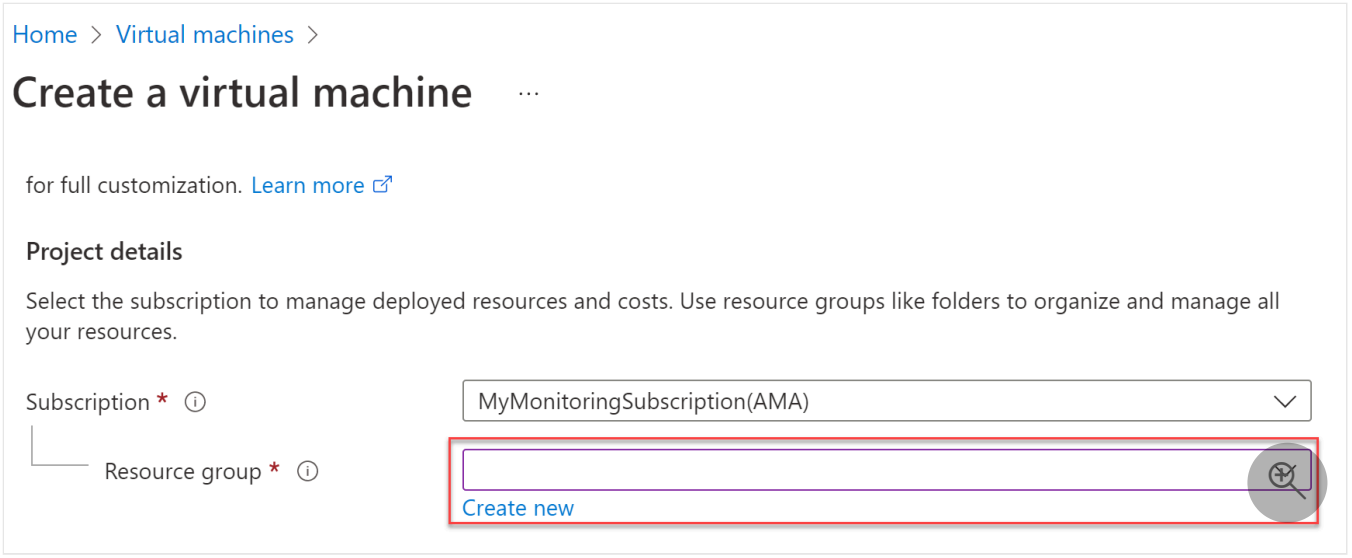
Step 1: Configure logical isolation for virtual machines

Begin by isolating virtual machines within a dedicated resource group. You can isolate virtual machines into different resource groups based on purpose, data classification, and governance requirements, such as the need to control permissions and monitoring.

Using dedicated resource groups allows you to set policies and permissions that apply to all the virtual machines within the resource group. You can then use role based access control (RBAC) to create least privileged access to the Azure resources contained in the resource group.

For more information on creating and managing resource groups, see Manage Azure resource groups by using the Azure portal.

You assign a virtual machine to a resource group when you first create the virtual machine, as shown here.



Step 2: Leverage Role Based Access Control (RBAC)

Zero Trust requires configuring least privileged access. To do so, you need to limit user access with just-in-time and just-enough access (JIT/JEA) based on their role, workload, and data classification.

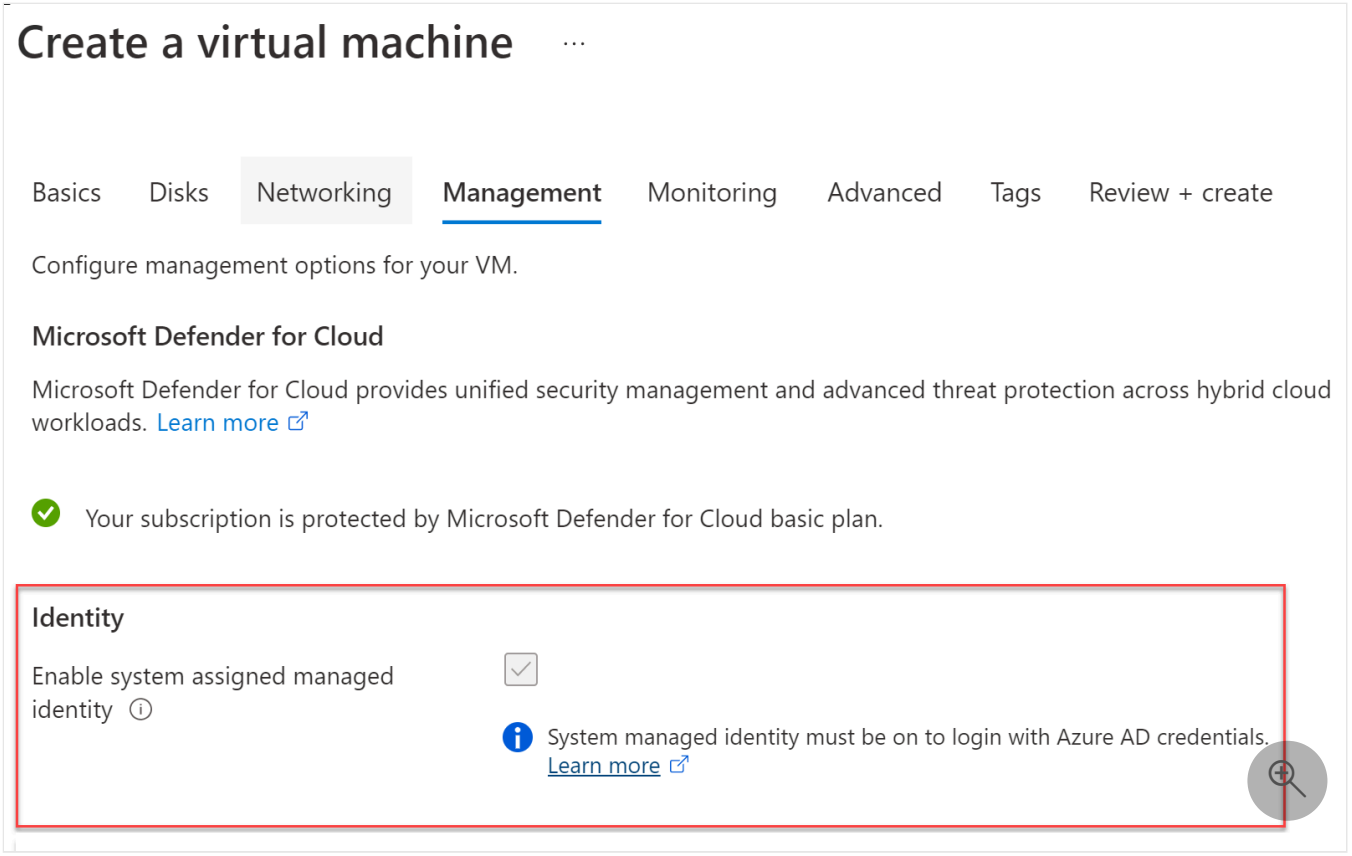
The following built-in roles are commonly used for virtual machine access:

- **Virtual Machine User Login:** View virtual machines in the portal and sign-in as a regular user.
- **Virtual Machine Administration Login:** View virtual machines in the portal and sign-in to virtual machines as an Administrator.
- **Virtual Machine Contributor:** Create and manage virtual machines, including reset root user's password and managed disks. Doesn't grant access to the management virtual network (VNet) or the ability to assign permissions to the resources.

To join a virtual machine to a VNet, you can use the custom permission **Microsoft.Network/virtualNetworks/subnets/join/action** to make a custom role.

When this custom role is used with Managed Identity and Conditional Access Policy, you can use device state, data classification, anomalies, location, and identity to force multifactor authentication and granularly allow access based on verified trust.

To extend your realm of control beyond the system and allow your Microsoft Entra ID tenant with Microsoft Intelligent Security Graph to support secure access, go to the **Management** blade of the virtual machine and turn on **System Assigned Managed Identity**, as shown here.



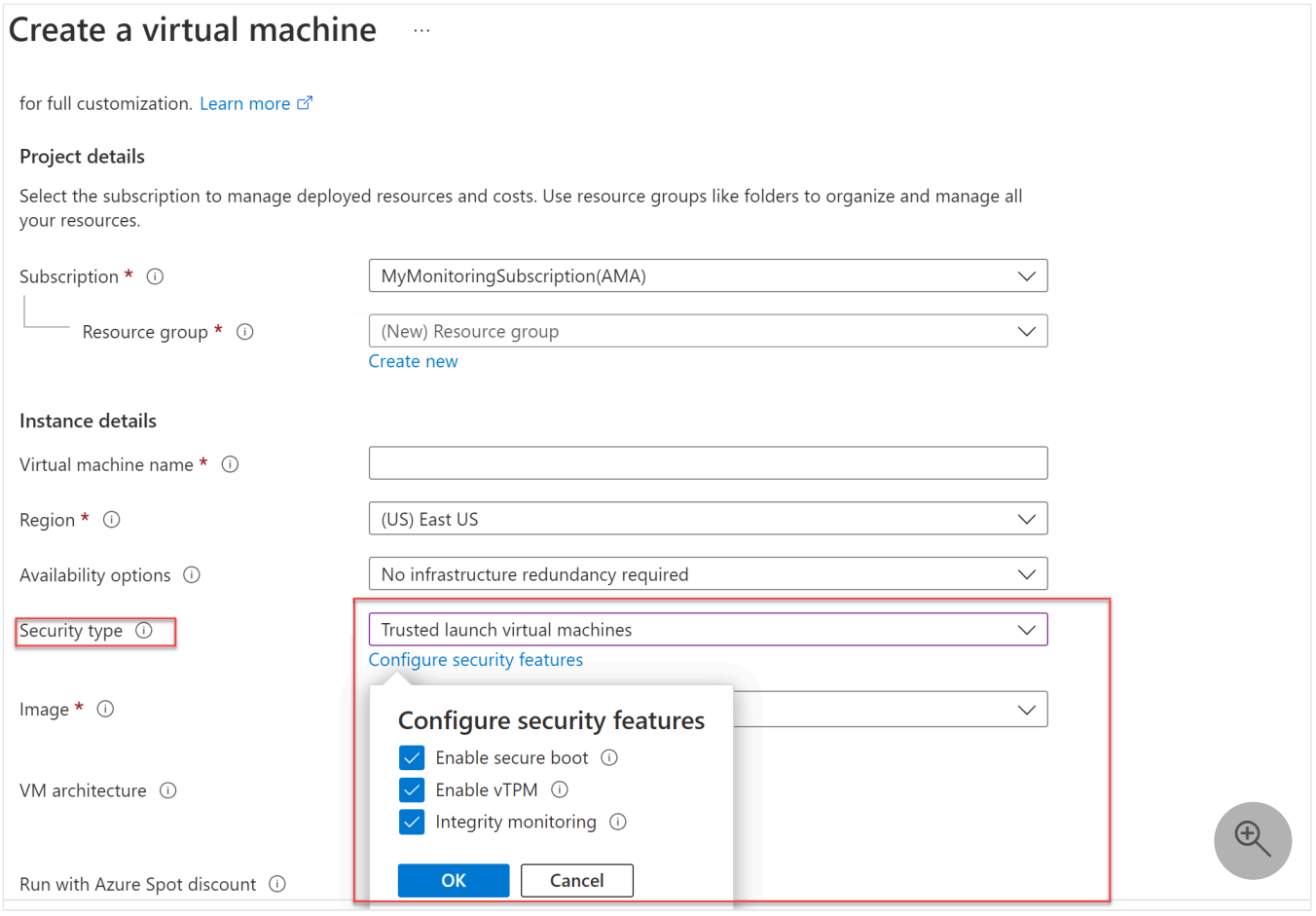
> **Note**
>
> This feature is only available for Azure Virtual Desktop, Windows Server 2019, Windows 10, and Linux Distros using certificate-based access.

Step 3: Secure virtual machine boot components

Follow these steps:

- When you create the virtual machine, be sure you configure security for the boot components. Enhanced deployment of virtual machines allows you to select security type and use Secure boot and vTPM.
- Securely deploy virtual machines with verified boot loaders, OS kernels, and drivers that are signed by trusted publishers to establish a "root ." If the image isn't signed by a trusted publisher, the virtual machine won't boot.
- Securely protect keys, certificates, and secrets in the virtual machines in a Trusted Platform Module.
- Gain insights and confidence of the entire boot chain's integrity.
- Ensure workloads are trusted and verifiable. The vTPM enables attestation by measuring the entire boot chain of your virtual machine (UEFI, OS, system, and drivers).

Enhanced deployment of virtual machines allows you to select security type and use secure boot and vTPM when you create them, as shown here.



Step 4: Enable customer-managed keys and double encryption

Using customer-managed keys and double encryption ensures that if a disk is exported, it isn't readable or able to function. By ensuring that the keys are privately held and disks are double encrypted, you protect against breaches that attempt to extract disk information.

For information on how to configure a customer-managed encryption key with Azure Key Vault, see Use the Azure portal to enable server-side encryption with customer-managed keys for managed disks. There's an additional cost for using Azure Key Vault.

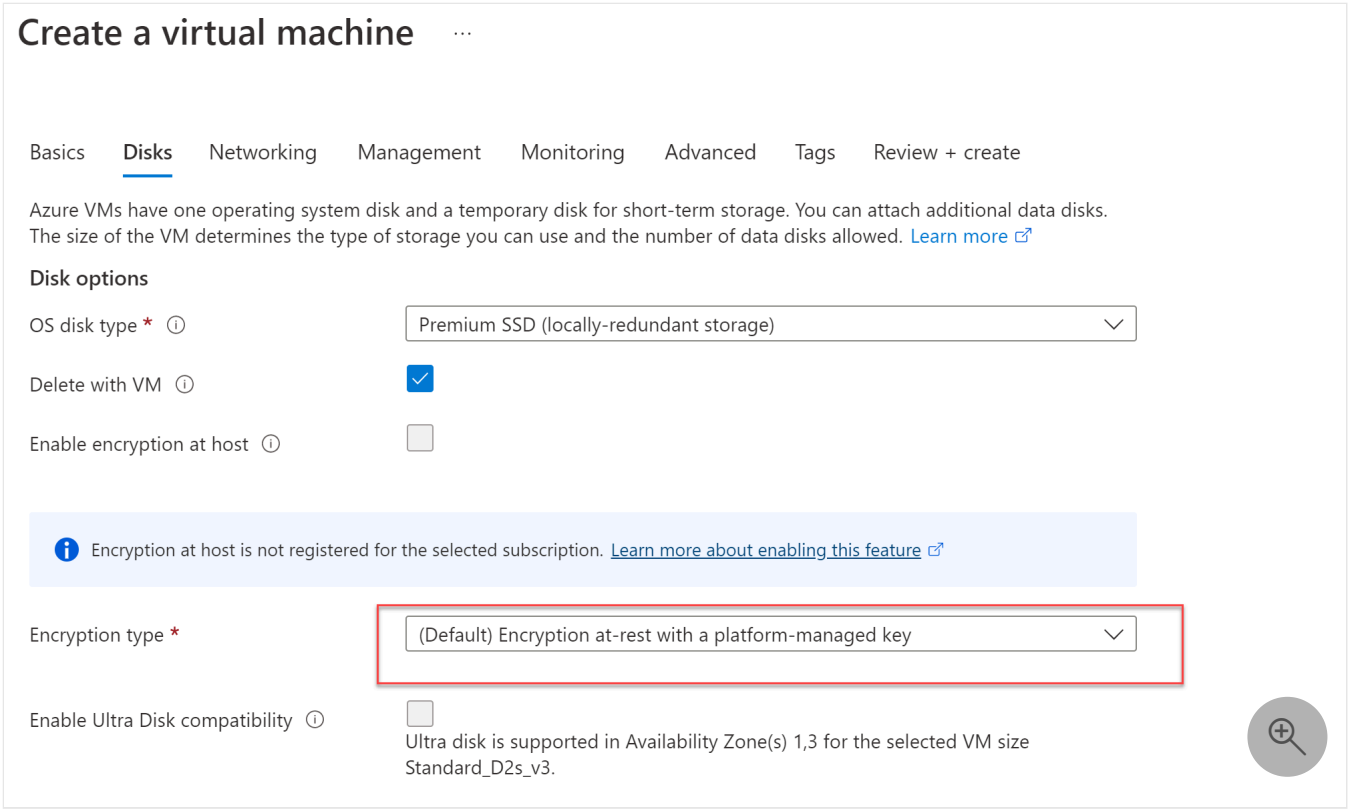
Enable server-side encryption of Azure Disk Storage for:

- FIPS 140-2 compliant transparent encryption with AES 256 encryption.
- Greater flexibility to manage controls.
- Hardware (HSM) or software-defined encryption.

Enable server-side encryption at the host for end-to-end encryption of your virtual machine data.

After completing these procedures, you use your customer-managed encryption key to encrypt the disks within your virtual machine.

You select the encryption type on the **Disks** blade for the virtual machine configuration. For **Encryption type**, select **Double encryption with platform-managed and customer-managed keys**, as shown here.



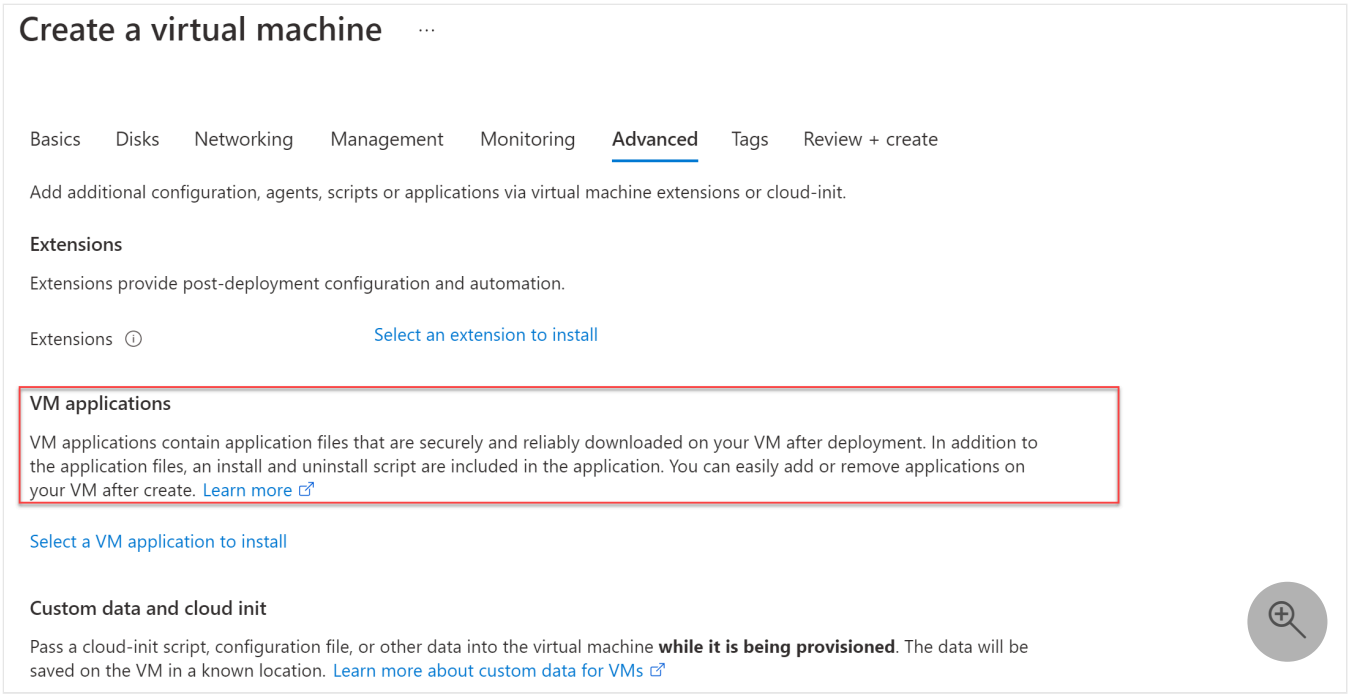
Step 5: Control the applications installed on virtual machines

It's important to control the applications that are installed on your virtual machines:

- Browser extensions (APIs) are difficult to secure which can lead to malicious URL delivery.
- Unsanctioned apps can go unpatched as they're shadow IT objects (the IT teams aren't prepared or have no knowledge that these are installed).

You can use the Virtual Machine Applications feature to control the applications that are installed on virtual machines. With this feature, you select which virtual machine applications to install. This feature uses the Azure Compute Gallery to simplify management of applications for virtual machines. When used together with RBAC, you can ensure that only trusted applications are available for users.

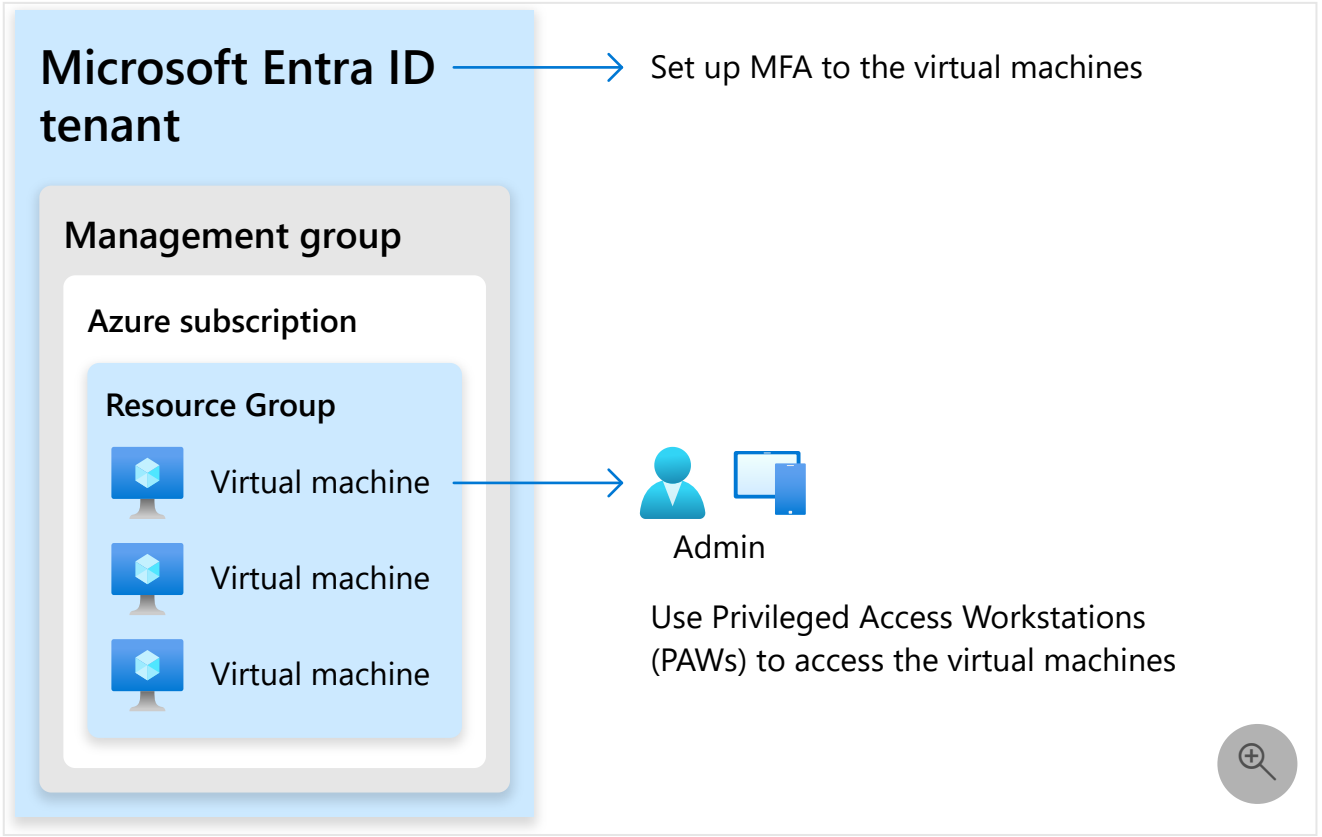
You select the virtual machine applications on the **Advanced** blade for the virtual machine configuration, as show here.



Step 6: Configure secure access

To configure secure access:

- Configure secure communication within the Azure environment between components that are accessing virtual machines directly
- Set up multifactor authentication with conditional access
- Use privileged access workstations (PAWs)



In the diagram:

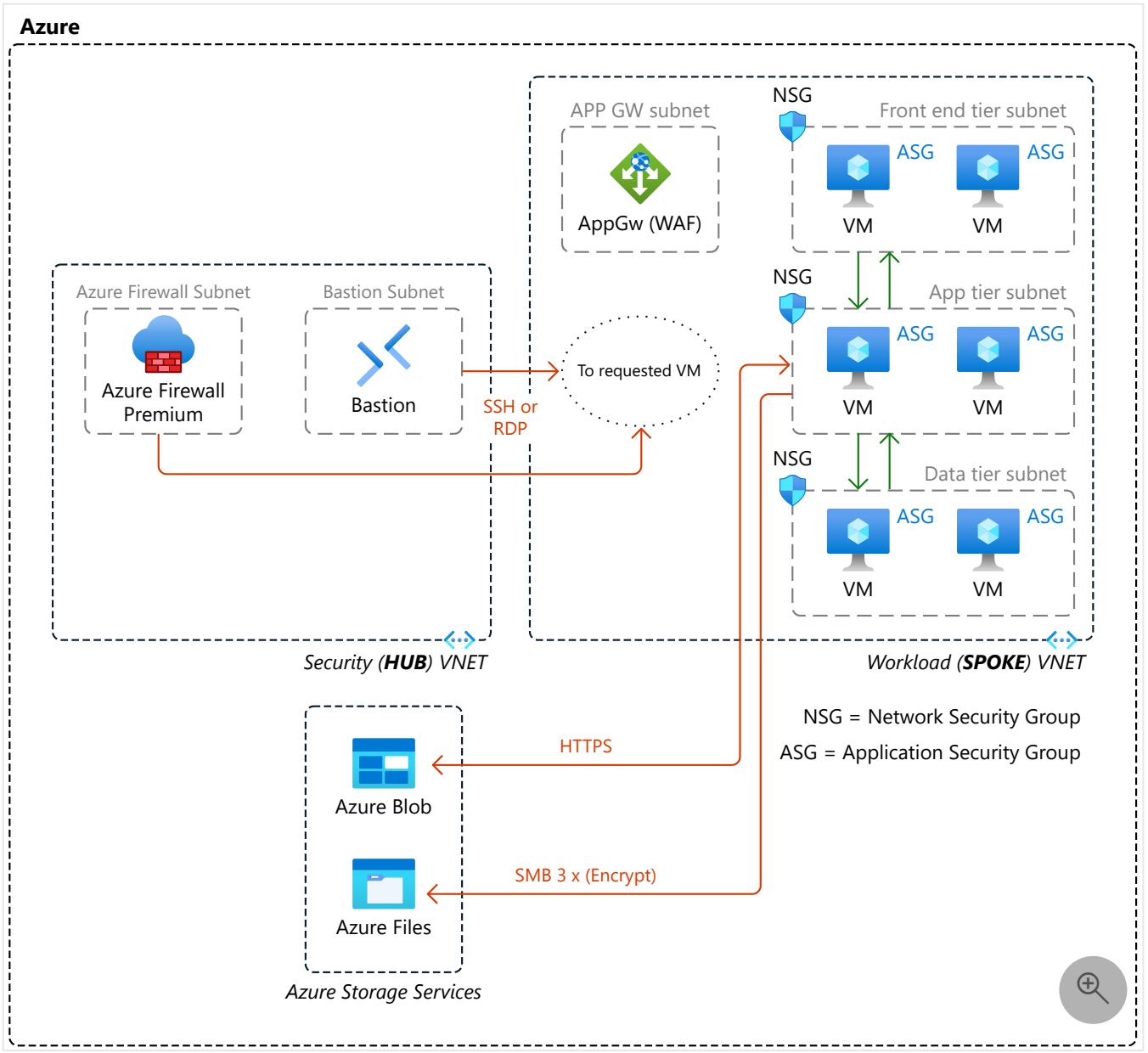
- multifactor authentication with conditional access is set up within Microsoft Entra ID and related portals.
- Admins use privileged access workstations (PAWs) to access virtual machines directly.

Configure secure communication within the Azure environment for virtual machines

First, be sure that communication between the components in the Azure environment is secure.

In the reference architecture, Azure Bastion provides secure connections to virtual machines. Azure Bastion acts as an RDP/SSH broker and doesn't interact with the RDP protocol of your physical system. This also enables you to reduce the number of public-facing IP addresses.

The following diagram shows the components of secure communications for virtual machines.

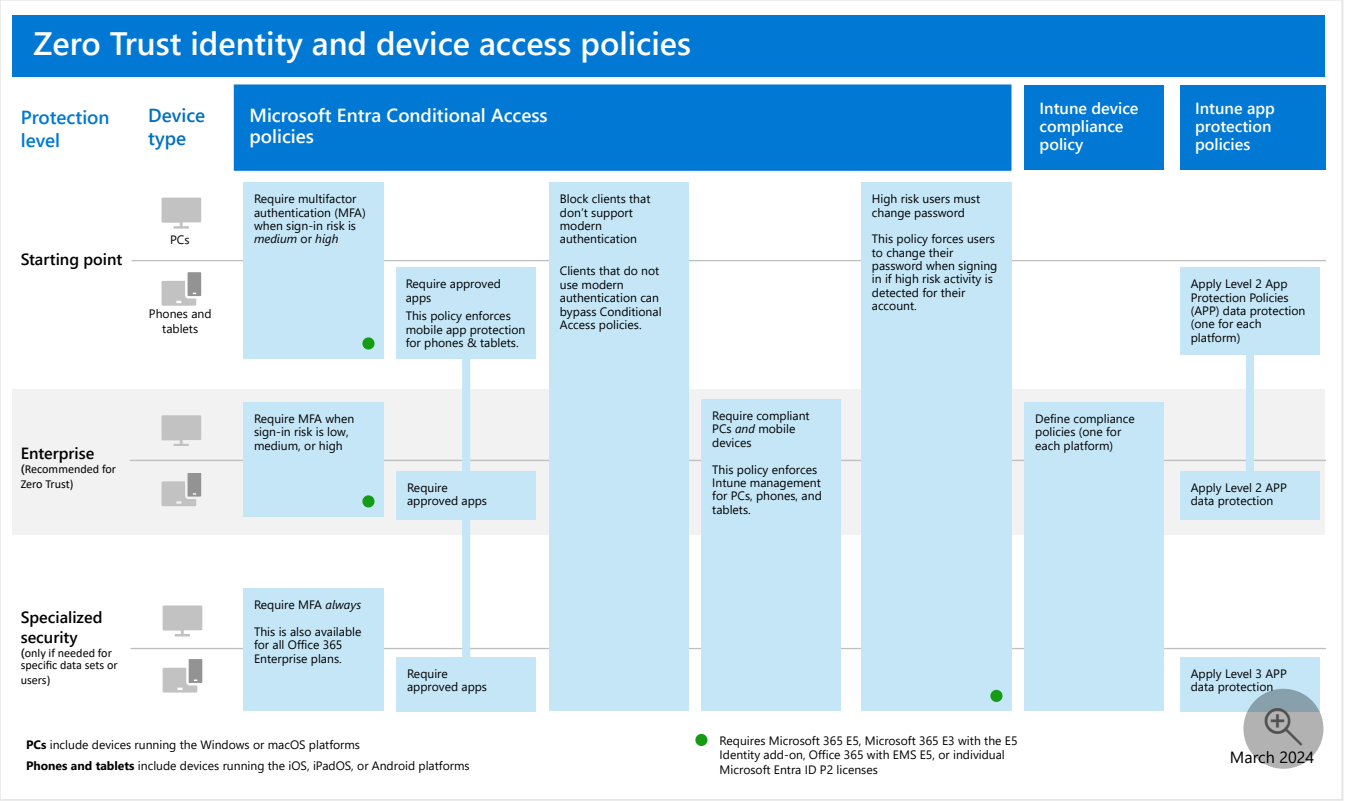


Set up multifactor authentication with conditional access

In Step 2. Leverage Role Based Access Control, you configured Microsoft Entra integration and managed identity. This allows you to set up Azure multifactor authentication for Azure Virtual Desktop or for servers running Windows Server 2019 or newer. You can also Log in to a Linux VM with Microsoft Entra credentials. The added benefit of this is the machine that connects to the virtual machine must also be registered to your Microsoft Entra ID tenant to be allowed to connect.

When configuring multifactor authentication with conditional access and related policies, use the recommended policy set for Zero Trust as a guide. This includes **Starting point** policies that don't require managing devices. Ideally, the devices accessing your virtual machines are managed and you can implement the **Enterprise** policies, which is recommended for Zero Trust. For more information, see Common Zero Trust identity and device access policies.

The following diagram shows the recommended policies for Zero Trust.



Remember that usernames and passwords can be 100% compromised. Using multifactor authentication, you reduce your risk of compromise by 99.9%. This requires Microsoft Entra ID P1 licenses.

> **Note**
>
> You can use VPNs used to connect to virtual machines in Azure as well. However, you should be sure to use methods to verify explicitly. Creating a tunnel that is "trusted" regardless of how they are used can be riskier than having specific connections that are highly verified.

No amount of security at the Network, Transport, or Application layers matters if you aren't coming from a trusted, verified, and secure source.

Use PAWs

Use Privileged Access Workstations (PAWs) to ensure devices that access virtual machines are healthy. PAWs are configured specifically for privileged access so that admins use a device that has:

- Security controls and policies that restrict local administrative access.
- Productivity tools to minimize the attack surface to only what's absolutely required for performing sensitive administrative tasks.

For more information on deployment options, see Privileged access deployment.

Step 7: Set up secure maintenance of virtual machines

Secure maintenance of virtual machines includes:

- Using anti-malware
- Automating virtual machine updates

Use anti-malware on virtual machines

Anti-malware helps protect your virtual machine from threats such as malicious files and adware, etc. You can use anti-malware software from an option of vendors such as Microsoft, Symantec, Trend Micro, and Kaspersky.

Microsoft Antimalware is a no-cost resource that provides real-time protection capability to assist in detection, quarantining and eradicating malicious software, spyware, and viruses:

- Runs in the background with the need of user interaction
- Provides alerts when unwanted or malicious software is downloaded, installed, or run
- Offers secure-by-default configuration and anti-malware monitoring
- Scheduled scanning
- Signature updates
- Antimalware Engine and Platform updates
- Active Protection
- Samples reporting
- Exclusions
- Antimalware event collection

Automate virtual machine updates

Automating updates to systems ensures they are protected from the latest malware and misconfiguration exploits. There's automatic updating with aid in the trusted platform verification process.

Concentrate on Azure Virtual Machine Maintenance and Updates to ensure your systems are hardened against configuration insecurities:

- Azure Automation Update Management can assist in the management of your update process. With this utility, you can check the update status of your systems, manage,

schedule, and reboot servers.
- The Azure Virtual Machine Agent is used to manage your virtual machines and gives you the ability to use extensions for management.

Operating systems supported by Update Management include the following:

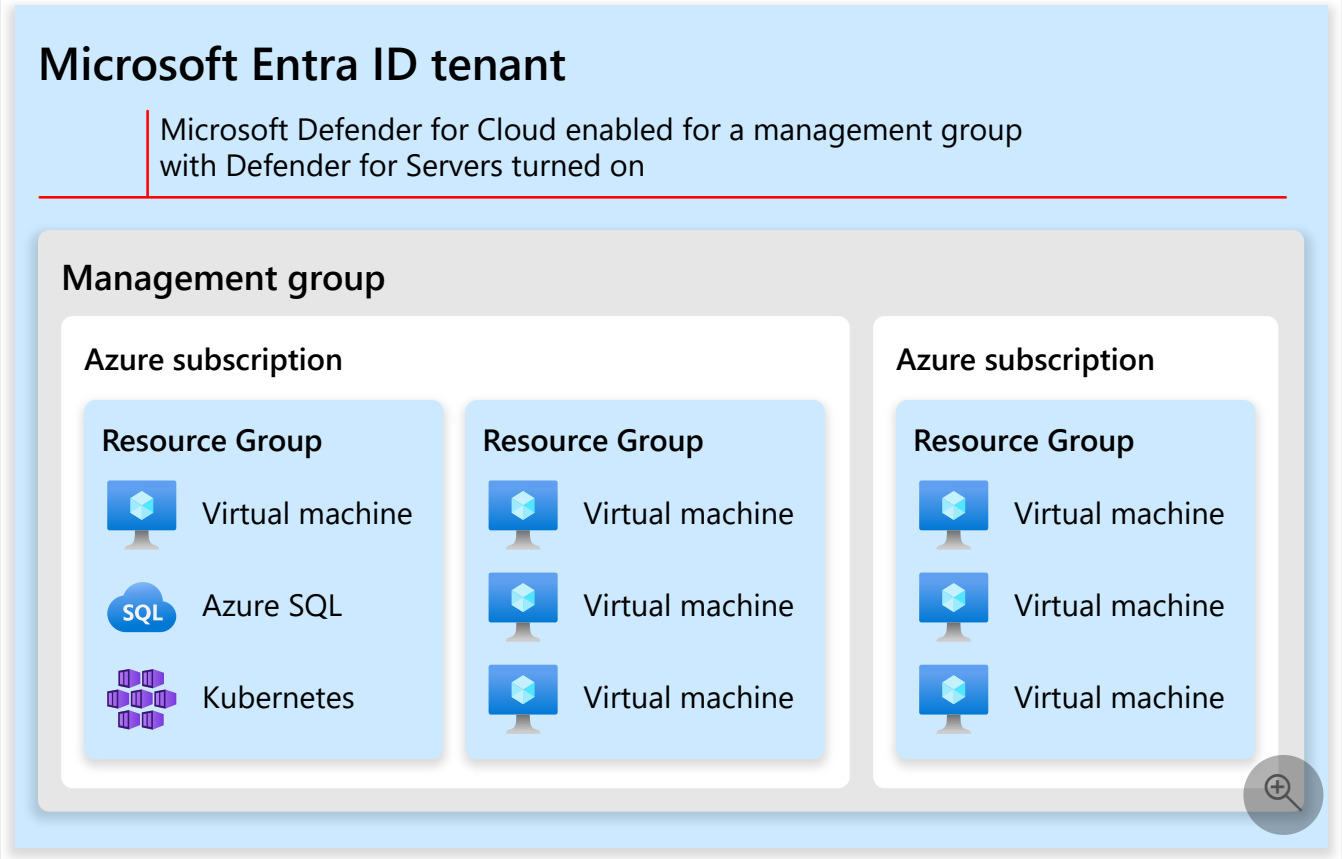
- Each Windows virtual machine - Update Management does a scan twice a day for each machine.
- Each Linux virtual machine - Update Management does a scan every hour.

See this additional guidance:

- Plan deployment for updating Windows VMs in Azure
- Use Azure Private Link to securely connect networks to Azure Automation Ensures virtual machines connect in an isolated controlled fashion and not over the internet for updates.

Step 8: Enable advanced threat detection and protection

Threat protection for Azure infrastructure is provided by Microsoft Defender for Cloud. This protection is extended to virtual machines when you provision Microsoft Defender for Servers, as shown in the following diagram.



In the diagram:

- As described in the Apply Zero Trust principles to Azure IaaS overview article, Defender for Cloud is enabled at the level of an Azure subscription or at the level of an Azure management group that includes multiple Azure subscriptions.
- In addition to enabling Defender for Cloud, Defender for Servers is provisioned.

Advanced threat protection verifies the activities occurring on virtual machines based on Microsoft's threat intelligence. It looks for specific configurations and activities that suggest that there could be a breach. It enables the **Verify explicitly** and **Assume breach** Zero Trust principles.

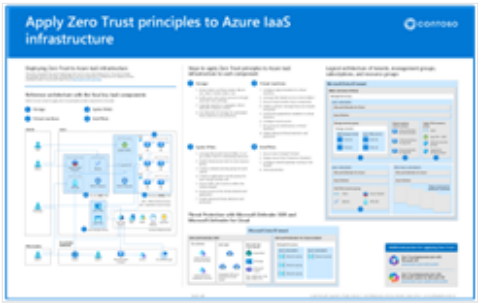
Microsoft Defender for Servers includes the following:

- Access to the Microsoft Defender for Endpoint data that is related to vulnerabilities, installed software, and alerts for your endpoints for endpoint detection and response (EDR).
- Defender for Cloud's integrated vulnerability assessment scanner for servers.
- Discover vulnerabilities and misconfigurations in real time with Microsoft Defender for Endpoint, and without the need of other agents or periodic scans.
- Defender for Cloud's integrated Qualys scanner for Azure and hybrid machines allows you to use a leading tool in real-time vulnerability identification without the need of a Qualys license.
- Implement Just-in-time virtual machine access in Defender for Cloud. This creates an explicit deny rule for RDP/SSH and gives you JIT access at the server level when you need it and allows you to limit the period of access.
- File integrity monitoring in Defender for Cloud provides you to change monitoring of files and registries of the operations system, application software, and other changes that allow you to validate the integrity of your file systems.
- Adaptive application controls in Defender for Cloud provides an automated solution for creating and defining allow list for known safe applications and generates security alerts if a new application runs other than those you define as safe for use.
- Adaptive network hardening in Defender for Cloud uses machine learning algorithms that calculate your current traffic, threat intelligence, indicators of compromise, and known trusted configurations to provide recommendations for hardening your Network Security Groups.

Technical illustrations

These illustrations are replicas of the reference illustrations in these articles. Download and customize these for your own organization and customers. Replace the Contoso logo with your own.

Expand table

Item	Description
 Download Visio Updated October 2024	**Apply Zero Trust principles to Azure IaaS** Use these illustrations with these articles: - Overview - Azure storage - Virtual machines - Azure spoke virtual networks - Azure hub virtual networks
 Download Visio Updated October 2024	**Apply Zero Trust principles to Azure IaaS — One page poster** A one-page overview of the process for applying the principles of Zero Trust to Azure IaaS environments.

For additional technical illustrations, see Zero Trust illustrations for IT architects and implementers.

Recommended training

Secure your Azure virtual machine disks

Expand table

Training	**Secure your Azure virtual machine disks**
	Learn how to use Azure Disk Encryption (ADE) to encrypt OS and data disks on existing and new virtual machines. In this module, you learn how to: • Determine which encryption method is best for your virtual machine. • Encrypt existing virtual machine disks using the Azure portal. • Encrypt existing virtual machine disks using PowerShell. • Modify Azure Resource Manager templates to automate disk encryption on new virtual machines.

Start >

For more training on Azure, see the entire Microsoft catalog:
Browse all - Training | Microsoft Learn

Implement virtual machine host security in Azure

Expand table

Training	**Implement virtual machine host security in Azure**
	In this learning path, learn how to protect and harden your virtual machines in Azure.

Start >

For more training on virtual machines in Azure, see these resources in the Microsoft catalog:
Virtual machines in Azure | Microsoft Learn

Next Steps

See these additional articles for applying Zero Trust principles to Azure:

- Azure IaaS overview
  - Azure storage
  - Spoke virtual networks
  - Spoke virtual networks with Azure PaaS services
  - Hub virtual networks
- Azure Virtual Desktop
- Azure Virtual WAN
- IaaS applications in Amazon Web Services
- Microsoft Sentinel and Microsoft Defender XDR

References

- Manage Azure resource groups with the Azure portal
- Secure boot
- Overview of vTPM
- Attestation
- Enable server-side encryption of Azure Disk Storage

- AES 256 encryption
- Azure Bastion
- Azure multifactor authentication for Azure Virtual Desktop
- Windows Servers 2019 or newer
- Log in to a Linux VM with Microsoft Entra credentials
- Common Zero Trust identity and device access policies
- Privileged Access Workstations (PAW)
- Privileged access deployment
- Microsoft Anti-malware
- Virtual Machine Agent
- Plan deployment for updating Windows VMs in Azure - Azure Example Scenarios
- Use Azure Private Link to securely connect networks to Azure Automation
- Microsoft Defender for Servers
- Microsoft Defender for Endpoint
- Defender for Cloud's integrated vulnerability assessment

What is Microsoft Defender for Cloud?

> **ⓘ Important**
>
> Microsoft Defender for Cloud is expanding to the Defender portal to provide a unified security experience across cloud and code environments. As part of this expansion, some features are now available in the Microsoft Defender Portal, and additional capabilities will be added to the Defender portal over time.
>
> This change is designed to:
>
> - Unlock new cloud and posture management experiences.
> - Provide deep integration with other Microsoft security services.
> - Empower security teams with streamlined workflows by bringing all tools together in one portal.
>
> To identify documentation specific for the Defender Portal, look for the portal entry point at the top of the article. This pivot indicates whether the content applies to the Defender portal or the Azure portal.
>
> Our documentation will be continuously updated to reflect these changes, so check back regularly for the latest guidance and feature availability.
>
> Review the **known limitations** article to understand current gaps during the preview phase.

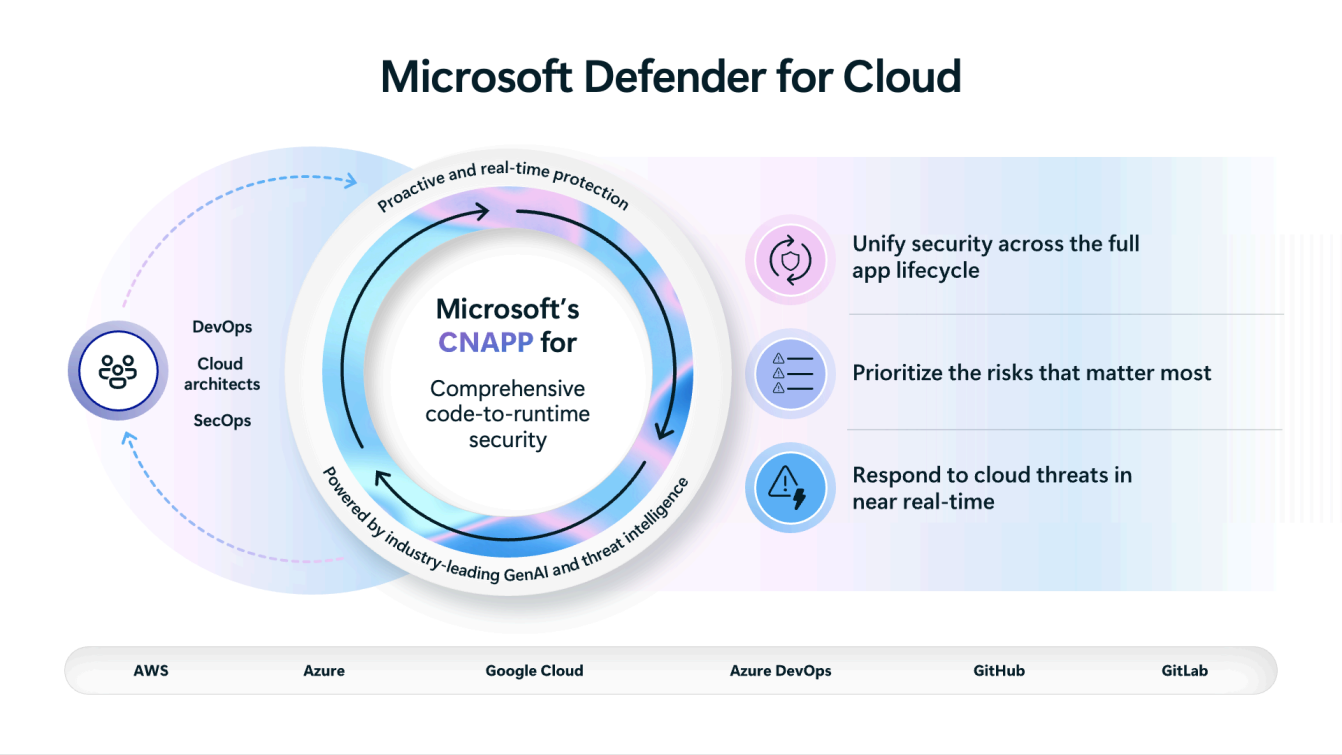
Microsoft Defender for Cloud is a Cloud Native Application Protection Platform (CNAPP), which is a unified solution that combines multiple cloud security tools to protect applications across their entire lifecycle. The solution provides a comprehensive view of your security posture across your cloud and on-premises resources. It also helps you secure multicloud and hybrid environments and integrates security into DevOps workflows. It has three core components:

- Cloud Security Posture Management (CSPM) checks and improves the security posture of cloud resources.
- Development Security Operations (DevSecOps) manages code-level security across multicloud and multi-pipeline environments.

- Cloud Workload Protection Platform (CWPP) defends workloads such as virtual machines (VMs), containers, storage, databases, and serverless functions from threats.

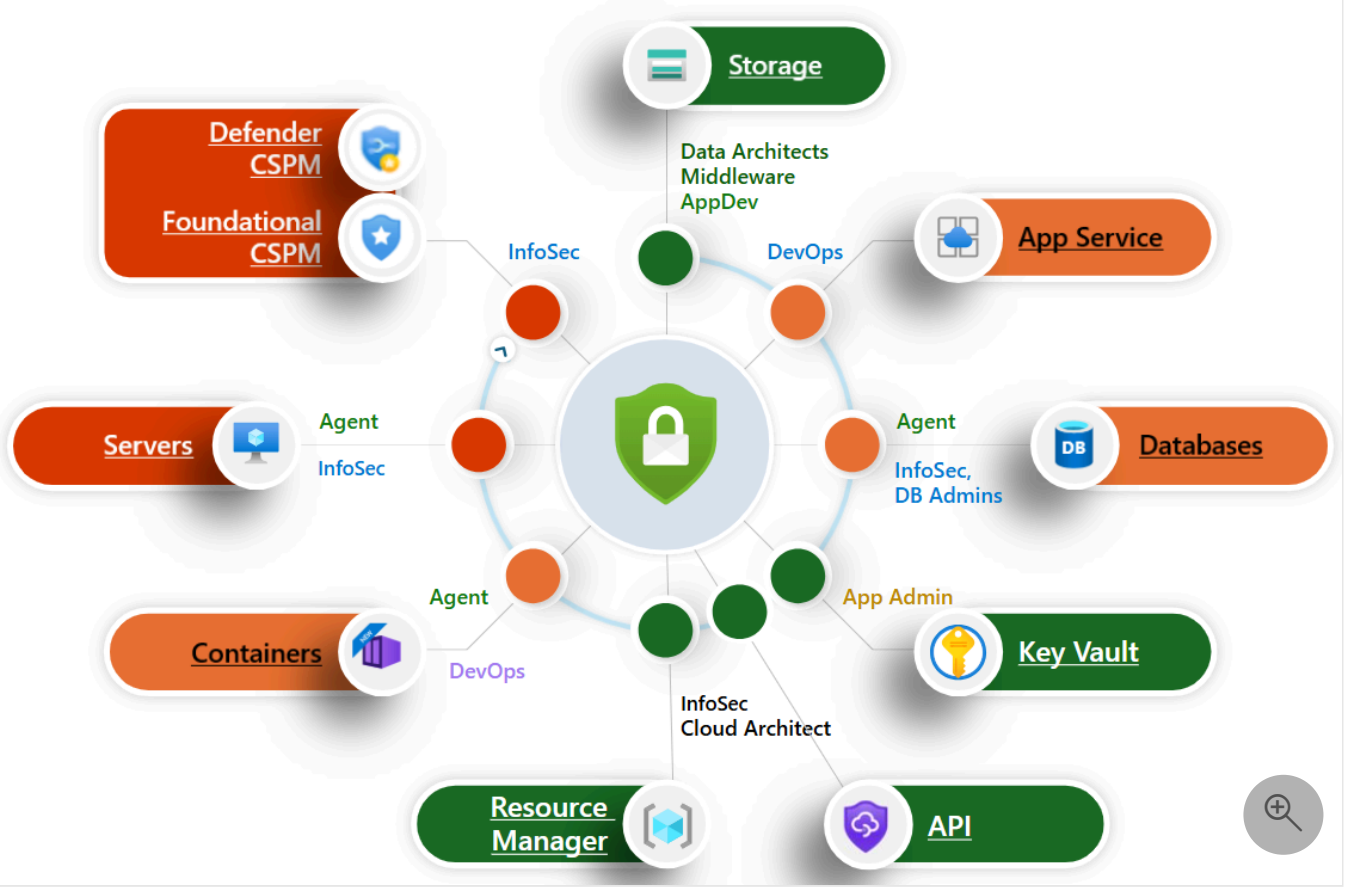
Defender for Cloud uses its broader Cloud Native Application Protection Platform (CNAPP) capabilities to unify protections into one experience. Defender for Cloud embeds security early in the development lifecycle. It helps DevOps teams find misconfigurations, apply policies, and fix risks early. Defender for Cloud integrates with the Defender XDR portal and the Microsoft Security ecosystem, offering unified posture management and SOC experiences.

In addition to its core CNAPP capabilities, Defender for Cloud delivers AI security and AI threat protection to safeguard generative AI workloads throughout their lifecycle. These features help you discover AI applications, identify vulnerabilities, reduce risks, and detect threats targeting your generative AI workloads.



> **Note**
>
> For pricing information, check out the Defender for Cloud pricing page. You can also estimate costs with the Defender for Cloud cost calculator.

Cloud Native Application Protection Platform (CNAPP)



After you enable the Defender for Cloud solution on your Azure subscription, the system collects security data from your multicloud and DevOps environments. Defender for Cloud uses the data to provide insights, recommendations, and actions that help you protect your cloud workloads and resources. You can increase your cloud workloads protection and coverage by enabling additional plans that are listed in the following section.

Defender for Cloud's available plans and their CNAPP benefits include:

Expand table

Defender for Cloud plan	CNAPP benefits	Relevant links
Defender CSPM / Foundational CSPM	Provides advanced security posture capabilities including agentless vulnerability scanning, data-aware security posture, the cloud security graph, and advanced threat hunting.	Check out the differences between the CSPM plans. Enable the Defender CSPM plan.
Defender for Servers	Provides threat detection and advanced defenses for Windows and Linux machines that run in Azure, AWS, GCP, and on-premises environments.	Plan your Defender for Servers deployment Check out the differences between the Defender for Servers plans

Defender for Cloud plan	CNAPP benefits	Relevant links
		Deploy Defender for Servers
Defender for Containers	Provides environment hardening, vulnerability assessment, run time protection of Kubernetes nodes and clusters.	Overview of Container security in Microsoft Defender for Containers Defender for Containers architecture Protect your Azure, IaaS, AWS, and GCP containers with Defender for Containers
Defender for Resource Manager	Detects unusual and potentially harmful activity by automatically monitoring the resource management operations.	Overview of Microsoft Defender for Resource Manager Protect your resources with Defender for Resource Manager
Defender for Storage	Protects against malware, storage specific threats, sensitive data leakage, and Shared Access Signature (SAS) token misuse.	Overview of Microsoft Defender for Storage Malware scanning Detect threats to sensitive data Deploy Microsoft Defender for Storage
Defender for App Service	Identifies attacks that target applications running over App Service.	Overview of Defender for App Service to protect your Azure App Service web apps and APIs Protect your applications with Defender for App Service
Defender for Databases	Protects your entire database estate with attack detection and threat response for the various database types in Azure.	Overview of Microsoft Defender for Azure SQL Protect your databases with Defender for Databases What is Microsoft Defender

Defender for Cloud plan	CNAPP benefits	Relevant links
		for open-source relational databases Overview of Microsoft Defender for Azure Cosmos DB
Defender for Key Vault	Detects unusual and potentially harmful attempts to access or exploit Key Vault accounts.	Overview of Microsoft Defender for Key Vault Protect your key vaults with Defender for Key Vault
Defender for APIs	Provides visibility into business critical APIs, improves API security posture, prioritization of vulnerability fixes, and quickly detect active real-time threats.	About Microsoft Defender for APIs Protect your APIs with Defender for APIs
AI Services	Identifies threats to generative AI applications in real time and helps respond to security issues.	AI threat protection Enable threat protection for AI services

You can also check out the E-book "From plan to deployment: Implementing a Cloud Native Application Protection Platform (CNAPP) strategy", to learn more about implementing CNAPP in Defender for Cloud.

Cloud security posture management (CSPM)

The security of your cloud and on-premises resources depends on proper configuration and deployment. Defenders for Cloud recommendations help you secure your environment.

Defender for Cloud includes free Foundational CSPM capabilities. Enable advanced CSPM capabilities by using the Defender CSPM plan.

> **Note**
>
> Defender CSPM provides broad recommendation coverage in Defender for Cloud. Some recommendations are plan-specific and require you to enable specific CWPP plans to access them.

Expand table

Capability	What problem does it solve?	Get started	Defender plan
Centralized policy management	Define the security conditions that you want to maintain across your environment. The policy translates to recommendations that identify resource configurations that violate your security policy. The Microsoft cloud security benchmark is a built-in standard that applies security principles with detailed technical implementation guidance for Azure and other cloud providers (such as Amazon Web Services (AWS) and Google Cloud Platform (GCP).	Customize a security policy	Foundational CSPM (Free)
Secure score	Summarize your security posture based on the security recommendations. As you remediate recommendations, your secure score improves.	Track your secure score	Foundational CSPM (Free)
Multicloud coverage	Connect to your multicloud environments by using agentless methods for CSPM insight and CWPP protection.	Connect your Amazon AWS and Google GCP cloud resources to Defender for Cloud	Foundational CSPM (Free)
Cloud Security Posture Management (CSPM)	Use the dashboard to see weaknesses in your security posture.	Enable CSPM tools	Foundational CSPM (Free)
Advanced Cloud Security Posture Management	Get advanced tools to identify weaknesses in your security posture, including: - Governance to drive actions to improve your security posture - Regulatory compliance to verify compliance with security standards - Cloud security explorer to build a comprehensive view of your environment	Enable CSPM tools	Defender CSPM
Data Security Posture Management	Data security posture management automatically discovers datastores containing sensitive data, and helps reduce risk of data breaches.	Enable data security posture management	Defender CSPM or Defender for Storage
Attack path analysis	Model traffic on your network to identify potential risks before you implement changes to your environment.	Build queries to analyze paths	Defender CSPM

Capability	What problem does it solve?	Get started	Defender plan
Cloud Security Explorer	A map of your cloud environment that lets you build queries to find security risks.	Build queries to find security risks	Defender CSPM
Security governance	Drive security improvements through your organization by assigning tasks to resource owners and tracking progress in aligning your security state with your security policy.	Define governance rules	Defender CSPM
AI SPM	Provides a comprehensive view of your organization's AI Bill of Materials (AI BOM) which assesses the security posture of the scanned AI workloads.	Discover generative AI workloads	Defender CSPM

Development security operations (DevSecOps)

Defender for Cloud adds security to the start of development. It helps you secure code pipelines and environments, and monitor your security posture from one place. Defender for Cloud enables security teams to manage DevOps security across multi-pipeline environments.

Applications require security awareness at the code, infrastructure, and runtime levels to ensure that deployed applications are hardened against attacks.

Expand table

Capability	What problem does it solve?	Get started	Defender plan
Code pipeline insights	Empowers security teams with the ability to protect applications and resources from code to cloud across multi-pipeline environments, including GitHub, Azure DevOps, and GitLab. DevOps security findings, such as Infrastructure as Code (IaC) misconfigurations and exposed secrets, can then be correlated with other contextual cloud security insights to prioritize remediation in code.	Connect Azure DevOps, GitHub, and GitLab repositories to Defender for Cloud	Foundational CSPM (Free) and Defender CSPM

Cloud workload protection platform (CWPP)

Proactive security principles require implementing security practices to protect your workloads from threats. Cloud workload protection platforms (CWPP) provide workload-specific recommendations to guide you to the right security controls to protect your workloads.

When your environment is threatened, security alerts immediately indicate the nature and severity of the threat so you can plan your response. After identifying a threat in your environment, respond quickly to limit the risk to your resources.

Expand table

Capability	What problem does it solve?	Get started	Defender plan
Protect cloud servers	Provide server protections through Microsoft Defender for Endpoint or extended protection with just-in-time network access, file integrity monitoring, vulnerability assessment, and more.	Secure your multicloud and on-premises servers	Defender for Servers
Identify threats to your storage resources	Detect unusual and potentially harmful attempts to access or exploit your storage accounts by using advanced threat detection capabilities and Microsoft Threat Intelligence data to provide contextual security alerts.	Protect your cloud storage resources	Defender for Storage
Protect cloud databases	Protect your entire database estate with attack detection and threat response for the most popular database types in Azure to protect the database engines and data types, according to their attack surface and security risks.	Deploy specialized protections for cloud and on-premises databases	- Defender for Azure SQL Databases - Defender for SQL servers on machines - Defender for Open-source relational databases - Defender for Azure Cosmos DB
Protect containers	Secure your containers so you can improve, monitor, and maintain the security of your clusters, containers, and their applications with environment hardening, vulnerability assessments, and run-time protection.	Find security risks in your containers	Defender for Containers
Infrastructure service insights	Diagnose weaknesses in your application infrastructure that can leave your environment susceptible to attack.	- Identify attacks targeting applications running over App Service - Detect attempts to exploit Key Vault accounts - Get alerted on	- Defender for App Service - Defender for Key Vault - Defender for Resource Manager - Defender for DNS

Capability	What problem does it solve?	Get started	Defender plan
		suspicious Resource Manager operations - Expose anomalous Domain Name System (DNS) activities	
Security alerts	Get informed of real-time events that threaten the security of your environment. Alerts are categorized and assigned severity levels to indicate proper responses.	Manage security alerts	Any workload protection Defender plan
Security incidents	Identify attack patterns by correlating alerts. Integrate with Security Information and Event Management (SIEM), Security Orchestration, Automation, and Response (SOAR), and traditional IT deployment solutions that respond to threats and reduce risk.	Export alerts to SIEM, SOAR, or ITSM systems	Any workload protection Defender plan

Important

- As of August 1, 2023, customers with an existing subscription to Defender for DNS can continue to use the service as a standalone plan.
- For new subscriptions, alerts about suspicious DNS activity are included as part of Defender for Servers Plan 2 (P2).
- There's no change to the protection scope: Defender for DNS continues to protect all Azure resources connected to Azure's default DNS resolvers. The change affects how DNS protection is billed and bundled, not what resources are covered.

AI security and threat protection

Microsoft Defender for Cloud provides AI security posture management and AI threat protection to help you secure your generative AI workloads across their entire lifecycle.

Expand table

Type of AI security	Description	Relevant links
AI security posture management (SPM)	Helps you discover generative AI applications, identify vulnerabilities, and reduce risks by using built-in recommendations and attack path analysis.	Learn more about AI security posture management

Type of AI security	Description	Relevant links
AI threat protection	Uses advanced threat detection techniques to identify and respond to threats targeting your generative AI workloads.	AI threat protection

Defender for Cloud also includes a Data and AI security dashboard. This dashboard gives you a central place to monitor and manage your data and AI resources, track risks, and check protection status.

Learn More

For more information about Defender for Cloud and how it works, see:

- A step-by-step walkthrough of Defender for Cloud
- An interview about Defender for Cloud with an expert in cybersecurity in Lessons Learned from the Field
- Microsoft Defender for Cloud - Use cases
- Microsoft Defender for Cloud PoC Series - Microsoft Defender for Containers
- Learn how Microsoft Defender for Cloud provides data security.

Next steps

Enable Microsoft Defender plans

Last updated on 04/19/2026

Azure security baseline for Virtual Machines - Linux Virtual Machines

> **Important**
>
> This security baseline is based on a previous version of Microsoft Cloud Security Benchmark (v1.0) and may contain outdated guidance. For the latest security guidance, see the Virtual Machines - Linux Virtual Machines documentation.

This security baseline applies guidance from the Microsoft cloud security benchmark version 1.0 to Virtual Machines - Linux Virtual Machines. The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the security controls defined by the Microsoft cloud security benchmark and the related guidance applicable to Virtual Machines - Linux Virtual Machines.

You can monitor this security baseline and its recommendations using Microsoft Defender for Cloud. Azure Policy definitions will be listed in the Regulatory Compliance section of the Microsoft Defender for Cloud portal page.

When a feature has relevant Azure Policy Definitions, they are listed in this baseline to help you measure compliance with the Microsoft cloud security benchmark controls and recommendations. Some recommendations may require a paid Microsoft Defender plan to enable certain security scenarios.

> **Note**
>
> **Features** not applicable to Virtual Machines - Linux Virtual Machines have been excluded. To see how Virtual Machines - Linux Virtual Machines completely maps to the Microsoft cloud security benchmark, see the full Virtual Machines - Linux Virtual Machines security baseline mapping file.

Security profile

The security profile summarizes high-impact behaviors of Virtual Machines - Linux Virtual Machines, which may result in increased security considerations.

Expand table

Service Behavior Attribute	Value
Product Category	Compute
Customer can access HOST / OS	Full Access
Service can be deployed into customer's virtual network	True
Stores customer content at rest	True

Network security

For more information, see the Microsoft cloud security benchmark: Network security.

NS-1: Establish network segmentation boundaries

Features

Virtual Network Integration

Description: Service supports deployment into customer's private Virtual Network (VNet). Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: Virtual networks and virtual machines in Azure

Network Security Group Support

Description: Service network traffic respects Network Security Groups rule assignment on its subnets. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use network security groups (NSG) to restrict or monitor traffic by port, protocol, source IP address, or destination IP address. Create NSG rules to restrict your service's open ports (such as preventing management ports from being accessed from untrusted networks). Be aware that by default, NSGs deny all inbound traffic but allow traffic from virtual network and Azure Load Balancers.

When you create an Azure virtual machine (VM), you must create a virtual network or use an existing virtual network and configure the VM with a subnet. Ensure that all deployed subnets have a Network Security Group applied with network access controls specific to your applications trusted ports and sources.

Reference: Network security groups

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
All network ports should be restricted on network security groups associated to your virtual machine	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive network hardening recommendations should be applied on internet facing virtual machines	Azure Security Center analyzes the traffic patterns of Internet facing virtual machines and provides Network Security Group rule recommendations that reduce the potential attack surface	AuditIfNotExists, Disabled	3.0.0

NS-2: Secure cloud services with network controls

Features

Disable Public Network Access

Description: Service supports disabling public network access either through using service-level IP ACL filtering rule (not NSG or Azure Firewall) or using a 'Disable Public Network Access' toggle switch. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Services such as iptables or firewalld may be installed in the Linux OS and provide network filtering to disable public access.

Identity management

For more information, see the Microsoft cloud security benchmark: Identity management.

IM-1: Use centralized identity and authentication system

Features

Azure AD Authentication Required for Data Plane Access

Description: Service supports using Azure AD authentication for data plane access. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Active Directory (Azure AD) as the default authentication method to control your data plane access.

Reference: Log in to a Linux virtual machine in Azure by using Azure AD and OpenSSH

Local Authentication Methods for Data Plane Access

Description: Local authentications methods supported for data plane access, such as a local username and password. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: A local administrator account is created by default during the initial deployment of the virtual machine. Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

IM-3: Manage application identities securely and automatically

Features

Managed Identities

Description: Data plane actions support authentication using managed identities. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Managed identity is traditionally leveraged by Linux VM to authenticate to other services. If the Linux VM supports Azure AD authentication then managed identity may be supported.

Configuration Guidance: Use Azure managed identities instead of service principals when possible, which can authenticate to Azure services and resources that support Azure Active Directory (Azure AD) authentication. Managed identity credentials are fully managed, rotated, and protected by the platform, avoiding hard-coded credentials in source code or configuration files.

Service Principals

Description: Data plane supports authentication using service principals. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Service principals may be used by applications running in the Linux VM.

Configuration Guidance: There is no current Microsoft guidance for this feature configuration. Please review and determine if your organization wants to configure this security feature.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	The Guest Configuration extension requires a system assigned managed identity. Azure virtual machines in the scope of this policy will be non-compliant when they have the Guest Configuration extension installed but do not have a system assigned managed identity. Learn more at https://aka.ms/gcpol	AuditIfNotExists, Disabled	1.0.1

IM-7: Restrict resource access based on conditions

Features

Conditional Access for Data Plane

Description: Data plane access can be controlled using Azure AD Conditional Access Policies. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Use Azure AD as a core authentication platform and a certificate authority to SSH into a Linux VM by using Azure AD and OpenSSH certificate-based authentication. This functionality allows organizations to manage access to VMs with Azure role-based access control (RBAC) and Conditional Access policies.

Configuration Guidance: Define the applicable conditions and criteria for Azure Active Directory (Azure AD) conditional access in the workload. Consider common use cases such as blocking or granting access from specific locations, blocking risky sign-in behavior, or requiring organization-managed devices for specific applications.

Reference: Log in to a Linux virtual machine in Azure by using Azure AD and OpenSSH

IM-8: Restrict the exposure of credential and secrets

Features

Service Credential and Secrets Support Integration and Storage in Azure Key Vault

Description: Data plane supports native use of Azure Key Vault for credential and secrets store. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Within the data plane or operating system, services may call Azure Key Vault for credentials or secrets.

Configuration Guidance: Ensure that secrets and credentials are stored in secure locations such as Azure Key Vault, instead of embedding them into code or configuration files.

Privileged access

For more information, see the Microsoft cloud security benchmark: Privileged access.

PA-1: Separate and limit highly privileged/administrative users

Features

Local Admin Accounts

Description: Service has the concept of a local administrative account. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: Quickstart: Create a Linux virtual machine in the Azure portal

PA-7: Follow just enough administration (least privilege) principle

Features

Azure RBAC for Data Plane

Description: Azure Role-Based Access Control (Azure RBAC) can be used to managed access to service's data plane actions. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Use Azure AD as a core authentication platform and a certificate authority to SSH into a Linux VM by using Azure AD and OpenSSH certificate-based authentication. This

functionality allows organizations to manage access to VMs with Azure role-based access control (RBAC) and Conditional Access policies.

Configuration Guidance: With RBAC, specify who can log in to a VM as a regular user or with administrator privileges. When users join your team, you can update the Azure RBAC policy for the VM to grant access as appropriate. When employees leave your organization and their user accounts are disabled or removed from Azure AD, they no longer have access to your resources.

Reference: Log in to a Linux virtual machine in Azure by using Azure AD and OpenSSH

PA-8: Determine access process for cloud provider support

Features

Customer Lockbox

Description: Customer Lockbox can be used for Microsoft support access. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: In support scenarios where Microsoft needs to access your data, use Customer Lockbox to review, then approve or reject each of Microsoft's data access requests.

Data protection

For more information, see the Microsoft cloud security benchmark: Data protection.

DP-1: Discover, classify, and label sensitive data

Features

Sensitive Data Discovery and Classification

Description: Tools (such as Azure Purview or Azure Information Protection) can be used for data discovery and classification in the service. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-2: Monitor anomalies and threats targeting sensitive data

Features

Data Leakage/Loss Prevention

Description: Service supports DLP solution to monitor sensitive data movement (in customer's content). Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-3: Encrypt sensitive data in transit

Features

Data in Transit Encryption

Description: Service supports data in-transit encryption for data plane. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Certain communication protocols such as SSH are encrypted by default. However, other services such as HTTP must be configured to use TLS for encryption.

Configuration Guidance: Enable secure transfer in services where there is a native data in transit encryption feature built in. Enforce HTTPS on any web applications and services and ensure TLS v1.2 or later is used. Legacy versions such as SSL 3.0, TLS v1.0 should be disabled. For remote management of Virtual Machines, use SSH (for Linux) or RDP/TLS (for Windows) instead of an unencrypted protocol.

Reference: In-transit encryption in VMs

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows machines should be configured to use secure communication protocols	To protect the privacy of information communicated over the Internet, your machines should use the latest version of the industry-standard cryptographic protocol, Transport Layer Security (TLS). TLS secures communications over a network by encrypting a connection between machines.	AuditIfNotExists, Disabled	4.1.1

DP-4: Enable data at rest encryption by default

Features

Data at Rest Encryption Using Platform Keys

Description: Data at-rest encryption using platform keys is supported, any customer content at rest is encrypted with these Microsoft managed keys. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: By default, managed disks use platform-managed encryption keys. All managed disks, snapshots, images, and data written to existing managed disks are automatically encrypted-at-rest with platform-managed keys.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: Server-side encryption of Azure Disk Storage - Platform-managed keys

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should encrypt temp disks, caches, and data flows between Compute and Storage resources	By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys. Temp disks, data caches and data flowing between compute and storage aren't encrypted. Disregard this recommendation if: 1. using encryption-at-host, or 2. server-side encryption on Managed Disks meets your security requirements. Learn more in: Server-side encryption of Azure Disk Storage: https://aka.ms/disksse, Different disk encryption offerings: https://aka.ms/diskencryptioncomparison	AuditIfNotExists, Disabled	2.0.3

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Linux virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys; temp disks and data caches aren't encrypted, and data isn't encrypted when flowing between compute and storage resources. Use Azure Disk Encryption or EncryptionAtHost to encrypt all this data.Visit https://aka.ms/diskencryptioncomparison to compare encryption offerings. This policy requires two prerequisites to be deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	1.2.0-preview

DP-5: Use customer-managed key option in data at rest encryption when required

Features

Data at Rest Encryption Using CMK

Description: Data at-rest encryption using customer-managed keys is supported for customer content stored by the service. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: You can choose to manage encryption at the level of each managed disk, with your own keys. When you specify a customer-managed key, that key is used to protect and control access to the key that encrypts your data. Customer-managed keys offer greater flexibility to manage access controls.

Configuration Guidance: If required for regulatory compliance, define the use case and service scope where encryption using customer-managed keys are needed. Enable and implement data at rest encryption using customer-managed key for those services.

Virtual disks on Virtual Machines (VM) are encrypted at rest using either Server-side encryption or Azure disk encryption (ADE). Azure Disk Encryption leverages the DM-Crypt feature of Linux to encrypt managed disks with customer-managed keys within the guest VM. Server-side encryption with customer-managed keys improves on ADE by enabling you to use any OS types and images for your VMs by encrypting data in the Storage service.

Reference: Server-side encryption of Azure Disk Storage - Customer-managed keys

DP-6: Use a secure key management process

Features

Key Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer keys, secrets, or certificates. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Key Vault to create and control the life cycle of your encryption keys, including key generation, distribution, and storage. Rotate and revoke your keys in Azure Key Vault and your service based on a defined schedule or when there is a key retirement or compromise. When there is a need to use customer-managed key (CMK) in the workload, service, or application level, ensure you follow the best practices for key management: Use a key hierarchy to generate a separate data encryption key (DEK) with your key encryption key (KEK) in your key vault. Ensure keys are registered with Azure Key Vault and referenced via key IDs from the service or application. If you need to bring your own key (BYOK) to the service (such as importing HSM-protected keys from your on-premises HSMs into Azure Key Vault), follow recommended guidelines to perform initial key generation and key transfer.

Reference: Creating and configuring a key vault for Azure Disk Encryption

DP-7: Use a secure certificate management process

Features

Certificate Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer certificates. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Asset management

For more information, see the Microsoft cloud security benchmark: Asset management.

AM-2: Use only approved services

Features

Azure Policy Support

Description: Service configurations can be monitored and enforced via Azure Policy. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Policy can be used to define the desired behavior for your organization's Windows VMs and Linux VMs. By using policies, an organization can enforce various conventions and rules throughout the enterprise and define and implement standard security configurations for Azure Virtual Machines. Enforcement of the desired behavior can help mitigate risk while contributing to the success of the organization.

Reference: Azure Policy built-in definitions for Azure Virtual Machines

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should be migrated to new Azure Resource Manager resources	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance, access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management	Audit, Deny, Disabled	1.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should be migrated to new Azure Resource Manager resources	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance, access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management	Audit, Deny, Disabled	1.0.0

AM-5: Use only approved applications in virtual machine

Features

Microsoft Defender for Cloud - Adaptive Application Controls

Description: Service can limit what customer applications run on the virtual machine using Adaptive Application Controls in Microsoft Defender for Cloud. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Microsoft Defender for Cloud adaptive application controls to discover applications running on virtual machines (VMs) and generate an application allow list to mandate which approved applications can run in the VM environment.

Reference: Use adaptive application controls to reduce your machines' attack surfaces

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive application controls for defining safe applications	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This	AuditIfNotExists, Disabled	3.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
should be enabled on your machines	helps harden your machines against malware. To simplify the process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.		

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive application controls for defining safe applications should be enabled on your machines	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This helps harden your machines against malware. To simplify the process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.	AuditIfNotExists, Disabled	3.0.0

Logging and threat detection

For more information, see the Microsoft cloud security benchmark: Logging and threat detection.

LT-1: Enable threat detection capabilities

Features

Microsoft Defender for Service / Product Offering

Description: Service has an offering-specific Microsoft Defender solution to monitor and alert on security issues. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Defender for Servers extends protection to your Windows and Linux machines running in Azure. Defender for Servers integrates with Microsoft Defender for Endpoint to provide endpoint detection and response (EDR), and also provides a host of additional threat protection features, such as security baselines and OS level assessments, vulnerability assessment scanning, adaptive application controls (AAC), file integrity monitoring (FIM), and more.

Reference: Plan your Defender for Servers deployment

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows Defender Exploit Guard should be enabled on your machines	Windows Defender Exploit Guard uses the Azure Policy Guest Configuration agent. Exploit Guard has four components that are designed to lock down devices against a wide variety of attack vectors and block behaviors commonly used in malware attacks while enabling enterprises to balance their security risk and productivity requirements (Windows only).	AuditIfNotExists, Disabled	2.0.0

LT-4: Enable logging for security investigation

Features

Azure Resource Logs

Description: Service produces resource logs that can provide enhanced service-specific metrics and logging. The customer can configure these resource logs and send them to their own data sink like a storage account or log analytics workspace. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Monitor starts automatically collecting metric data for your virtual machine host when you create the VM. To collect logs and performance data from the guest operating system of the virtual machine, though, you must install the Azure Monitor agent. You can install the agent and configure collection using either VM insights or by creating a data collection rule.

Reference: Log Analytics agent overview

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	Security Center uses the Microsoft Dependency agent to collect network traffic data from your Azure virtual machines to enable advanced network protection features such as traffic visualization on the network map, network hardening recommendations and specific network threats.	AuditIfNotExists, Disabled	1.0.2-preview

Posture and vulnerability management

For more information, see the Microsoft cloud security benchmark: Posture and vulnerability management.

PV-3: Define and establish secure configurations for compute resources

Features

Azure Automation State Configuration

Description: Azure Automation State Configuration can be used to maintain the security configuration of the operating system. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Automation State Configuration to maintain the security configuration of the operating system.

Reference: Configure a VM with Desired State Configuration

Azure Policy Guest Configuration Agent

Description: Azure Policy guest configuration agent can be installed or deployed as an extension to compute resources. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Azure Policy Guest Configuration is now called Azure Automanage Machine Configuration.

Configuration Guidance: Use Microsoft Defender for Cloud and Azure Policy guest configuration agent to regularly assess and remediate configuration deviations on your Azure compute resources, including VMs, containers, and others.

Reference: Understand the machine configuration feature of Azure Automanage

Custom VM Images

Description: Service supports using user-supplied VM images or pre-built images from the marketplace with certain baseline configurations pre-applied. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use a pre-configured hardened image from a trusted supplier such as Microsoft or build a desired secure configuration baseline into the VM image template.

Reference: Tutorial: Create a custom image of an Azure VM with the Azure CLI

PV-4: Audit and enforce secure configurations for compute resources

Features

Trusted Launch Virtual Machine

Description: Trusted Launch protects against advanced and persistent attack techniques by combining infrastructure technologies like secure boot, vTPM, and integrity monitoring. Each technology provides another layer of defense against sophisticated threats. Trusted launch allows the secure deployment of virtual machines with verified boot loaders, OS kernels, and drivers, and securely protects keys, certificates, and secrets in the virtual machines. Trusted launch also provides insights and confidents of the entire boot chain's integrity and ensures workloads are trusted and verifiable. Trusted launch is integrated with Microsoft Defender for Cloud to ensure VMs are properly configured, by remotely attesting VM is booted in a healthy way. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature note: Trusted launch is available for generation 2 VMs. Trusted launch requires the creation of new virtual machines. You can't enable trusted launch on existing virtual machines that were initially created without it.

Configuration Guidance: Trusted launch may be enabled during the deployment of the VM. Enable all three - Secure Boot, vTPM, and integrity boot monitoring to ensure the best security posture for the virtual machine. Please note that there are a few prerequisites including onboarding your subscription to Microsoft Defender for Cloud, assigning certain Azure Policy initiatives, and configuring firewall policies.

Reference: Deploy a VM with trusted launch enabled

PV-5: Perform vulnerability assessments

Features

Vulnerability Assessment using Microsoft Defender

Description: Service can be scanned for vulnerability scan using Microsoft Defender for Cloud or other Microsoft Defender services embedded vulnerability assessment capability (including Microsoft Defender for server, container registry, App Service, SQL, and DNS). Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Follow recommendations from Microsoft Defender for Cloud for performing vulnerability assessments on your Azure virtual machines.

Reference: Plan your Defender for Servers deployment

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
A vulnerability assessment solution should be enabled on your virtual machines	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.	AuditIfNotExists, Disabled	3.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
A vulnerability assessment solution should be enabled on your virtual machines	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard	AuditIfNotExists, Disabled	3.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
	pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.		

PV-6: Rapidly and automatically remediate vulnerabilities

Features

Azure Update Manager

Description: Service can use Azure Update Manager to deploy patches and updates automatically. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Customer

Configuration Guidance: Use Azure Update Manager to ensure that the most recent security updates are installed on your Linux VMs.

Reference: Manage updates and patches for your VMs

Azure Guest Patching Service

Description: Service can use Azure Guest Patching to deploy patches and updates automatically. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Services can leverage the different update mechanisms such as Auto OS Image Upgrades and Auto Guest Patching. The capabilities are recommended to apply the latest security and critical updates to your Virtual Machine's Guest OS by following the Safe Deployment Principles.

Auto Guest Patching allows you to automatically assess and update your Azure virtual machines to maintain security compliance with Critical and Security updates released each month. Updates are applied during off-peak hours, including VMs within an availability set. This capability is available for VMSS Flexible Orchestration, with future support on the roadmap for Uniform Orchestration.

If you run a stateless workload, Auto OS Image Upgrades are ideal to apply the latest update for your VMSS Uniform. With rollback capability, these updates are compatible with Marketplace or Custom images. Future rolling upgrade support on the roadmap for Flexible Orchestration.

Reference: Automatic VM Guest Patching for Azure VMs

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
System updates should be installed on your machines	Missing security system updates on your servers will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	4.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: System updates should be installed on your machines (powered by Update Center)	Your machines are missing system, security, and critical updates. Software updates often include critical patches to security holes. Such holes are frequently exploited in malware attacks so it's vital to keep your software updated. To install all outstanding patches and secure your machines, follow the remediation steps.	AuditIfNotExists, Disabled	1.0.0-preview

Endpoint security

For more information, see the Microsoft cloud security benchmark: Endpoint security.

ES-1: Use Endpoint Detection and Response (EDR)

Features

EDR Solution

Description: Endpoint Detection and Response (EDR) feature such as Azure Defender for servers can be deployed into the endpoint. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Defender for servers (with Microsoft Defender for Endpoint integrated) provides EDR capability to prevent, detect, investigate, and respond to advanced threats. Use Microsoft Defender for Cloud to deploy Azure Defender for servers for your endpoint and integrate the alerts to your SIEM solution such as Azure Sentinel.

Reference: Plan your Defender for Servers deployment

ES-2: Use modern anti-malware software

Features

Anti-Malware Solution

Description: Anti-malware feature such as Microsoft Defender Antivirus, Microsoft Defender for Endpoint can be deployed on the endpoint. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: For Linux, customers can have the choice of installing Microsoft Defender for Endpoint for Linux. Alternatively, customers also have the choice of installing third-party anti-malware products.

Reference: Microsoft Defender for Endpoint on Linux

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0

ES-3: Ensure anti-malware software and signatures are updated

Features

Anti-Malware Solution Health Monitoring

Description: Anti-malware solution provides health status monitoring for platform, engine, and automatic signature updates. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Security intelligence and product updates apply to Defender for Endpoint which can be installed on the Linux VMs.

Configuration Guidance: Configure your anti-malware solution to ensure the platform, engine and signatures are updated rapidly and consistently and their status can be monitored.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here -	AuditIfNotExists, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
resolved on your machines	https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.		

Backup and recovery

For more information, see the Microsoft cloud security benchmark: Backup and recovery.

BR-1: Ensure regular automated backups

Features

Azure Backup

Description: The service can be backed up by the Azure Backup service. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Enable Azure Backup and target Azure Virtual Machines (VM), as well as the desired frequency and retention periods. This includes complete system state backup. If you are using Azure disk encryption, Azure VM backup automatically handles the backup of customer-managed keys. For Azure Virtual Machines, you can use Azure Policy to enable automatic backups.

Reference: Backup and restore options for virtual machines in Azure

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Azure Backup should be enabled for Virtual Machines	Ensure protection of your Azure Virtual Machines by enabling Azure Backup. Azure Backup is a secure and cost effective data protection solution for Azure.	AuditIfNotExists, Disabled	3.0.0

Next steps

- See the Microsoft cloud security benchmark overview
- Learn more about Azure security baselines

Last updated on 02/25/2025

Azure security baseline for Virtual Machines - Windows Virtual Machines

> **Important**
>
> This security baseline is based on a previous version of Microsoft Cloud Security Benchmark (v1.0) and may contain outdated guidance. For the latest security guidance, see the Virtual Machines - Windows Virtual Machines documentation.

This security baseline applies guidance from the Microsoft cloud security benchmark version 1.0 to Virtual Machines - Windows Virtual Machines. The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the security controls defined by the Microsoft cloud security benchmark and the related guidance applicable to Virtual Machines - Windows Virtual Machines.

You can monitor this security baseline and its recommendations using Microsoft Defender for Cloud. Azure Policy definitions will be listed in the Regulatory Compliance section of the Microsoft Defender for Cloud portal page.

When a feature has relevant Azure Policy Definitions, they are listed in this baseline to help you measure compliance with the Microsoft cloud security benchmark controls and recommendations. Some recommendations may require a paid Microsoft Defender plan to enable certain security scenarios.

> **Note**
>
> **Features** not applicable to Virtual Machines - Windows Virtual Machines have been excluded. To see how Virtual Machines - Windows Virtual Machines completely maps to the Microsoft cloud security benchmark, see the full Virtual Machines - Windows Virtual Machines security baseline mapping file.

Security profile

The security profile summarizes high-impact behaviors of Virtual Machines - Windows Virtual Machines, which may result in increased security considerations.

Expand table

Service Behavior Attribute	Value
Product Category	Compute
Customer can access HOST / OS	Full Access
Service can be deployed into customer's virtual network	True
Stores customer content at rest	True

Network security

For more information, see the Microsoft cloud security benchmark: Network security.

NS-1: Establish network segmentation boundaries

Features

Virtual Network Integration

Description: Service supports deployment into customer's private Virtual Network (VNet). Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: Virtual networks and virtual machines in Azure

Network Security Group Support

Description: Service network traffic respects Network Security Groups rule assignment on its subnets. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use network security groups (NSG) to restrict or monitor traffic by port, protocol, source IP address, or destination IP address. Create NSG rules to restrict your service's open ports (such as preventing management ports from being accessed from untrusted networks). Be aware that by default, NSGs deny all inbound traffic but allow traffic from virtual network and Azure Load Balancers.

When you create an Azure virtual machine (VM), you must create a virtual network or use an existing virtual network and configure the VM with a subnet. Ensure that all deployed subnets have a Network Security Group applied with network access controls specific to your applications trusted ports and sources.

Reference: Network security groups

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
All network ports should be restricted on network security groups associated to your virtual machine	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive network hardening recommendations should be applied on internet facing virtual machines	Azure Security Center analyzes the traffic patterns of Internet facing virtual machines and provides Network Security Group rule recommendations that reduce the potential attack surface	AuditIfNotExists, Disabled	3.0.0

NS-2: Secure cloud services with network controls

Features

Disable Public Network Access

Description: Service supports disabling public network access either through using service-level IP ACL filtering rule (not NSG or Azure Firewall) or using a 'Disable Public Network Access' toggle switch. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use services at the OS-level such as Windows Defender Firewall to provide network filtering to disable public access.

Identity management

For more information, see the Microsoft cloud security benchmark: Identity management.

IM-1: Use centralized identity and authentication system

Features

Azure AD Authentication Required for Data Plane Access

Description: Service supports using Azure AD authentication for data plane access. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Active Directory (Azure AD) as the default authentication method to control your data plane access.

Reference: Log in to a Windows virtual machine in Azure by using Azure AD including passwordless

Local Authentication Methods for Data Plane Access

Description: Local authentications methods supported for data plane access, such as a local username and password. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: A local administrator account is created by default during the initial deployment of the virtual machine. Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

IM-3: Manage application identities securely and automatically

Features

Managed Identities

Description: Data plane actions support authentication using managed identities. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Managed identity is typically leveraged by Windows VM to authenticate to other services. If the Windows VM supports Azure AD authentication then managed identity may be supported.

Configuration Guidance: Use Azure managed identities instead of service principals when possible, which can authenticate to Azure services and resources that support Azure Active

Directory (Azure AD) authentication. Managed identity credentials are fully managed, rotated, and protected by the platform, avoiding hard-coded credentials in source code or configuration files.

Service Principals

Description: Data plane supports authentication using service principals. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Service principals may be used by applications running in the Windows VM.

Configuration Guidance: There is no current Microsoft guidance for this feature configuration. Please review and determine if your organization wants to configure this security feature.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	The Guest Configuration extension requires a system assigned managed identity. Azure virtual machines in the scope of this policy will be non-compliant when they have the Guest Configuration extension installed but do not have a system assigned managed identity. Learn more at https://aka.ms/gcpol	AuditIfNotExists, Disabled	1.0.1

IM-7: Restrict resource access based on conditions

Features

Conditional Access for Data Plane

Description: Data plane access can be controlled using Azure AD Conditional Access Policies. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Use Azure AD as a core authentication platform to RDP into Windows Server 2019 Datacenter edition and later, or Windows 10 1809 and later. You can then centrally control and enforce Azure role-based access control (RBAC) and Conditional Access policies that allow or deny access to the VMs.

Configuration Guidance: Define the applicable conditions and criteria for Azure Active Directory (Azure AD) conditional access in the workload. Consider common use cases such as blocking or granting access from specific locations, blocking risky sign-in behavior, or requiring organization-managed devices for specific applications.

Reference: Log in to a Windows virtual machine in Azure by using Azure AD including passwordless

IM-8: Restrict the exposure of credential and secrets

Features

Service Credential and Secrets Support Integration and Storage in Azure Key Vault

Description: Data plane supports native use of Azure Key Vault for credential and secrets store. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Within the data plane or operating system, services may call Azure Key Vault for credentials or secrets.

Configuration Guidance: Ensure that secrets and credentials are stored in secure locations such as Azure Key Vault, instead of embedding them into code or configuration files.

Privileged access

For more information, see the Microsoft cloud security benchmark: Privileged access.

PA-1: Separate and limit highly privileged/administrative users

Features

Local Admin Accounts

Description: Service has the concept of a local administrative account. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: Quickstart: Create a Windows virtual machine in the Azure portal

PA-7: Follow just enough administration (least privilege) principle

Features

Azure RBAC for Data Plane

Description: Azure Role-Based Access Control (Azure RBAC) can be used to managed access to service's data plane actions. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Use Azure AD as a core authentication platform to RDP into Windows Server 2019 Datacenter edition and later, or Windows 10 1809 and later. You can then centrally control and enforce Azure role-based access control (RBAC) and Conditional Access policies that allow or deny access to the VMs.

Configuration Guidance: With RBAC, specify who can log in to a VM as a regular user or with administrator privileges. When users join your team, you can update the Azure RBAC policy for the VM to grant access as appropriate. When employees leave your organization and their user accounts are disabled or removed from Azure AD, they no longer have access to your resources.

Reference: Log in to a Windows virtual machine in Azure by using Azure AD including passwordless

PA-8: Determine access process for cloud provider support

Features

Customer Lockbox

Description: Customer Lockbox can be used for Microsoft support access. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: In support scenarios where Microsoft needs to access your data, use Customer Lockbox to review, then approve or reject each of Microsoft's data access requests.

Data protection

For more information, see the Microsoft cloud security benchmark: Data protection.

DP-1: Discover, classify, and label sensitive data

Features

Sensitive Data Discovery and Classification

Description: Tools (such as Azure Purview or Azure Information Protection) can be used for data discovery and classification in the service. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-2: Monitor anomalies and threats targeting sensitive data

Features

Data Leakage/Loss Prevention

Description: Service supports DLP solution to monitor sensitive data movement (in customer's content). Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-3: Encrypt sensitive data in transit

Features

Data in Transit Encryption

Description: Service supports data in-transit encryption for data plane. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Certain communication protocols such as SSH are encrypted by default. However, other services such as HTTP must be configured to use TLS for encryption.

Configuration Guidance: Enable secure transfer in services where there is a native data in transit encryption feature built in. Enforce HTTPS on any web applications and services and ensure TLS v1.2 or later is used. Legacy versions such as SSL 3.0, TLS v1.0 should be disabled. For remote management of Virtual Machines, use SSH (for Linux) or RDP/TLS (for Windows) instead of an unencrypted protocol.

Reference: In-transit encryption in VMs

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows machines should be configured to use secure communication protocols	To protect the privacy of information communicated over the Internet, your machines should use the latest version of the industry-standard cryptographic protocol, Transport Layer Security (TLS). TLS secures communications over a network by encrypting a connection between machines.	AuditIfNotExists, Disabled	4.1.1

DP-4: Enable data at rest encryption by default

Features

Data at Rest Encryption Using Platform Keys

Description: Data at-rest encryption using platform keys is supported, any customer content at rest is encrypted with these Microsoft managed keys. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: By default, managed disks use platform-managed encryption keys. All managed disks, snapshots, images, and data written to existing managed disks are automatically encrypted-at-rest with platform-managed keys.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: Server-side encryption of Azure Disk Storage - Platform-managed Keys

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should encrypt temp disks, caches, and data flows between Compute and Storage resources	By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys. Temp disks, data caches and data flowing between compute and storage aren't encrypted. Disregard this recommendation if: 1. using encryption-at-host, or 2. server-side encryption on Managed Disks meets your security requirements. Learn more in: Server-side encryption of Azure Disk Storage: https://aka.ms/disksse, Different disk encryption offerings: https://aka.ms/diskencryptioncomparison	AuditIfNotExists, Disabled	2.0.3

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Linux virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys; temp disks and data caches aren't encrypted, and data isn't encrypted when flowing between compute and storage resources. Use Azure Disk Encryption or EncryptionAtHost to encrypt all this data.Visit https://aka.ms/diskencryptioncomparison to compare encryption offerings. This policy requires two prerequisites to be deployed to the policy	AuditIfNotExists, Disabled	1.2.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
	assignment scope. For details, visit https://aka.ms/gcpol .		

DP-5: Use customer-managed key option in data at rest encryption when required

Features

Data at Rest Encryption Using CMK

Description: Data at-rest encryption using customer-managed keys is supported for customer content stored by the service. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: You can choose to manage encryption at the level of each managed disk, with your own keys. When you specify a customer-managed key, that key is used to protect and control access to the key that encrypts your data. Customer-managed keys offer greater flexibility to manage access controls.

Configuration Guidance: If required for regulatory compliance, define the use case and service scope where encryption using customer-managed keys are needed. Enable and implement data at rest encryption using customer-managed key for those services.

Virtual disks on Virtual Machines (VM) are encrypted at rest using either Server-side encryption or Azure disk encryption (ADE). Azure Disk Encryption leverages the BitLocker feature of Windows to encrypt managed disks with customer-managed keys within the guest VM. Server-side encryption with customer-managed keys improves on ADE by enabling you to use any OS types and images for your VMs by encrypting data in the Storage service.

Reference: Server-side encryption of Azure Disk Storage

DP-6: Use a secure key management process

Features

Key Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer keys, secrets, or certificates. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Key Vault to create and control the life cycle of your encryption keys, including key generation, distribution, and storage. Rotate and revoke your keys in Azure Key Vault and your service based on a defined schedule or when there is a key retirement or compromise. When there is a need to use customer-managed key (CMK) in the workload, service, or application level, ensure you follow the best practices for key management: Use a key hierarchy to generate a separate data encryption key (DEK) with your key encryption key (KEK) in your key vault. Ensure keys are registered with Azure Key Vault and referenced via key IDs from the service or application. If you need to bring your own key (BYOK) to the service (such as importing HSM-protected keys from your on-premises HSMs into Azure Key Vault), follow recommended guidelines to perform initial key generation and key transfer.

Reference: Create and configure a key vault for Azure Disk Encryption on a Windows VM

DP-7: Use a secure certificate management process

Features

Certificate Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer certificates. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Asset management

For more information, see the Microsoft cloud security benchmark: Asset management.

AM-2: Use only approved services

Features

Azure Policy Support

Description: Service configurations can be monitored and enforced via Azure Policy. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Policy can be used to define the desired behavior for your organization's Windows VMs and Linux VMs. By using policies, an organization can enforce various conventions and rules throughout the enterprise and define and implement standard security configurations for Azure Virtual Machines. Enforcement of the desired behavior can help mitigate risk while contributing to the success of the organization.

Reference: Azure Policy built-in definitions for Azure Virtual Machines

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should be migrated to new Azure	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance,	Audit, Deny, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Resource Manager resources	access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management		

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should be migrated to new Azure Resource Manager resources	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance, access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management	Audit, Deny, Disabled	1.0.0

AM-5: Use only approved applications in virtual machine

Features

Microsoft Defender for Cloud - Adaptive Application Controls

Description: Service can limit what customer applications run on the virtual machine using Adaptive Application Controls in Microsoft Defender for Cloud. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Microsoft Defender for Cloud adaptive application controls to discover applications running on virtual machines (VMs) and generate an application allow list to mandate which approved applications can run in the VM environment.

Reference: Use adaptive application controls to reduce your machines' attack surfaces

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive application controls for defining safe applications should be enabled on your machines	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This helps harden your machines against malware. To simplify the process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.	AuditIfNotExists, Disabled	3.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive application controls for defining safe applications should be enabled on your machines	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This helps harden your machines against malware. To simplify the process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.	AuditIfNotExists, Disabled	3.0.0

Logging and threat detection

For more information, see the Microsoft cloud security benchmark: Logging and threat detection.

LT-1: Enable threat detection capabilities

Features

Microsoft Defender for Service / Product Offering

Description: Service has an offering-specific Microsoft Defender solution to monitor and alert on security issues. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Defender for Servers extends protection to your Windows and Linux machines running in Azure. Defender for Servers integrates with Microsoft Defender for Endpoint to provide endpoint detection and response (EDR), and also provides a host of additional threat protection features, such as security baselines and OS level assessments, vulnerability assessment scanning, adaptive application controls (AAC), file integrity monitoring (FIM), and more.

Reference: Plan your Defender for Servers deployment

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows Defender Exploit Guard should be enabled on your machines	Windows Defender Exploit Guard uses the Azure Policy Guest Configuration agent. Exploit Guard has four components that are designed to lock down devices against a wide variety of attack vectors and block behaviors commonly used in malware attacks while enabling enterprises to balance their security risk and productivity requirements (Windows only).	AuditIfNotExists, Disabled	2.0.0

LT-4: Enable logging for security investigation

Features

Azure Resource Logs

Description: Service produces resource logs that can provide enhanced service-specific metrics and logging. The customer can configure these resource logs and send them to their own data sink like a storage account or log analytics workspace. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Monitor starts automatically collecting metric data for your virtual machine host when you create the VM. To collect logs and performance data from the guest operating system of the virtual machine, though, you must install the Azure Monitor agent. You can install the agent and configure collection using either VM insights or by creating a data collection rule.

Reference: Log Analytics agent overview

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	Security Center uses the Microsoft Dependency agent to collect network traffic data from your Azure virtual machines to enable advanced network protection features such as traffic visualization on the network map, network hardening recommendations and specific network threats.	AuditIfNotExists, Disabled	1.0.2-preview

Posture and vulnerability management

For more information, see the Microsoft cloud security benchmark: Posture and vulnerability management.

PV-3: Define and establish secure configurations for compute resources

Features

Azure Automation State Configuration

Description: Azure Automation State Configuration can be used to maintain the security configuration of the operating system. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Automation State Configuration to maintain the security configuration of the operating system.

Reference: Configure a VM with Desired State Configuration

Azure Policy Guest Configuration Agent

Description: Azure Policy guest configuration agent can be installed or deployed as an extension to compute resources. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Azure Policy Guest Configuration is now called Azure Automanage Machine Configuration.

Configuration Guidance: Use Microsoft Defender for Cloud and Azure Policy guest configuration agent to regularly assess and remediate configuration deviations on your Azure compute resources, including VMs, containers, and others.

Reference: Understand the machine configuration feature of Azure Automanage

Custom VM Images

Description: Service supports using user-supplied VM images or pre-built images from the marketplace with certain baseline configurations pre-applied. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use a pre-configured hardened image from a trusted supplier such as Microsoft or build a desired secure configuration baseline into the VM image template

Reference: Tutorial: Create Windows VM images with Azure PowerShell

PV-4: Audit and enforce secure configurations for compute resources

Features

Trusted Launch Virtual Machine

Description: Trusted Launch protects against advanced and persistent attack techniques by combining infrastructure technologies like secure boot, vTPM, and integrity monitoring. Each technology provides another layer of defense against sophisticated threats. Trusted launch allows the secure deployment of virtual machines with verified boot loaders, OS kernels, and drivers, and securely protects keys, certificates, and secrets in the virtual machines. Trusted launch also provides insights and confidents of the entire boot chain's integrity and ensures workloads are trusted and verifiable. Trusted launch is integrated with Microsoft Defender for Cloud to ensure VMs are properly configured, by remotely attesting VM is booted in a healthy way. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature note: Trusted launch is available for generation 2 VMs. Trusted launch requires the creation of new virtual machines. You can't enable trusted launch on existing virtual machines that were initially created without it.

Configuration Guidance: Trusted launch may be enabled during the deployment of the VM. Enable all three - Secure Boot, vTPM, and integrity boot monitoring to ensure the best security posture for the virtual machine. Please note that there are a few prerequisites including onboarding your subscription to Microsoft Defender for Cloud, assigning certain Azure Policy initiatives, and configuring firewall policies.

Reference: Deploy a VM with trusted launch enabled

PV-5: Perform vulnerability assessments

Features

Vulnerability Assessment using Microsoft Defender

Description: Service can be scanned for vulnerability scan using Microsoft Defender for Cloud or other Microsoft Defender services embedded vulnerability assessment capability (including Microsoft Defender for server, container registry, App Service, SQL, and DNS). Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Follow recommendations from Microsoft Defender for Cloud for performing vulnerability assessments on your Azure virtual machines.

Reference: Plan your Defender for Servers deployment

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
A vulnerability assessment solution should be enabled on your virtual machines	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.	AuditIfNotExists, Disabled	3.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
A vulnerability assessment solution should be enabled on your virtual machines	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.	AuditIfNotExists, Disabled	3.0.0

PV-6: Rapidly and automatically remediate vulnerabilities

Features

Azure Update Manager

Description: Service can use Azure Update Manager to deploy patches and updates automatically. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	True	Customer

Configuration Guidance: Use Azure Update Manager to ensure that the most recent security updates are installed on your Windows VMs. For Windows VMs, ensure Windows Update has been enabled and set to update automatically.

Reference: Manage updates and patches for your VMs

Azure Guest Patching Service

Description: Service can use Azure Guest Patching to deploy patches and updates automatically. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Services can leverage the different update mechanisms such as Auto OS Image Upgrades and Auto Guest Patching. The capabilities are recommended to apply the latest security and critical updates to your Virtual Machine's Guest OS by following the Safe Deployment Principles.

Auto Guest Patching allows you to automatically assess and update your Azure virtual machines to maintain security compliance with Critical and Security updates released each month. Updates are applied during off-peak hours, including VMs within an availability set. This capability is available for VMSS Flexible Orchestration, with future support on the roadmap for Uniform Orchestration.

If you run a stateless workload, Auto OS Image Upgrades are ideal to apply the latest update for your VMSS Uniform. With rollback capability, these updates are compatible with Marketplace or Custom images. Future rolling upgrade support on the roadmap for Flexible Orchestration.

Reference: Automatic VM Guest Patching for Azure VMs

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
System updates should be installed on your machines	Missing security system updates on your servers will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	4.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: System updates should be installed on your	Your machines are missing system, security, and critical updates. Software updates often include critical patches to security holes. Such holes are	AuditIfNotExists, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
machines (powered by Update Center)	frequently exploited in malware attacks so it's vital to keep your software updated. To install all outstanding patches and secure your machines, follow the remediation steps.		

Endpoint security

For more information, see the Microsoft cloud security benchmark: Endpoint security.

ES-1: Use Endpoint Detection and Response (EDR)

Features

EDR Solution

Description: Endpoint Detection and Response (EDR) feature such as Azure Defender for servers can be deployed into the endpoint. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Defender for servers (with Microsoft Defender for Endpoint integrated) provides EDR capability to prevent, detect, investigate, and respond to advanced threats. Use Microsoft Defender for Cloud to deploy Azure Defender for servers for your endpoint and integrate the alerts to your SIEM solution such as Azure Sentinel.

Reference: Plan your Defender for Servers deployment

ES-2: Use modern anti-malware software

Features

Anti-Malware Solution

Description: Anti-malware feature such as Microsoft Defender Antivirus, Microsoft Defender for Endpoint can be deployed on the endpoint. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: For Windows Server 2016 and above, Microsoft Defender for Antivirus is installed by default. For Windows Server 2012 R2 and above, customers can install SCEP (System Center Endpoint Protection). Alternatively, customers also have the choice of installing third-party anti-malware products.

Reference: Defender for Endpoint onboarding Windows Server

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint	AuditIfNotExists, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
should be resolved on your machines	protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.		

ES-3: Ensure anti-malware software and signatures are updated

Features

Anti-Malware Solution Health Monitoring

Description: Anti-malware solution provides health status monitoring for platform, engine, and automatic signature updates. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Security intelligence and product updates apply to Defender for Endpoint which can be installed on the Windows VMs.

Configuration Guidance: Configure your anti-malware solution to ensure the platform, engine and signatures are updated rapidly and consistently and their status can be monitored.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0

Backup and recovery

For more information, see the Microsoft cloud security benchmark: Backup and recovery.

BR-1: Ensure regular automated backups

Features

Azure Backup

Description: The service can be backed up by the Azure Backup service. Learn more.

Expand table

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Enable Azure Backup and configure the backup source (such as Azure Virtual Machines, SQL Server, HANA databases, or File Shares) on a desired frequency and with a desired retention period. For Azure Virtual Machines, you can use Azure Policy to enable automatic backups.

Reference: Backup and restore options for virtual machines in Azure

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Azure Backup should be enabled for Virtual Machines	Ensure protection of your Azure Virtual Machines by enabling Azure Backup. Azure Backup is a secure and cost effective data protection solution for Azure.	AuditIfNotExists, Disabled	3.0.0

Next steps

- See the Microsoft cloud security benchmark overview
- Learn more about Azure security baselines

Last updated on 02/25/2025

Trusted Launch for Azure virtual machines

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure offers Trusted Launch as a seamless way to improve the security of Generation 2 virtual machines (VM). Trusted Launch protects against advanced and persistent attack techniques. Trusted Launch is composed of several coordinated infrastructure technologies that can be enabled independently. Each technology provides another layer of defense against sophisticated threats.

Trusted Launch is supported for both x64 and Arm64 architectures.

> **ⓘ Important**
>
> - Trusted Launch is the default state for newly created Azure Gen2 VM and scale sets. See the Trusted Launch FAQs if your new VM requires features that aren't supported with Trusted launch.
> - You can upgrade existing Azure Gen1 VMs to Gen2-Trusted launch to enable Secure Boot and vTPM. See Upgrade existing Gen1 VMs to Gen2-Trusted launch.
> - Existing VMs can have Trusted Launch enabled after being created. For more information, see Enable Trusted Launch on existing Gen2 VMs.
> - Existing virtual machine scale sets can have Trusted Launch enabled after being created. For more information, see Enable Trusted Launch on existing scale set.

Benefits

- Securely deploy VMs with verified boot loaders, operating system (OS) kernels, and drivers.
- Securely protect keys, certificates, and secrets in the VMs.
- Gain insights and confidence of the entire boot chain's integrity.
- Ensure that workloads are trusted and verifiable.

Virtual machines sizes

Expand table

Type	Supported size families	Currently not supported size families	Not supported size families
General purpose	B-family, D-family, Dpsv6-series[1], Dplsv6-series[1]	Dpsv5-series, Dpdsv5-series, Dplsv5-series, Dpldsv5-series	A-family, Dv2-series, Dv3-series, DC-Confidential-family
Compute optimized	F-family, Fx-family	All sizes supported.	
Memory optimized	E-family, Eb-family, Epsv6-series[1]	M-family	EC-Confidential-family
Storage optimized	L-family	All sizes supported.	
GPU	NC-family, ND-family, NV-family	NDasrA100_v4-series, NDm_A100_v4-series	NC-series, NV-series, NP-series
High Performance Compute	HBv2-series[2], HBv3-series, HBv4-series, HBv5-series, HC-series[3], HX-series	All sizes supported.	

[1]Arm64 Cobalt 100-based sizes that support Trusted Launch.

[2]HBv2-series is currently supported for Trusted Launch but is scheduled for retirement on May 31, 2027. For new HPC Trusted Launch deployments, prefer HBv5-series, HX-series, HBv4-series, or HBv3-series sizes.

[3]HC-series sizes (Standard_HC44rs, Standard_HC44-16rs, Standard_HC44-32rs) are scheduled for retirement on May 31, 2027. After this date, remaining HC-series VMs will be deallocated and stop incurring charges, and HC-series will no longer have SLA or support. Sales of 1-year and 3-year Reserved Instances ended April 2, 2026. For new HPC Trusted Launch deployments, consider HBv5-series for higher performance and better price-performance, or HX-series for high-memory HPC workloads. Plan to transition off HC-series well before the retirement date to avoid disruption.

> **Note**
>
> - Installation of the *CUDA & GRID drivers on Secure Boot-enabled Windows VMs* doesn't require any extra steps.
> - Installation of the *CUDA driver on Secure Boot-enabled Ubuntu VMs* requires extra steps. For more information, see Install NVIDIA GPU drivers on N-series VMs

running Linux. Secure Boot should be disabled for installing CUDA drivers on other Linux VMs.

- Installation of the *GRID driver* requires Secure Boot to be disabled for Linux VMs.
- *Not supported* size families don't support Generation 2 VMs. Change the VM size to equivalent *supported size families* for enabling Trusted Launch.

Operating systems supported

Expand table

OS	Version
Alma Linux	8.7, 8.8, 9.0
Azure Linux	1.0, 2.0
Debian	11, 12
Oracle Linux	8.3, 8.4, 8.5, 8.6, 8.7, 8.8 LVM, 9.0, 9.1 LVM
Red Hat Enterprise Linux	8.6, 8.8, 8.10, 9.4, 9.5, 9.6
Rocky Linux from CIQ	8.6, 8.10, 9.2, 9.4, 9.6
SUSE Enterprise Linux	15SP3, 15SP4, 15SP5
Ubuntu Server	18.04 LTS, 20.04 LTS, 22.04 LTS, 23.04, 23.10
Windows 10	Pro, Enterprise, Enterprise Multi-Session *
Windows 11	Pro, Enterprise, Enterprise Multi-Session *
Windows Server	2016, 2019, 2022, 2022-Azure-Edition, 2025, 2025-Azure-Edition *

* Variations of this OS are supported.

Note

Trusted Launch on Arm64 is supported when using applicable Arm64 Marketplace images for supported distributions and versions. For Cobalt 100 sizes, deploy Trusted Launch using Arm64 images available in Azure Marketplace.

More information

Regions:

- All public regions
- All Azure Government regions
- All Azure China regions

Pricing: Trusted Launch doesn't increase existing VM pricing costs.

Unsupported features

Currently, the following VM features aren't supported with Trusted Launch:

- Managed Image (customers are encouraged to use Azure Compute Gallery).
- Linux VM Hibernation

Secure Boot

At the root of Trusted Launch is Secure Boot for your VM. Secure Boot, which is implemented in platform firmware, protects against the installation of malware-based rootkits and boot kits. Secure Boot works to ensure that only signed operating systems and drivers can boot. It establishes a "root of trust" for the software stack on your VM.

With Secure Boot enabled, all OS boot components (boot loader, kernel, kernel drivers) require trusted publishers signing. Both Windows and select Linux distributions support Secure Boot. If Secure Boot fails to authenticate that the image is signed with a trusted publisher, the VM fails to boot. For more information, see Secure Boot.

vTPM

Trusted Launch also introduces virtual Trusted Platform Module (vTPM) for Azure VMs. This virtualized version of a hardware Trusted Platform Module is compliant with the TPM2.0 spec. It serves as a dedicated secure vault for keys and measurements.

Trusted Launch provides your VM with its own dedicated TPM instance that runs in a secure environment outside the reach of any VM. The vTPM enables attestation by measuring the entire boot chain of your VM (UEFI, OS, system, and drivers).

Trusted Launch uses the vTPM to perform remote attestation through the cloud. Attestations enable platform health checks and are used for making trust-based decisions. As a health check, Trusted Launch can cryptographically certify that your VM booted correctly.

If the process fails, possibly because your VM is running an unauthorized component, Microsoft Defender for Cloud issues integrity alerts. The alerts include details on which components failed to pass integrity checks.

Virtualization-based security

Virtualization-based security (VBS) uses the hypervisor to create a secure and isolated region of memory. Windows uses these regions to run various security solutions with increased protection against vulnerabilities and malicious exploits. Trusted Launch lets you enable hypervisor code integrity (HVCI) and Windows Defender Credential Guard.

HVCI is a powerful system mitigation that protects Windows kernel-mode processes against injection and execution of malicious or unverified code. It checks kernel mode drivers and binaries before they run, preventing unsigned files from loading into memory. Checks ensure that executable code can't be modified after it's allowed by HVCI to load. For more information about VBS and HVCI, see Virtualization-based security and hypervisor-enforced code integrity.

With Trusted Launch and VBS, you can enable Windows Defender Credential Guard. Credential Guard isolates and protects secrets so that only privileged system software can access them. It helps prevent unauthorized access to secrets and credential theft attacks, like Pass-the-Hash attacks. For more information, see Credential Guard.

Microsoft Defender for Cloud integration

Trusted Launch is integrated with Defender for Cloud to ensure that your VMs are properly configured. Defender for Cloud continually assesses compatible VMs and issues relevant recommendations:

- **Recommendation to enable Secure Boot**: The Secure Boot recommendation only applies for VMs that support Trusted Launch. Defender for Cloud identifies VMs that have Secure boot disabled. It issues a low-severity recommendation to enable it.
- **Recommendation to enable vTPM**: If vTPM is enabled for VM, Defender for Cloud can use it to perform guest attestation and identify advanced threat patterns. If Defender for Cloud identifies VMs that support Trusted Launch with vTPM disabled, it issues a low-severity recommendation to enable it.
- **Recommendation to install guest attestation extension**: If your VM has Secure Boot and vTPM enabled but it doesn't have the Guest Attestation extension installed, Defender for

Cloud issues low-severity recommendations to install the Guest Attestation extension on it. This extension allows Defender for Cloud to proactively attest and monitor the boot integrity of your VMs. Boot integrity is attested via remote attestation.

- **Attestation health assessment or boot integrity monitoring**: If your VM has Secure Boot and vTPM enabled and the Attestation extension installed, Defender for Cloud can remotely validate that your VM booted in a healthy way. This practice is known as boot integrity monitoring. Defender for Cloud issues an assessment that indicates the status of remote attestation.

  If your VMs are properly set up with Trusted Launch, Defender for Cloud can detect and alert you of VM health problems.

- **Alert for VM attestation failure**: Defender for Cloud periodically performs attestation on your VMs. The attestation also happens after your VM boots. If the attestation fails, it triggers a medium-severity alert.

  > ⓘ **Note**
  >
  > VM client boot attestation alerts surfaced in Microsoft Defender for Cloud are informational and are not currently presented in the Defender portal.

  VM attestation can fail for the following reasons:

  - The attested information, which includes a boot log, deviates from a trusted baseline. Any deviation can indicate that untrusted modules are loaded, and the OS could be compromised.
  - The attestation quote couldn't be verified to originate from the vTPM of the attested VM. An unverified origin can indicate that malware is present and could be intercepting traffic to the vTPM.

    > ⓘ **Note**
    >
    > Alerts are available for VMs with vTPM enabled and the Attestation extension installed. Secure Boot must be enabled for attestation to pass. Attestation fails if Secure Boot is disabled. If you must disable Secure Boot, you can suppress this alert to avoid false positives.

- **Alert for untrusted Linux kernel module**: For Trusted Launch with Secure Boot enabled, it's possible for a VM to boot even if a kernel driver fails validation and is prohibited from loading. If kernel driver validation failure happens, Defender for Cloud issues low-severity alerts. While there's no immediate threat, because the untrusted driver didn't load, these events should be investigated. Ask yourself:
  - Which kernel driver failed? Am I familiar with the failed kernel driver and do I expect it to load?
  - Is the exact version of the driver same as expected? Are the driver binaries intact? If failed driver is a partner driver, did the partner pass the OS compliance tests to get it signed?

(Preview) Trusted Launch as default

> **Important**
>
> Trusted Launch default is currently in preview. This Preview is intended for testing, evaluation, and feedback purposes only. Production workloads aren't recommended. When registering to preview, you agree to the supplemental terms of use. Some aspects of this feature might change with general availability (GA).

Trusted Launch as default (TLaD) is available in preview for new Gen2 Virtual machines (VM) and Virtual machine scale sets (scale sets).

TLaD is a fast and zero-touch means of improving the security posture of new Gen2 based Azure VM and Virtual Machine Scale Sets deployments. With Trusted Launch as default, any new Gen2 VMs or scale sets created through any client tools (like ARM template, Bicep) defaults to Trusted Launch VMs with secure boot and vTPM enabled.

The public preview release allows you to validate these changes in your respective environment for all new Azure Gen2 VM, scale set, and prepare for this upcoming change.

> **Note**
>
> All new Gen2 VM, scale set, deployments using any client tool (ARM template, Bicep, Terraform, etc.) defaults to Trusted launch post on-boarding to preview. This change does NOT override inputs provided as part of the deployment code.

Enable TLaD preview

Register preview feature `TrustedLaunchByDefaultPreview` under `Microsoft.Compute` namespace on virtual machine subscription. For more information, see Set up preview features in Azure subscription

To create a new Gen2 VM or scale set with Trusted launch default, execute your existing deployment script as is through Azure SDK, Terraform, or another method that isn't Azure portal, CLI, or PowerShell. The new VM or scale set created in the registered subscription results in a Trusted Launch VM or Virtual Machine Scale Set.

VM & scale sets deployments with TLaD preview

Existing behavior

To create Trusted launch VM & scale set, you need to add following securityProfile element in deployment:

JSON

```
"securityProfile": {
    "securityType": "TrustedLaunch",
    "uefiSettings": {
        "secureBootEnabled": true,
        "vTpmEnabled": true,
    }
}
```

Absence of securityProfile element in deployment code deploys VM & scale set without enabling Trusted launch.

Examples

- vm-windows-admincenter – The Azure Resource Manager (ARM) template deploys Gen2 VM without enabling Trusted launch.
- vm-simple-windows – The ARM template deploys Trusted launch VM (without default as `securityProfile` is explicitly added to ARM template)

New behavior

By using API version 2021-11-01 or higher AND on-boarding to preview, absence of `securityProfile` element from deployment will enable Trusted launch by default to new VM & scale set deployed if following conditions are met:

- Source Marketplace OS image supports Trusted launch.

- Source ACG OS image supports and is validated for Trusted launch.
- Source disk supports Trusted launch.
- VM size supports Trusted launch.

The deployment won't default to Trusted launch if one ore more of the listed condition(s) aren't met and complete successfully to create new Gen2 VM & scale set without Trusted launch.

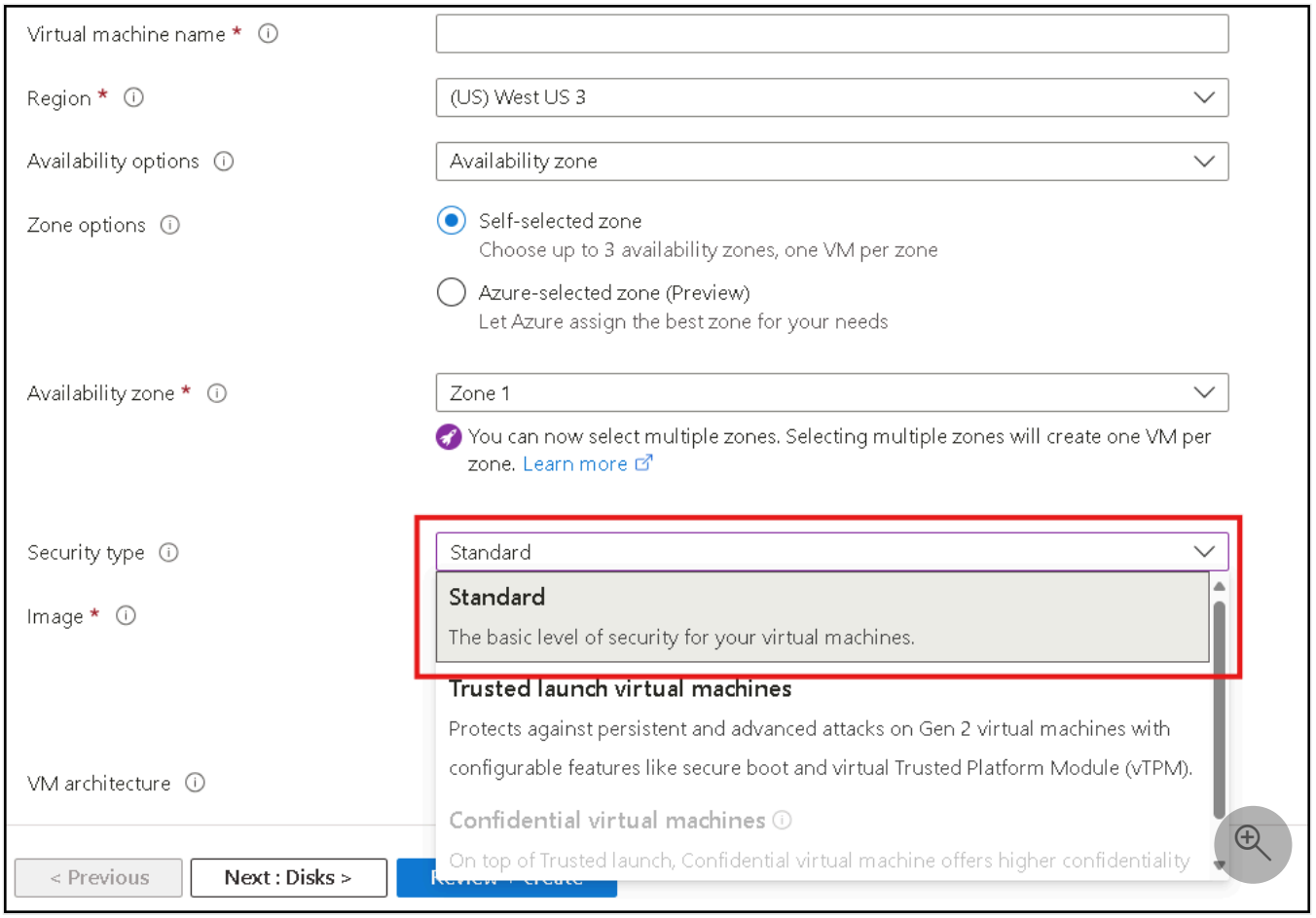
You can choose to explicitly bypass default for VM & scale set deployment by setting `Standard` as value of parameter `securityType`. For more information, see Can I disable Trusted Launch for a new VM deployment.

Known limitations

Unable to bypass Trusted launch default and create Gen2 (Non-Trusted launch) VM using Azure portal after registering to preview.

After registering subscription to preview, setting security type to `Standard` in Azure portal will deploy the VM or scale set `Trusted launch`. This limitation will be addressed prior to the Trusted launch default general availability.

To mitigate this limitation, you can un-register the preview feature by removing feature flag `TrustedLaunchByDefaultPreview` under `Microsoft.Compute` namespace on given subscription.



Unable to re-size VM or VMSS to un-supported Trusted launch VM size family (like M-Series) post default to Trusted launch.

Re-sizing Trusted launch VM to VM size family not supported with Trusted launch will not be supported.

As mitigation, please register feature flag `UseStandardSecurityType` under `Microsoft.Compute` namespace AND roll-back VM from Trusted launch to Gen2-only (Non-Trusted launch) by setting `securityType = Standard` using available client tools (except Azure portal).

TLaD preview feedback

Reach out to us with any feedback, queries, or concerns regarding this upcoming change at Trusted launch default preview feedback survey.

Disable TLaD preview

To disable the TLaD preview, unregister the preview feature `TrustedLaunchByDefaultPreview` under `Microsoft.Compute` namespace on virtual machine subscription. For more information, see Unregister preview feature

Related content

- Deploy a Trusted Launch VM.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/17/2026

Deploy a virtual machine with Trusted Launch enabled

Summarize this article for me

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets.

Trusted Launch is a way to improve the security of Generation 2 virtual machines (VMs). Trusted Launch protects against advanced and persistent attack techniques by combining infrastructure technologies like virtual Trusted Platform Module (vTPM) and secure boot.

Trusted Launch is supported on both x64 and Arm64 Generation 2 VMs where available, including Arm64 marketplace images.

Prerequisites

- We recommend that you onboard your subscription to Microsoft Defender for Cloud if it isn't already. Defender for Cloud has a free tier, which offers useful insights for various Azure and hybrid resources. With the absence of Defender for Cloud, Trusted Launch VM users can't monitor boot integrity of VM.
- If you have existing Generation 1 VMs, you can upgrade them to Generation 2 with Trusted launch. See Upgrade existing Gen1 VMs to Gen2-Trusted launch.
- Assign Azure policy initiatives to your subscription. These policy initiatives need to be assigned only once per subscription. Policies will help deploy and audit for Trusted Launch VMs while automatically installing all required extensions on all supported VMs.
  - Configure the Trusted Launch VMs' built-in policy initiative.
  - Configure prerequisites to enable Guest Attestation on Trusted Launch-enabled VMs.
  - Configure machines to automatically install the Azure Monitor and Azure Security agents on VMs.
- Allow the service tag `AzureAttestation` in network security group outbound rules to allow traffic for Azure Attestation. For more information, see Virtual network service tags.
- Make sure that the firewall policies allow access to `*.attest.azure.net`.

> **Note**
>
> If you're using a Linux image and anticipate that the VM might have kernel drivers either unsigned or not signed by the Linux distro vendor, you might want to consider turning off secure boot. In the Azure portal, on the **Create a virtual machine** page for the `Security`

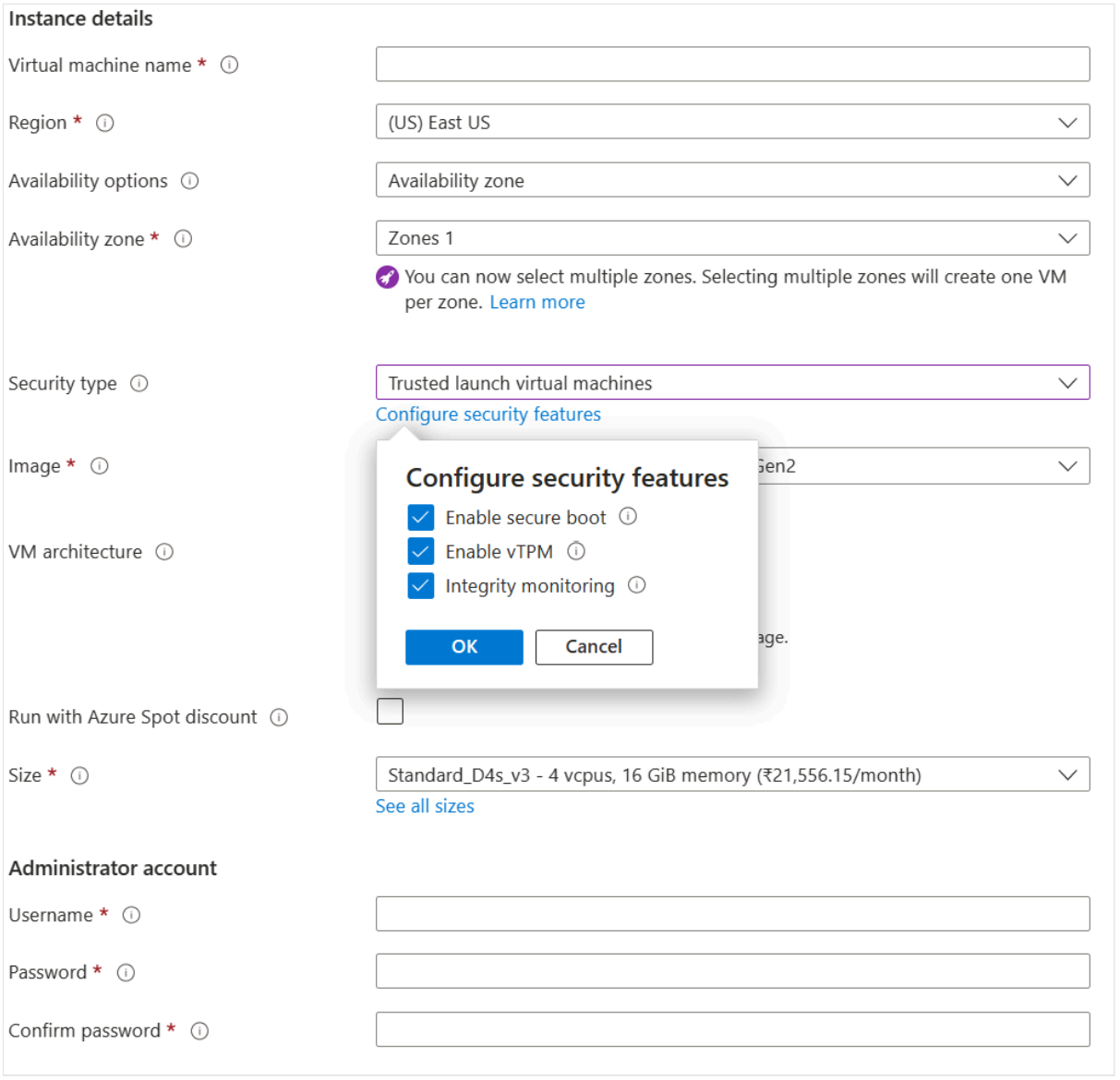
`type` parameter with **Trusted Launch Virtual Machines** selected, select **Configure security features** and clear the **Enable secure boot** checkbox. In the Azure CLI, PowerShell, or SDK, set the secure boot parameter to `false`.

Deploy a Trusted Launch VM

Choose one of the deployment methods to create a new Trusted launch VM

Portal

1. Sign in to the Azure portal.
2. Search for **Virtual Machines**.
3. Under **Services**, select **Virtual machines**.
4. On the **Virtual machines** page, select **Add**, and then select **Virtual machine**.
5. Under **Project details**, make sure the correct subscription is selected.
6. Under **Resource group**, select **Create new**. Enter a name for your resource group or select an existing resource group from the dropdown list.
7. Under **Instance details**, enter a name for the VM name and choose a region that supports Trusted Launch.
8. For **Security type**, select **Trusted launch virtual machines**. When the options **Secure boot**, **vTPM**, and **Integrity Monitoring** appear, select the appropriate options for your deployment. For more information, see Trusted Launch-enabled security features.



9. Under **Image**, select an image from **Recommended Gen 2 images compatible with Trusted launch**. Arm64 Gen2 images that support Trusted Launch are available in the Azure Marketplace. For a list, see Trusted Launch.

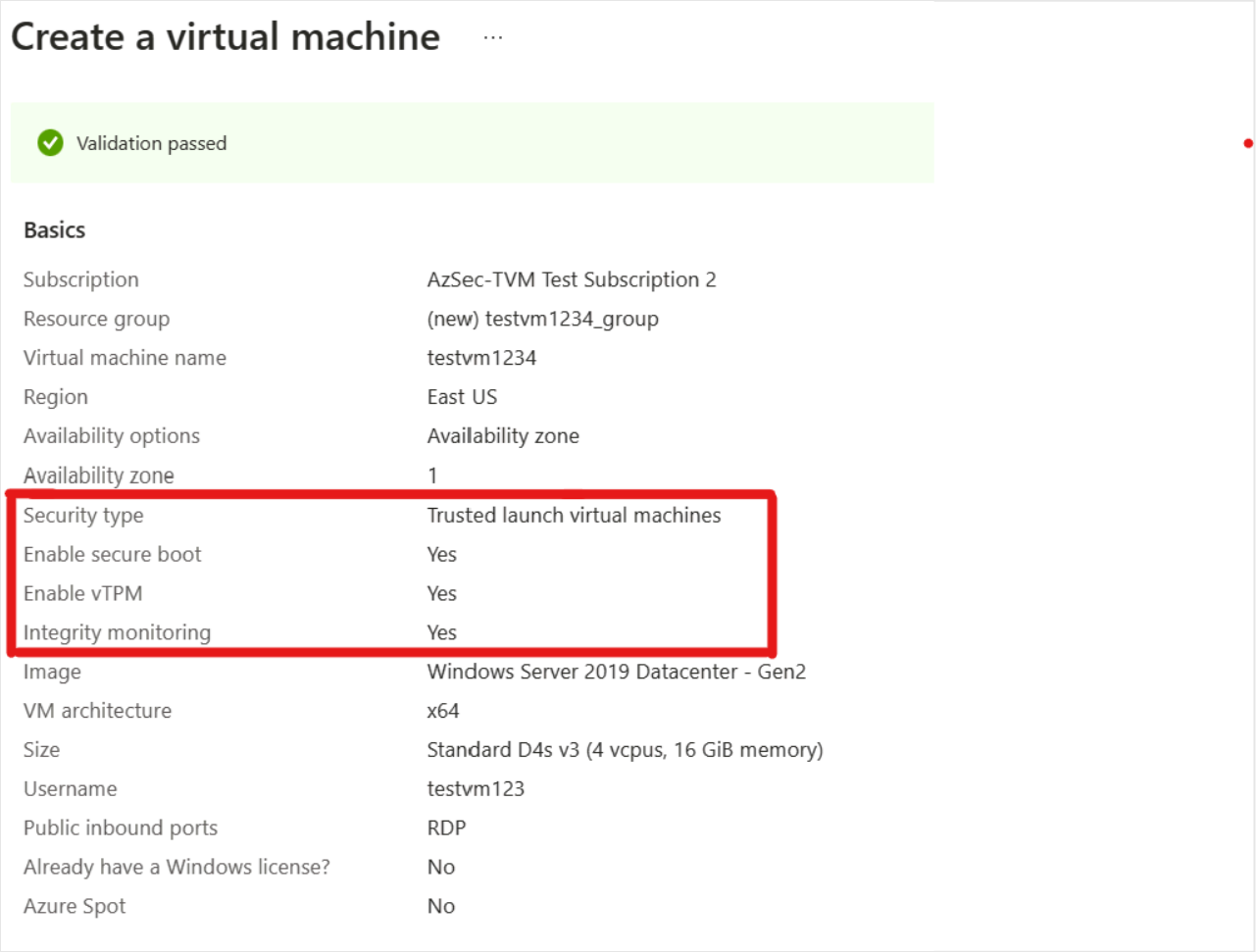
   > **Tip**
   >
   > If you don't see the Gen2 version of the image that you want in the dropdown list, select **See all images**. Then change the **Security type** filter to **Trusted Launch**.

10. Select a VM size that supports Trusted Launch. For more information, see the list of supported sizes.

11. Fill in the **Administrator account** information and then **Inbound port rules**.

12. At the bottom of the page, select **Review + Create**.

13. On the **Create a virtual machine** page, you can see the information about the VM you're about to deploy. After validation shows as passed, select **Create**.



It takes a few minutes for your VM to be deployed.

Deploy a Trusted Launch VM from an Azure Compute Gallery image

Azure Trusted Launch VMs support the creation and sharing of custom images by using Azure Compute Gallery. There are two types of images that you can create, based on the security types of the image:

- **Recommended**: Trusted Launch VM supported (TrustedLaunchSupported) images are images where the source doesn't have VM Guest state information and can be used to create either Generation 2 VMs or Trusted Launch VMs.
- Trusted Launch VM (TrustedLaunch) images are images where the source usually has VM Guest State information and can be used to create only Trusted Launch VMs.

Arm64 Gen2 images are supported. Ensure the image architecture matches the intended Arm64 VM size when creating Trusted Launch VMs.

Trusted Launch VM supported images

For the following image sources, the security type on the image definition should be set to `TrustedLaunchsupported`:

- Gen2 operating system (OS) disk VHD
- Gen2 Managed Image
- Gen2 Gallery Image version

No VM Guest State information can be included in the image source.

You can use the resulting image version to create either Azure Gen2 VMs or Trusted Launch VMs.

These images can be shared by using Azure Compute Gallery - Direct Shared Gallery and Azure Compute Gallery - Community Gallery.

> **Note**
>
> The OS disk VHD, Managed Image, or Gallery Image version should be created from a Gen2 image that's compatible with Trusted Launch VMs.

Portal

1. Sign in to the Azure portal.
2. Search for and select **VM image versions** in the search bar.
3. On the **VM image versions** page, select **Create**.
4. On the **Create VM image version** page, on the **Basics** tab:
   a. Select the Azure subscription.
   b. Select an existing resource group or create a new resource group.
   c. Select the Azure region.
   d. Enter an image version number.
   e. For **Source**, select either **Storage Blobs (VHD)** or **Managed Image** or another **VM Image Version**.
   f. If you selected **Storage Blobs (VHD)**, enter an OS disk VHD (without the VM Guest state). Make sure to use a Gen2 VHD.
   g. If you selected **Managed Image**, select an existing managed image of a Gen2 VM.
   h. If you selected **VM Image Version**, select an existing Gallery Image version of a Gen2 VM.
   i. For **Target Azure compute gallery**, select or create a gallery to share the image.
   j. For **Operating system state**, select either **Generalized** or **Specialized** depending on your use case. If you're using a managed image as the source, always select

Generalized. If you're using a storage blob (VHD) and want to select **Generalized**, follow the steps to generalize a Linux VHD or generalize a Windows VHD before you continue. If you're using an existing VM image version, select either **Generalized** or **Specialized** based on what's used in the source VM image definition.

k. For **Target VM Image Definition**, select **Create new**.

l. On the **Create a VM image definition** pane, enter a name for the definition. Make sure the security type is set to **Trustedlaunch Supported**. Enter the publisher, offer, and SKU information. Then select **OK**.

5. On the **Replication** tab, enter the replica count and target regions for image replication, if necessary.
6. On the **Encryption** tab, enter SSE encryption-related information, if necessary.
7. Select **Review + Create**.
8. After the configuration is successfully validated, select **Create** to finish creating the image.
9. After the image version is created, select **Create VM**.
10. On the **Create a virtual machine** page, under **Resource group**, select **Create new**. Enter a name for your resource group or select an existing resource group from the dropdown list.
11. Under **Instance details**, enter a name for the VM name and choose a region that supports Trusted Launch.
12. For **Security type**, select **Trusted launch virtual machines**. The **Secure Boot** and **vTPM** checkboxes are enabled by default.
13. Fill in the **Administrator account** information and then **Inbound port rules**.
14. On the validation page, review the details of the VM.
15. After the validation succeeds, select **Create** to finish creating the VM.

Trusted Launch VM images

The security type on the image definition should be set to `TrustedLaunch` for the following image sources:

- Trusted Launch VM capture
- Managed OS disk
- Managed OS disk snapshot

You can use the resulting image version to create Azure Trusted Launch VMs only.

Portal

1. Sign in to the Azure portal.
2. To create an Azure Compute Gallery Image from a VM, open an existing Trusted Launch VM and select **Capture**.
3. On the **Create an Image** page, allow the image to be shared to the gallery as a VM image version. Creation of managed images isn't supported for Trusted Launch VMs.
4. Create a new target Azure Compute Gallery or select an existing gallery.
5. Select the **Operating system state** as either **Generalized** or **Specialized**. If you want to create a generalized image, ensure that you generalize the VM to remove machine-specific information before you select this option. If Bitlocker-based encryption is enabled on your Trusted Launch Windows VM, you might not be able to generalize the same.
6. Create a new image definition by providing a name, publisher, offer, and SKU details. **Security type** for the image definition should already be set to **Trusted launch**.
7. Provide a version number for the image version.
8. Modify replication options, if necessary.
9. At the bottom of the **Create an Image** page, select **Review + Create**. After validation shows as passed, select **Create**.
10. After the image version is created, go to the image version directly. Alternatively, you can go to the required image version through the image definition.
11. On the **VM image version** page, select **+ Create VM** to go to the **Create a virtual machine** page.
12. On the **Create a virtual machine** page, under **Resource group**, select **Create new**. Enter a name for your resource group or select an existing resource group from the dropdown list.
13. Under **Instance details**, enter a name for the VM name and choose a region that supports Trusted Launch.
14. The image and the security type are already populated based on the selected image version. The **Secure Boot** and **vTPM** checkboxes are enabled by default.
15. Fill in the **Administrator account** information and then **Inbound port rules**.
16. At the bottom of the page, select **Review + Create**.
17. On the validation page, review the details of the VM.
18. After the validation succeeds, select **Create** to finish creating the VM.

If you want to use either a managed disk or a managed disk snapshot as a source of the image version (instead of a Trusted Launch VM), follow these steps.

1. Sign in to the Azure portal.
2. Search for **VM Image Versions** and select **Create**.
3. Provide the subscription, resource group, region, and image version number.
4. Select the source as **Disks and/or Snapshots**.

5. Select the OS disk as a managed disk or a managed disk snapshot from the dropdown list.
6. Select a **Target Azure Compute Gallery** to create and share the image. If no gallery exists, create a new gallery.
7. Select the **Operating system state** as either **Generalized** or **Specialized**. If you want to create a generalized image, ensure that you generalize the disk or snapshot to remove machine-specific information.
8. For the **Target VM Image Definition** select **Create new**. In the window that opens, select an image definition name and ensure that **Security type** is set to **Trusted launch**. Provide the publisher, offer, and SKU information and select **OK**.
9. The **Replication** tab can be used to set the replica count and target regions for image replication, if required.
10. The **Encryption** tab can also be used to provide SSE encryption-related information, if required.
11. Select **Create** on the **Review + create** tab to create the image.
12. After the image version is successfully created, select **+ Create VM** to go to the **Create a virtual machine** page.
13. Follow steps 12 to 18 as mentioned earlier to create a Trusted Launch VM by using this image version.

Trusted Launch built-in policies

To help users adopt Trusted Launch, Azure policies are available to help resource owners adopt Trusted Launch. The main objective is to help upgrade Generation 1 and 2 VMs that are Trusted launch capable.

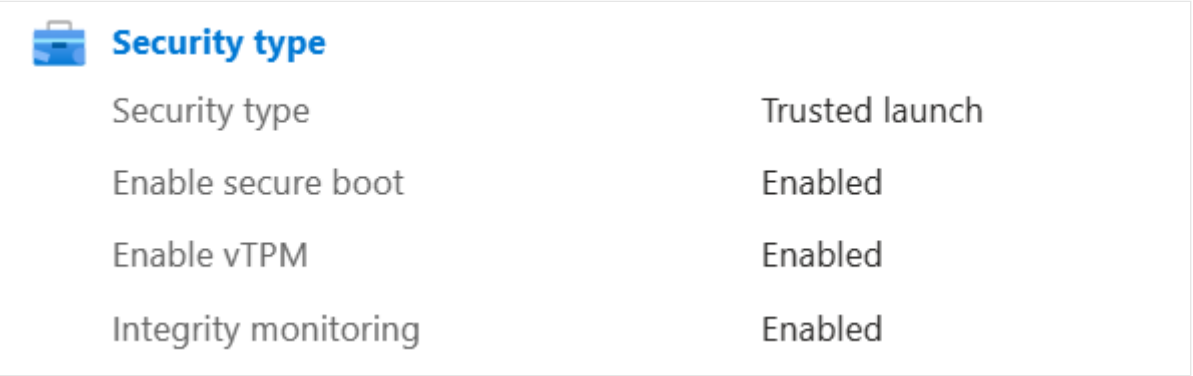
The **Virtual machine should have Trusted launch enabled** single policy checks if the VM is currently enabled with Trusted Launch security configurations. The **Disks and OS supported for Trusted launch** policy checks if previously created VMs have the capable Generation 2 OS and VM size to deploy a Trusted Launch VM.

These two policies come together to make the Trusted Launch policy initiative. This initiative enables you to group several related policy definitions to simplify assignments and management resources to include Trusted Launch configuration.

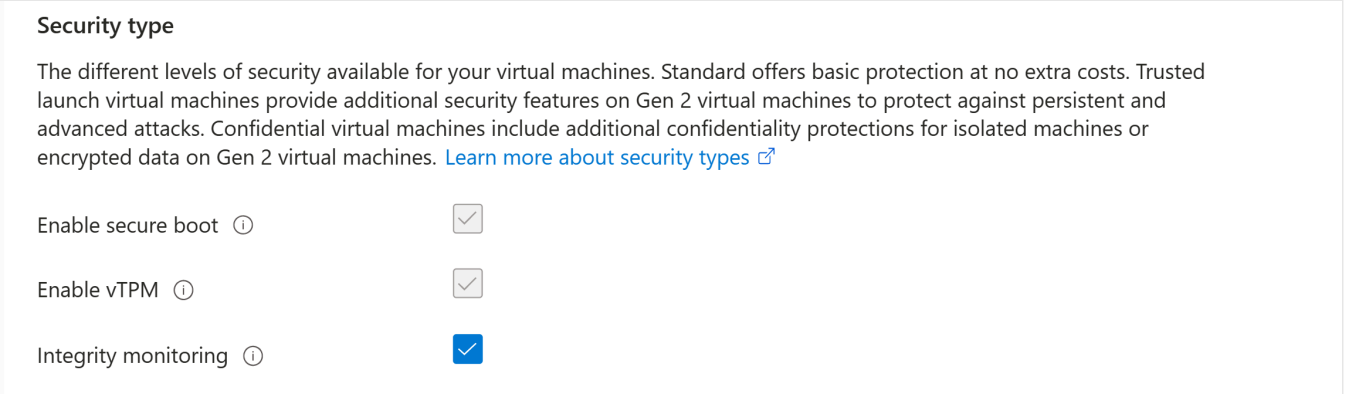
To learn more and start deploying, see Trusted Launch built-in policies.

Verify or update your settings

For VMs created with Trusted Launch enabled, you can view the Trusted Launch configuration by going to the **Overview** page for the VM in the Azure portal. The **Properties** tab shows the status of Trusted Launch features.



To change the Trusted Launch configuration, on the left menu, under **Settings**, select **Configuration**. In the **Security type** section, you can enable or disable **Secure Boot**, **vTPM**, and **Integrity monitoring**. Select **Save** at the top of the page when you're finished.



If the VM is running, you receive a message that the VM will restart. Select **Yes** and then wait for the VM to restart for changes to take effect.

Related content

- Learn more about Trusted Launch and boot integrity monitoring VMs.
- If you have existing VMs or VM scale sets, you can upgrade them to Gen2-Trusted launch. For more information, see Upgrade existing Gen1 VMs to Gen2-Trusted launch, Upgrade existing Gen2 VMs to Gen2-Trusted launch, Upgrade existing Gen1 or Gen2 VM scale sets to Gen2-Trusted launch

---

Last updated on 02/10/2026

Enable Trusted launch on existing Azure Gen2 VMs

Summarize this article for me

Applies to: ✔ Linux VM ✔ Windows VM

Azure Virtual Machines supports enabling Azure Trusted launch on existing Azure Generation 2 virtual machines (VM) by upgrading to the Trusted launch security type.

Trusted launch is a way to enable foundational compute security on Azure Generation 2 VMs and protects against advanced and persistent attack techniques like boot kits and rootkits. It does so by combining infrastructure technologies like Secure Boot, virtual Trusted Platform Module (vTPM), and boot integrity monitoring on your VM.

> ⓘ **Important**
>
> Support for enabling Trusted launch on existing Azure Generation 1 VMs is available. Refer to **Upgrade existing Azure Gen1 VMs to Gen2-Trusted launch**.

Prerequisites

Azure VM is configured with:

- Trusted launch supported size family.
- Trusted launch supported operating system (OS) version. For custom OS images or disks, the base image should be *Trusted launch capable*.
- For Arm64 VMs, use Trusted launch-capable Arm64 images from Azure Marketplace and a supported Arm64 size family. Cobalt 100-based Arm64 sizes support Trusted launch.
- Azure VM isn't using features currently not supported with Trusted launch.
- Azure Backup, if enabled, for VMs should be configured with the Enhanced Backup policy. The Trusted launch security type can't be enabled for VMs configured with *Standard policy* backup protection.
  - Existing Azure VM backup can be migrated from the *Standard* to the *Enhanced* policy. Follow the steps in Migrate Azure VM backups from Standard to Enhanced policy (preview).

Best practices

- Enable Trusted launch on a test Generation 2 VM and determine if any changes are required to meet the prerequisites before you enable Trusted launch on Generation 2 VMs associated with production workloads.
- Create restore points for Azure Generation 2 VMs associated with production workloads before you enable the Trusted launch security type. You can use the restore points to re-create the disks and Generation 2 VM with the previous well-known state.
- *For Linux VMs*, validate secure boot compatibility using `SBInfo` tool. Refer to Linux Trusted launch secure boot validation for distribution-based `SBInfo` installation commands.

Enable Trusted launch on an existing VM

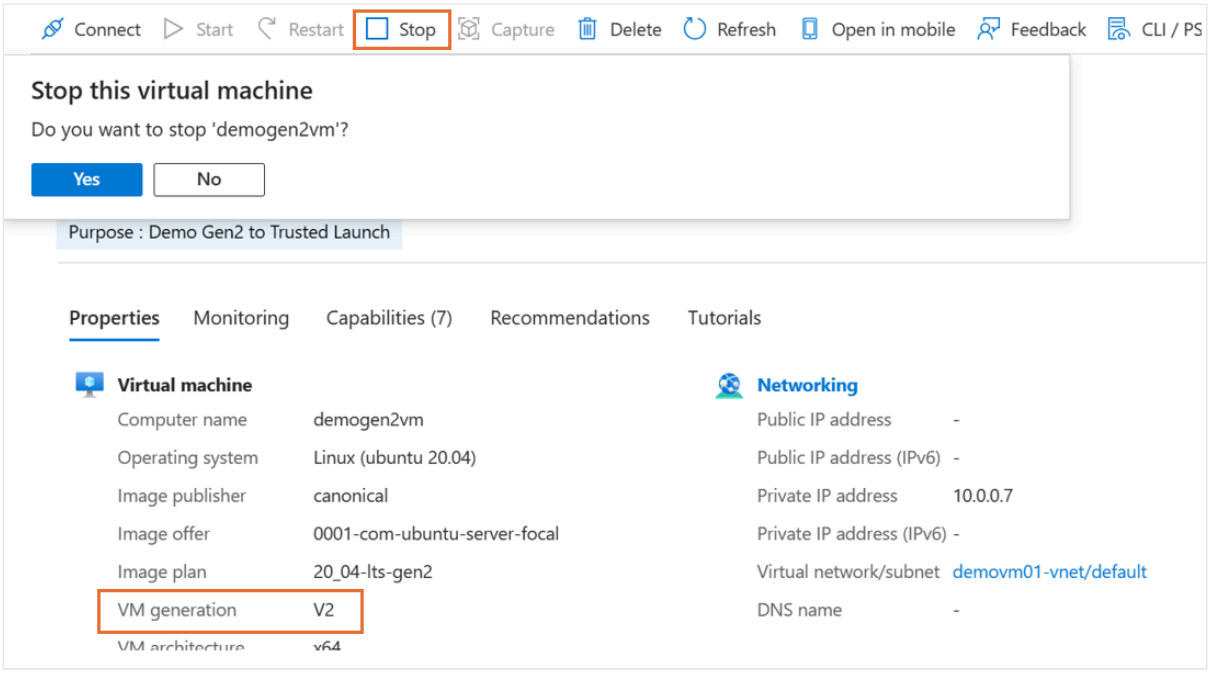
The following steps apply to both x64 and Arm64-based Generation 2 VMs when using Trusted launch-capable images.

> **Note**
>
> - These steps apply equally to x64 and Arm64 Gen2 VMs.
> - vTPM is enabled by default.
> - Secure boot isn't enabled by default. We strongly recommend that you enable Secure Boot, if you aren't using custom unsigned kernel or drivers. Secure Boot preserves boot integrity and enables foundational security for VMs.

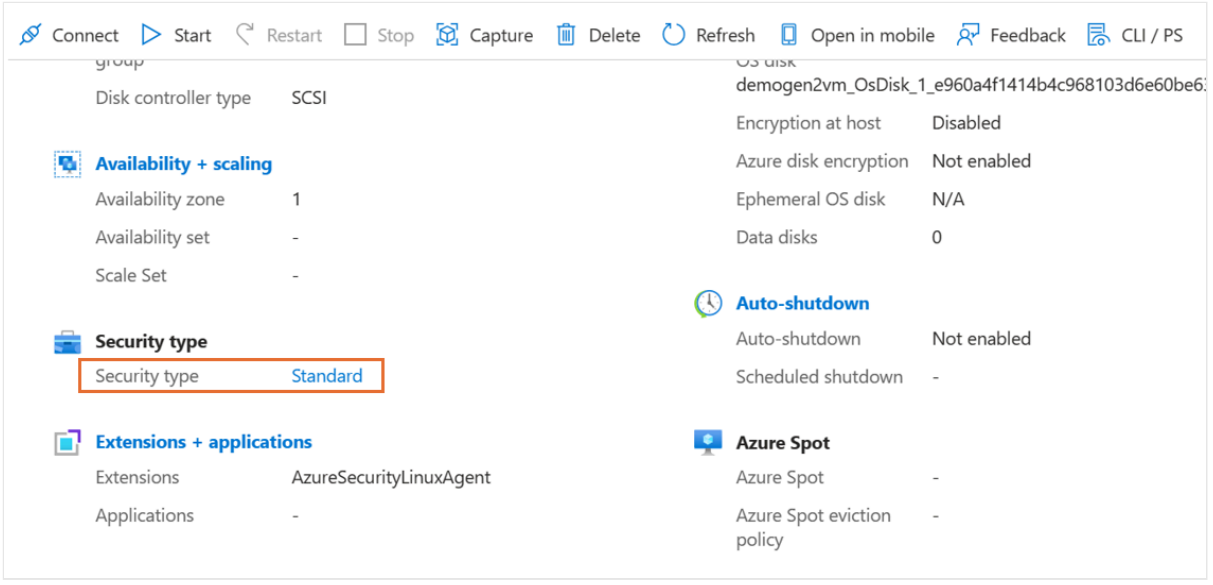
Portal

Enable Trusted launch on an existing Azure Generation 2 VM by using the Azure portal.

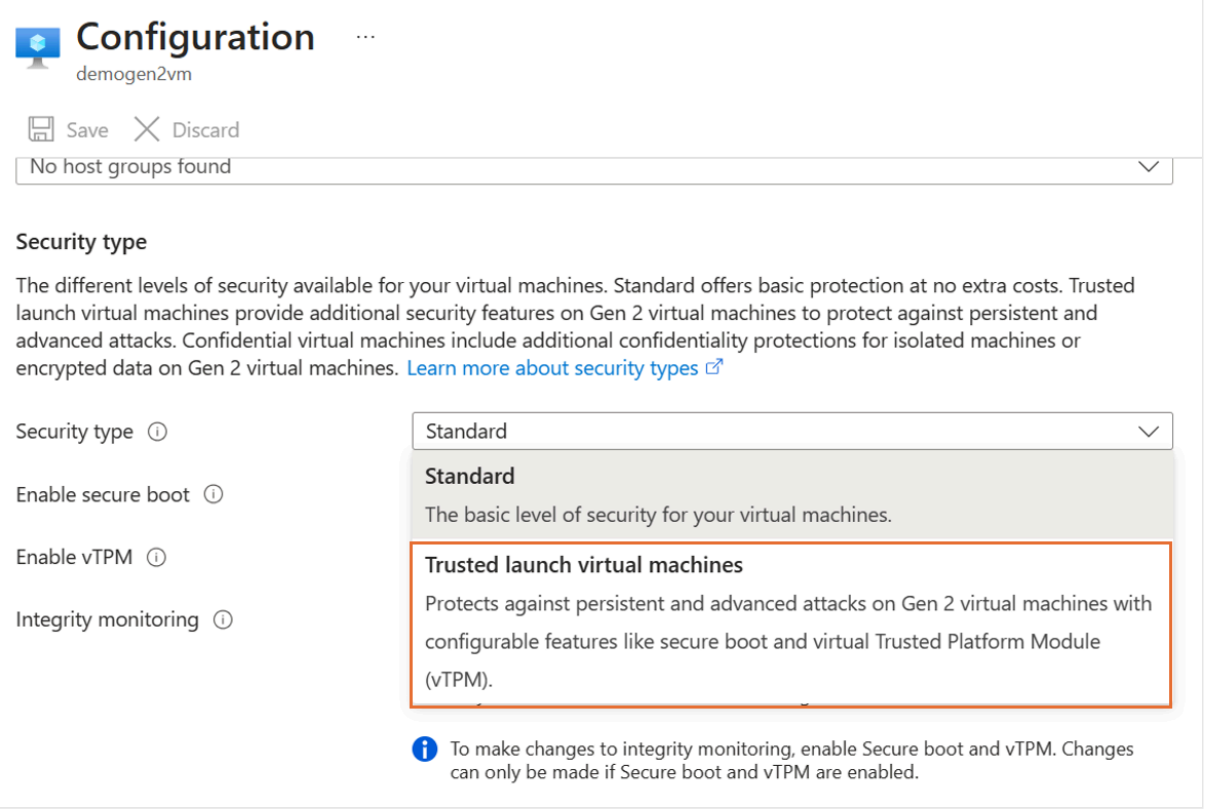
1. Sign in to the Azure portal.
2. Confirm that the VM generation is **V2** and select **Stop** for the VM.



3. On the **Overview** page in the VM properties, under **Security type**, select **Standard**. The **Configuration** page for the VM opens.

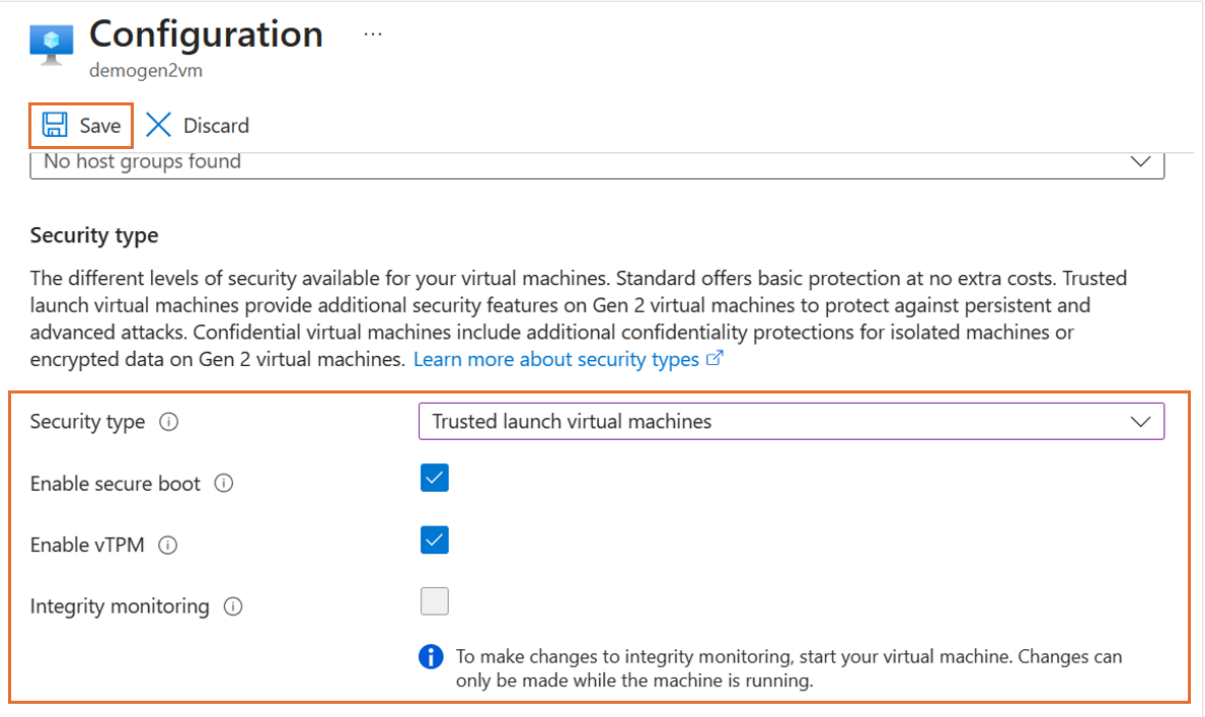


4. On the **Configuration** page, under the **Security type** section, select the **Security type** dropdown list.

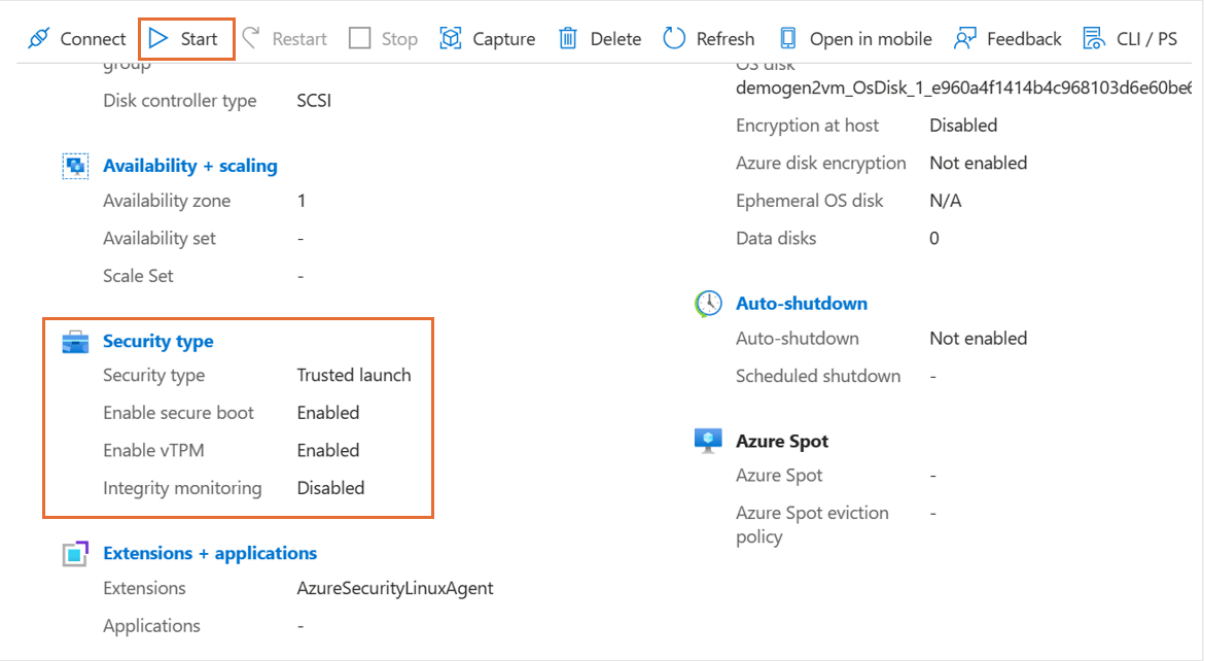


5. Under the dropdown list, select **Trusted launch**. Select checkboxes to enable **Secure Boot** and **vTPM**. After you make the changes, select **Save**.

> **Note**
>
> - Generation 2 VMs created by using Azure Compute Gallery (ACG), Managed image, or an OS disk can't be upgraded to Trusted launch by using the portal. Ensure that the OS version is supported for Trusted launch. Use PowerShell, the Azure CLI, or an Azure Resource Manager template (ARM template) to run the upgrade.



6. After the update successfully finishes, close the **Configuration** page. On the **Overview** page in the VM properties, confirm the **Security type** settings.



7. Start the upgraded Trusted launch VM. Verify that you can sign in to the VM by using either the Remote Desktop Protocol (RDP) for Windows VMs or the Secure Shell Protocol (SSH) for Linux VMs.

Roll back

> **Note**

Register feature `UseStandardSecurityType` under `Microsoft.Compute` namespace on virtual machine subscription for roll-back support. For more information, see Set up preview features in Azure subscription

To roll-back changes from Trusted launch to previous Gen2 known good configuration, you need to set `securityType` of VM to **Standard**.

Portal

Roll-back of Trusted launch to Gen2 (Non-Trusted launch) configuration is currently not supported in Azure portal.

Azure Advisor Recommendation

Azure Advisor populates an **Enable Trusted launch foundational excellence, and modern security for Existing Generation 2 VM(s)** operational excellence recommendation for existing Generation 2 VMs to adopt Trusted launch, a higher security posture for Azure VMs at no extra cost to you. Ensure Generation 2 VM has all prerequisites to migrate to Trusted launch, follow all the best practices including validation of OS image, VM Size, and creating restore points. For the Advisor recommendation to be considered complete, follow the steps outlined in the **Enable Trusted launch on an existing VM** to upgrade the virtual machines security type and enable Trusted launch.

What if there is Generation 2 VMs, that doesn't fit the prerequisites for Trusted launch?

For a Generation 2 VM that doesn't meet the prerequisites to upgrade to Trusted launch, look how to fulfill the prerequisites. For example, If using a virtual machine size not supported, look for an equivalent Trusted launch supported size that supports Trusted launch.

> **Note**
>
> Dismiss the recommendation if Gen2 virtual machine is configured with VM size families that are currently not supported with Trusted launch like MSv2-series.

Related content

- Refer to Deploy Trusted launch virtual machines for enabling Trusted launch on new virtual machine & scale set deployments.

- Refer to boot integrity monitoring for enabling boot integrity monitoring and monitor the health of the VM by using Microsoft Defender for Cloud.
- Learn more about Trusted launch and review frequently asked questions.
- Learn about Cobalt 100-based Arm64 VM sizes that support Trusted Launch.

Last updated on 02/10/2026

Upgrade existing Azure Gen1 VMs to Trusted launch

Summarize this article for me

Applies to: ✔ Linux VM ✔ Windows VM ✔ Generation 1 VM

Azure Virtual Machines supports upgrading Generation 1 virtual machines (VM) to Generation 2 by upgrading to the Trusted launch security type.

Trusted launch is a way to enable foundational compute security on Azure Generation 2 VMs and protects against advanced and persistent attack techniques like boot kits and rootkits. It does so by combining infrastructure technologies like Secure Boot, virtual Trusted Platform Module (vTPM), and boot integrity monitoring on your VM.

> **ⓘ Important**
>
> - *Upgrade of Gen1 VMs to Gen2 without enabling Trusted launch* is **not supported**.
> - This upgrade path doesn't change the VM CPU architecture. To use Arm64 with Trusted launch, create a new VM from an Arm64 Trusted launch image in Azure Marketplace on supported **Cobalt 100 sizes** .

Prerequisites

- Azure VM is configured with:
  - Trusted launch supported size family.
  - Trusted launch supported operating system (OS) version (*excluding Windows Server 2016, Debian, Azure Linux*). For custom OS images or disks, the base image should be *Trusted launch capable*.
- Azure VM isn't using features currently not supported with Trusted launch.
- Azure Backup, if enabled, for VMs should be configured with the Enhanced Backup policy. The Trusted launch security type can't be enabled for VMs configured with *Standard policy* backup protection.
  - Existing Azure VM backup can be migrated from the *Standard* to the *Enhanced* policy. Follow the steps in Migrate Azure VM backups from Standard to Enhanced policy.
- Upgrade a test Gen1 VM to Trusted launch and determine if any changes are required to meet the prerequisites before you upgrade Gen1 VMs associated with production workloads to Trusted launch.
- Disable any *Windows OS volume encryption* including BitLocker before upgrade if enabled. All Windows OS volume encryptions should be re-enabled post successful

upgrade. This action isn't required for data disks or Linux OS volume.

Unsupported Gen1 VM configurations

Gen1 to Trusted launch VM upgrade is **NOT** supported if Gen1 VM is configured with:

- **Operating system**: Windows Server 2016, Azure Linux, Debian, and any other operating system not listed under Trusted launch supported operating system (OS) version. For *Windows Server 2016 only*, workaround is to update the Guest OS to Windows Server 2019 or 2022.
- **VM size**: Gen1 VM configured with VM size not listed under Trusted launch supported size families. As workaround, update the VM size to Trusted launch supported VM size family.
- **Azure Backup**: Gen1 VM configured with Azure Backup using *Standard policy*. As workaround, migrate Gen1 VM backups from Standard to Enhanced policy.
- **BitLocker or equivalent encryption**: Windows Gen1 VM *guest OS volume* is encrypted using BitLocker or equivalent encryption technology. As workaround, disable Windows OS volume encryption before upgrade and re-enable post successful completion of Trusted launch upgrade.

Best practices

> **Important**
>
> VM once upgraded to Trusted launch can't be rolled back to Gen1 configuration. It has to be fully recovered using Backup taken before upgrade.

- Take full backup OR Create restore points for Azure VMs associated with production workloads before you enable the Trusted launch security type. You can use the backup OR restore points to re-create the disks and VM with the previous Gen1 configuration well-known state.
- Review known issues and roll-back steps before executing Trusted launch upgrade.
- You won't be able to extend Windows OS disk system volume after `MBR to GPT conversion` as part of upgrade. Recommendation is to extend system volume for future before upgrading to Trusted launch.
- *Windows OS disk volume* should be defragmented using command `Defrag C: /U /V`. Defragmentation of OS volume reduces the risk of MBR (Master boot record) to GPT (GUID partition table) conversion failure by freeing up end of partitions. Refer to defrag.

- *For Linux VMs*, validate secure boot compatibility using `SBInfo` tool. Refer to Linux Trusted launch secure boot validation for distribution-based `SBInfo` installation commands.

Update Guest OS volume

Generation 2 VMs requires guest OS volume with following configurations:

- **GPT Disk layout**: Guest OS volume of Gen1 VMs is mostly set to `MBR` and requires to be updated to `GPT`
- **EFI system partition**: Guest OS volume of Gen1 VMs mostly doesn't include this partition.

Complete the update of OS volume with required configuration before upgrading Azure Gen1 VM to Trusted launch.

> **ⓘ Important**
>
> - Upgrade a test Gen1 VM to Trusted launch and determine if any changes are required to meet the prerequisites before you upgrade Gen1 VMs associated with production workloads to Trusted launch.

Windows

Using in-built MBR2GPT.exe utility, you can enable `GPT` disk layout AND add `EFI system partition` required for Gen2 upgrade.

> **⊗ Caution**
>
> You won't be able to extend Windows OS disk system volume after `MBR to GPT conversion`. Recommendation is to extend system volume for future before executing the upgrade.

> **ⓘ Note**
>
> Windows Server 2016 doesn't support `MBR2GPT.exe`. Workaround is to update the Guest OS to Windows Server 2019 or 2022 and followed by `MBR to GPT conversion`. Refer to In-place upgrade for VMs running Windows Server in Azure.

1. Remotely connect (using RDP or command-line) to Gen1 Windows VM for executing conversion.
2. Run command `MBR2GPT /validate /allowFullOS` and ensure `Disk layout validation` completes successfully. **Do not proceed** if the `Disk layout validation` fails. Refer to Known issues for list of common causes and associated resolution for failure. For more information and troubleshooting, see MBR2GPT troubleshooting.
3. Run command `MBR2GPT /convert /allowFullOS` to **execute** MBR to GPT conversion. Sample successful output:

   log

   ```
   If conversion is successful the disk can only be booted in GPT mode.
   These changes cannot be undone!

   MBR2GPT: Attempting to convert disk 0
   MBR2GPT: Retrieving layout of disk
   MBR2GPT: Validating layout, disk sector size is: 512 bytes
   MBR2GPT: Trying to shrink the OS partition
   MBR2GPT: Creating the EFI system partition
   MBR2GPT: Installing the new boot files
   MBR2GPT: Performing the layout conversion
   MBR2GPT: Migrating default boot entry
   MBR2GPT: Adding recovery boot entry
   MBR2GPT: Fixing drive letter mapping
   MBR2GPT: Conversion completed successfully
   MBR2GPT: Before the new system can boot properly you need to switch the
   firmware to boot to UEFI mode!
   ```

Upgrade Gen1 VM to Trusted launch

> ⓘ **Note**
>
> - vTPM is enabled by default.
> - Secure boot isn't enabled by default. We strongly recommend that you enable Secure Boot, if you aren't using custom unsigned kernel or drivers. Secure Boot preserves boot integrity and enables foundational security for VMs.

PowerShell

Make sure that you install the latest Azure PowerShell and are signed in to an Azure account with Connect-AzAccount.

Follow the steps to upgrade existing Gen1 VM to Gen2 and enable Trusted launch by using Azure PowerShell.

1. Sign in to the VM Azure subscription.

Azure PowerShell

```
Connect-AzAccount -SubscriptionId 00000000-0000-0000-0000-000000000000
```

2. Deallocate the VM.

Azure PowerShell

```
Stop-AzVM -ResourceGroupName myResourceGroup -Name myVm
```

3. Enable Trusted launch by setting `-SecurityType` to `TrustedLaunch`.

Azure PowerShell

```
Get-AzVM -ResourceGroupName myResourceGroup -VMName myVm `
    | Update-AzVM -SecurityType TrustedLaunch `
        -EnableSecureBoot $true -EnableVtpm $true
```

4. Validate `securityProfile` in the updated VM configuration.

Azure PowerShell

```
# Following command output should be `TrustedLaunch`

(Get-AzVM -ResourceGroupName myResourceGroup -VMName myVm `
    | Select-Object -Property SecurityProfile `
        -ExpandProperty SecurityProfile).SecurityProfile.SecurityType

# Following command output should return `SecureBoot` and `vTPM` settings
(Get-AzVM -ResourceGroupName myResourceGroup -VMName myVm `
    | Select-Object -Property SecurityProfile `
        -ExpandProperty SecurityProfile).SecurityProfile.Uefisettings
```

5. Start the VM.

Azure PowerShell

```
Start-AzVM -ResourceGroupName myResourceGroup -Name myVm
```

6. Start the upgraded Trusted launch VM. Verify that you can sign in to the VM by using either RDP (for Windows VMs) or SSH (for Linux VMs).

Roll back

Azure VM once upgraded to Trusted launch can't be rolled back to Gen1 configuration. You can disable Trusted launch to roll-back VM from Trusted launch to Gen2 (Non-Trusted launch) configuration.

Use the Backup or Restore point of Gen1 VM taken before upgrade and restore entire VM along with disks to roll back fully to Gen1 VM.

Known issues

VM image reference doesn't change post Trusted launch upgrade

Post upgrade of Azure Gen1 VM to Trusted launch, the image reference still reflects source as Gen1 OS image. Image reference not updated is a known limitation.

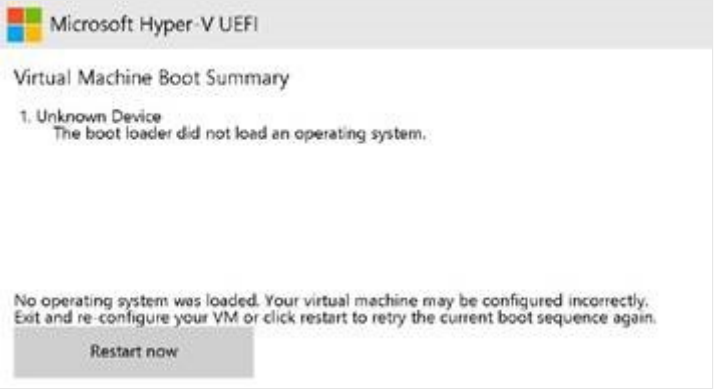
This limitation doesn't impacts the VM or hosted application functionality post Trusted launch upgrade. Following VM operations are impacted due to disconnect between image reference and running OS:

- **Guest patching**: For *Server* OS, Automatic guest patching installs updates based on the image reference of VM.
- **Reimage**: Reimaging VM using Gen1 image reference post Trusted launch upgrade causes VM boot failure.

Windows 11 boot fails after Trusted launch upgrade of Windows 10 VM

Windows 10 Gen1 VM is successfully upgraded to Trusted launch followed by successful Windows 11 in-place upgrade. However, the Windows 11 boot fails after Azure VM is stopped and started with error shown.

Resolution: This issue is fixed with 24H2 build version 26100.2314 and 23H2 build version 22631.5624.



> **Note**
>
> The upgrade can fail if you use outdated Windows 11 installation media. To avoid this issue, ensure you use the most recent Windows 11 image. You can download the latest ISO from the Microsoft Software Download page. If you have a volume licensing agreement, use the Microsoft 365 admin center to access updated Windows 11 images.

Windows Server 2022 boot fails after in-place upgrade of Trusted launch VM migrated from Gen1

Windows Server 2019 Gen1 VMs upgraded to Trusted launch followed by an in-place upgrade to Windows Server 2022 may fail to boot with **"The boot loader did not load an operating system"**. This issue is caused by a partition number mismatch in the UEFI boot entry after MBR-to-GPT conversion and is fixed in Windows Server 2025 and later. Take a snapshot of the OS disk before performing the in-place upgrade.

[Windows] MBR to GPT conversion fails with error Cannot find room for the EFI system partition

This error occurs for one of following reason:

- There's no free space available on the system volume
- System volume is corrupted. You can validate by trying to Shrink Volume by few MBs under Disk Management console. Use command `chkdsk C:/v/f` to repair system volume.
- `Virtual Disk` service isn't running or unable to communicate successfully. Service startup type should be set to `Manual`.

- `Optimize Drives` service isn't running or unable to communicate successfully. Service startup type should be set to `Manual`.
- System volume disk is already configured with four MBR partitions (maximum supported by MBR disk layout). You need to delete one of the partitions to make room for EFI system partition.

    1. Run `ReAgentc /info` to identify partition actively used by Recovery. Example: `Windows RE location: \\?\GLOBALROOT\device\harddisk0\partition4\Recovery\WindowsRE`
    2. Run PowerShell cmdlet `Get-Partition -DiskNumber 0` to identify current partitions that are configured.
    3. Run PowerShell cmdlet `Remove-Partition -DiskNumber 0 -PartitionNumber X` to remove any extra Recovery partition not actively used by Recovery service as identified in Step 1.

[Windows] MBR to GPT failed to update ReAgent.xml

For certain Windows OS versions like Windows 10, MBR2GPT generates pass-through error `MBR2GPT: Failed to update ReAgent.xml, please try to manually disable and enable WinRE.`

The warning indicates that `MBR2GPT.exe` was unable to update Recovery partition GUID in `C:\Windows\System32\Recovery\ReAgent.xml`. This issue shouldn't cause issue with Guest OS functionality or further upgrade operations like Windows 11 Upgrade to succeed or Windows 11 to work properly post Upgrade. `ReAgent.xml` is updated with correct GUID post Windows 11 upgrade, that is, if you don’t take any action between MBR2GPT conversion & Windows 11 upgrade, the GUID value in `ReAgent.xml` is synced with Windows recovery configuration.

To manually fix The warning, you can disable and re-enable WinRE post MBR2GPT conversion or post Gen1 to Trusted launch upgrade before Windows 11 upgrade using listed steps, these steps forces sync the WinRE GUID in `ReAgent.xml`

1. Complete Gen1 -> Trusted launch upgrade.
2. Run `reagentc /disable`
3. Run `reagentc /enable`
4. Run `reagentc /info`
5. Open `C:\Windows\System32\Recovery\ReAgent.xml` and validate BCD GUID in xml file is matching the output from Step 4.

[Windows] D Drive assigned to System Reserved Post upgrade

Temporary storage Drive letter assignment 'D' is changed to 'E' with previous letter assigned to System Reserved post-upgrade of Gen1 VM. Execute below steps manually post-upgrade to work around the issue:

After the upgrade check the disks on the server, if system reserved partition has the letter D, do the following actions:

1. Reconfigure pagefile from D: to C:
2. Reboot the VM
3. Remove letter D: from the partition
4. Reboot the VM to show the temporary storage disk with D: letter

Frequently asked questions

What if there is Generation 1 VMs, that doesn't fit the prerequisites for Trusted launch?

For a Generation 1 VM that doesn't meet the prerequisites to upgrade to Trusted launch, look how to fulfill the prerequisites. For example, If using a virtual machine size not supported, look for an equivalent Trusted launch supported size that supports Trusted launch.

Why is upgrade to Generation 2 only (without Trusted launch) not supported?

Trusted launch enables foundational compute security on Azure Generation 2 VMs with no extra cost associated. Also, Trusted launch VMs are majorly on parity with Generation 2 VMs. Hence, there's no added benefit for upgrading only to Generation 2 VM without enabling Trusted launch.

Related content

- Refer to Deploy Trusted launch virtual machines for enabling Trusted launch on new virtual machine & scale set deployments.
- Refer to boot integrity monitoring for enabling boot integrity monitoring and monitor the health of the VM by using Microsoft Defender for Cloud.
- Learn more about Trusted launch and review frequently asked questions.

Last updated on 03/09/2026

Enable Trusted launch on existing Uniform scale set

06/18/2025

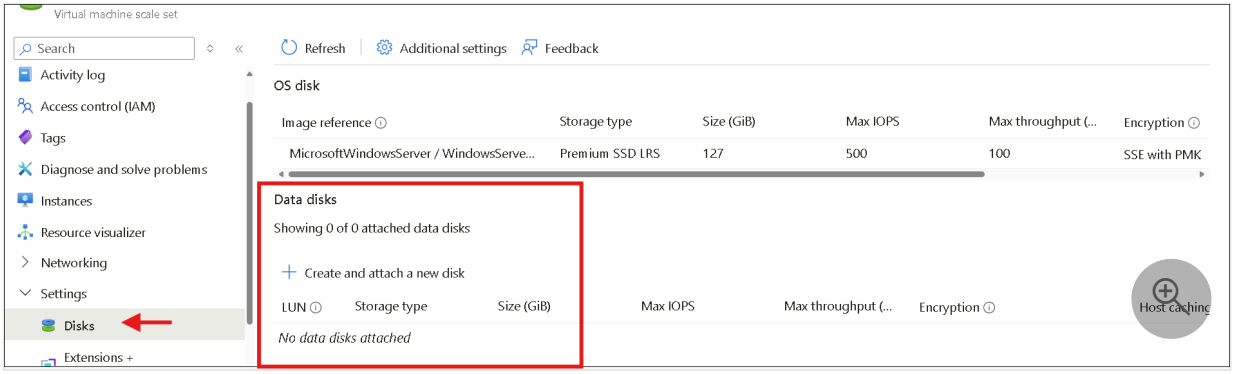
Applies to: ✔ Uniform scale set ✔ Flex scale set ✖ Service fabric

Azure Virtual machine Scale sets supports enabling Trusted launch on existing Uniform Scale sets virtual machine (VM) by upgrading to Trusted launch security type.

Trusted launch enables foundational compute security on Azure Generation 2 virtual machines & scale sets and protects them against advanced and persistent attack techniques like boot kits and rootkits. It does so by combining infrastructure technologies like Secure Boot, vTPM, and Boot Integrity Monitoring on your Scale set.

Limitations

- Enabling Trusted launch on existing virtual machine Scale sets with data disks attached requires upgrade mode set to Rolling upgrade with max surge
  - To validate if scale set is configured with data disk, navigate to scale set -> **Disks** under **Settings** menu -> check under heading **Data disks**



- Enabling Trusted launch on existing virtual machine Scale sets Flex is currently in preview. Register for preview of enabling Trusted launch on existing Flex scale set preview

- Enabling Trusted launch on existing Service fabric clusters and Service fabric managed clusters is currently not supported.

Prerequisites

- Scale set isn't dependent on features currently not supported with Trusted launch.
- Scale set should be configured with Trusted launch supported size family

> **Note**
>
> - Virtual machine size can be changed along with Trusted launch upgrade. Ensure quota for new VM Size is in-place to avoid upgrade failures. Refer to Check vCPU quotas.
> - Change to Virtual machine size re-creates the Virtual machine instance with new size and requires downtime of individual Virtual machine instance. It can be done in a Rolling Upgrade fashion to avoid Scale set downtime.

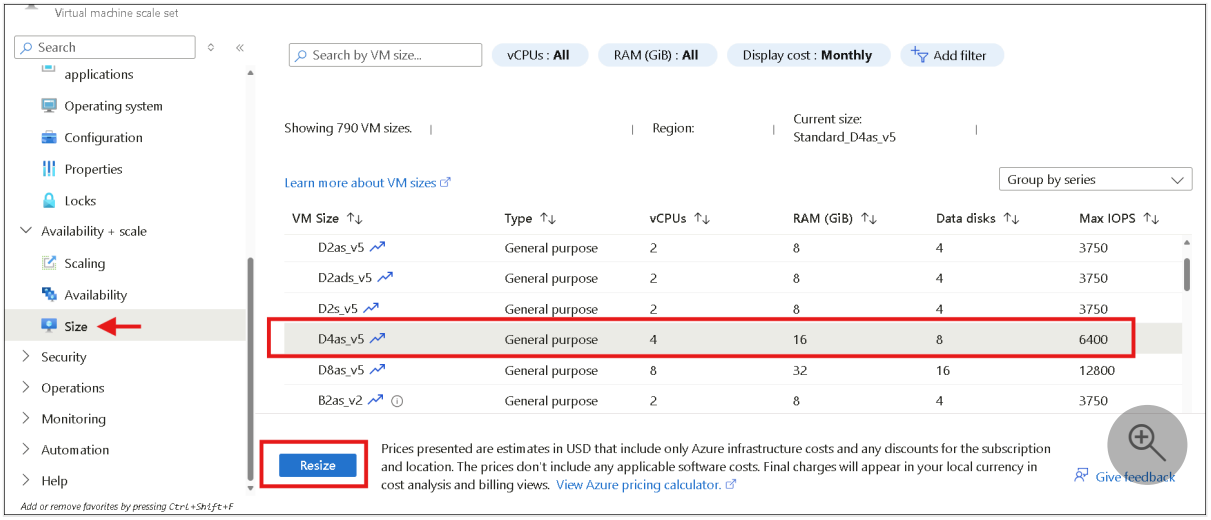
- Scale set should be configured with Trusted launch supported OS Image. For Azure compute gallery OS image, ensure image definition is marked as TrustedLaunchSupported

Enable Trusted launch on existing Scale set Uniform

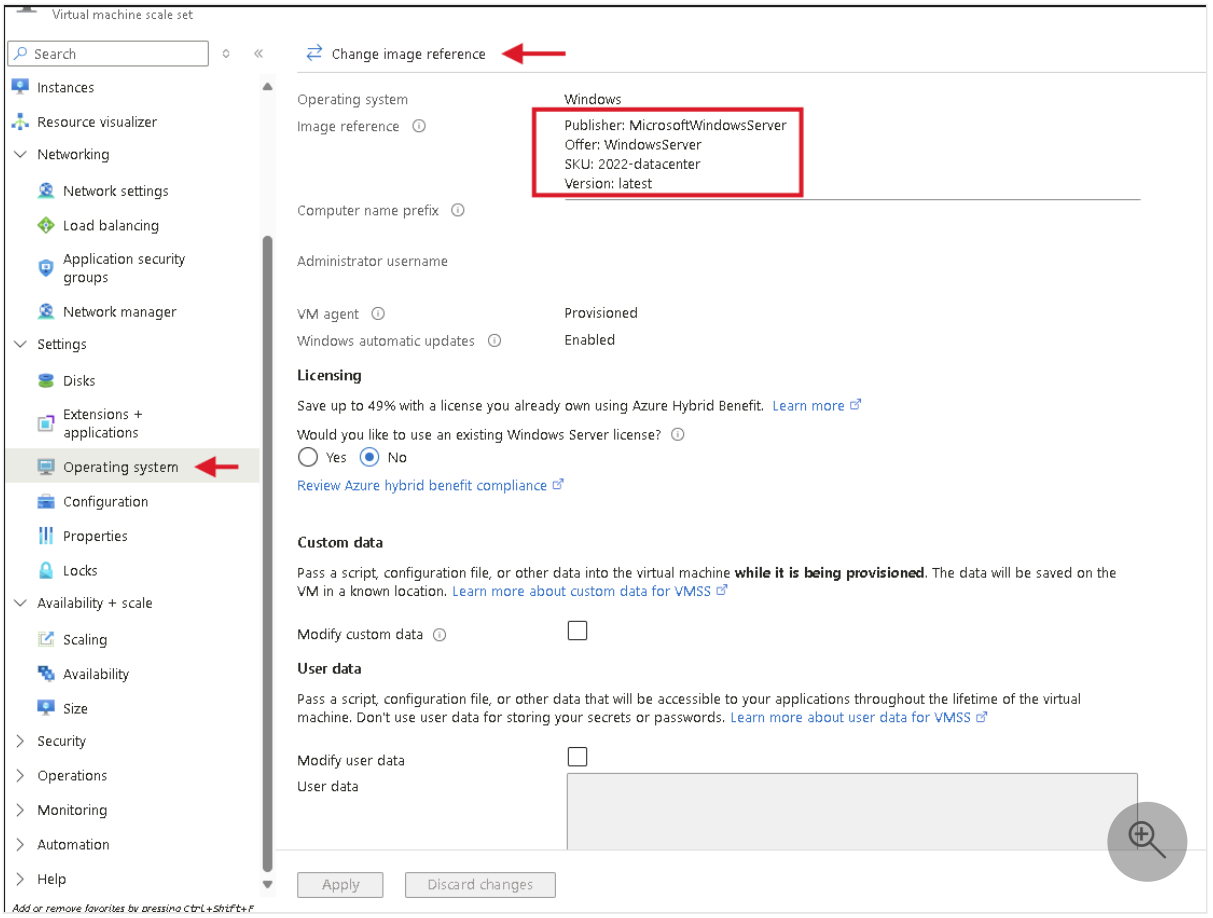
Portal

Following steps details how to enable Trusted launch on existing uniform scale set using Azure portal.

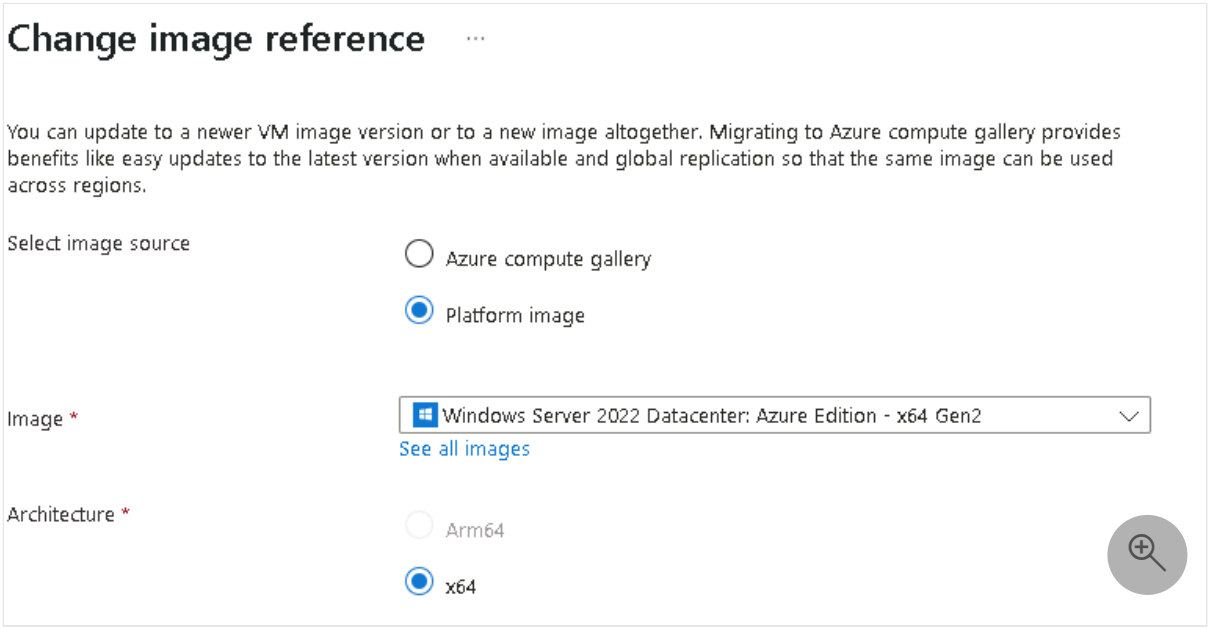
1. (Optional) **Scale set Size**: Navigate to `Size` under `Availability + scale` -> Modify the Scale set size if current size family isn't supported with Trusted launch security configuration -> Click **Apply**.



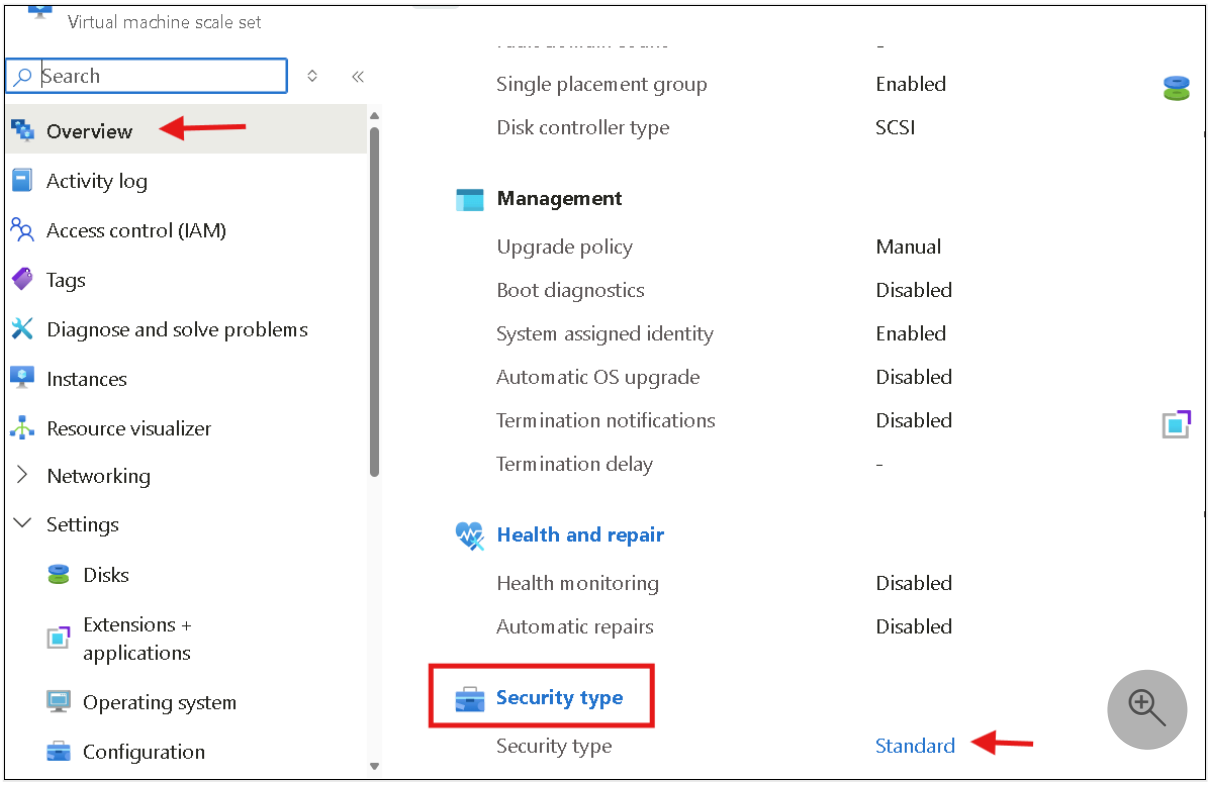
2. **OS Image**: Navigate to `Operating system` under `Settings` -> Click on `Change image reference`.



3. Update the OS Image reference to Gen2-Trusted launch supported OS image. Make sure the source Gen2 image has `TrustedLaunchSupported` security type if using Azure Compute Gallery OS image -> Click **Apply**.

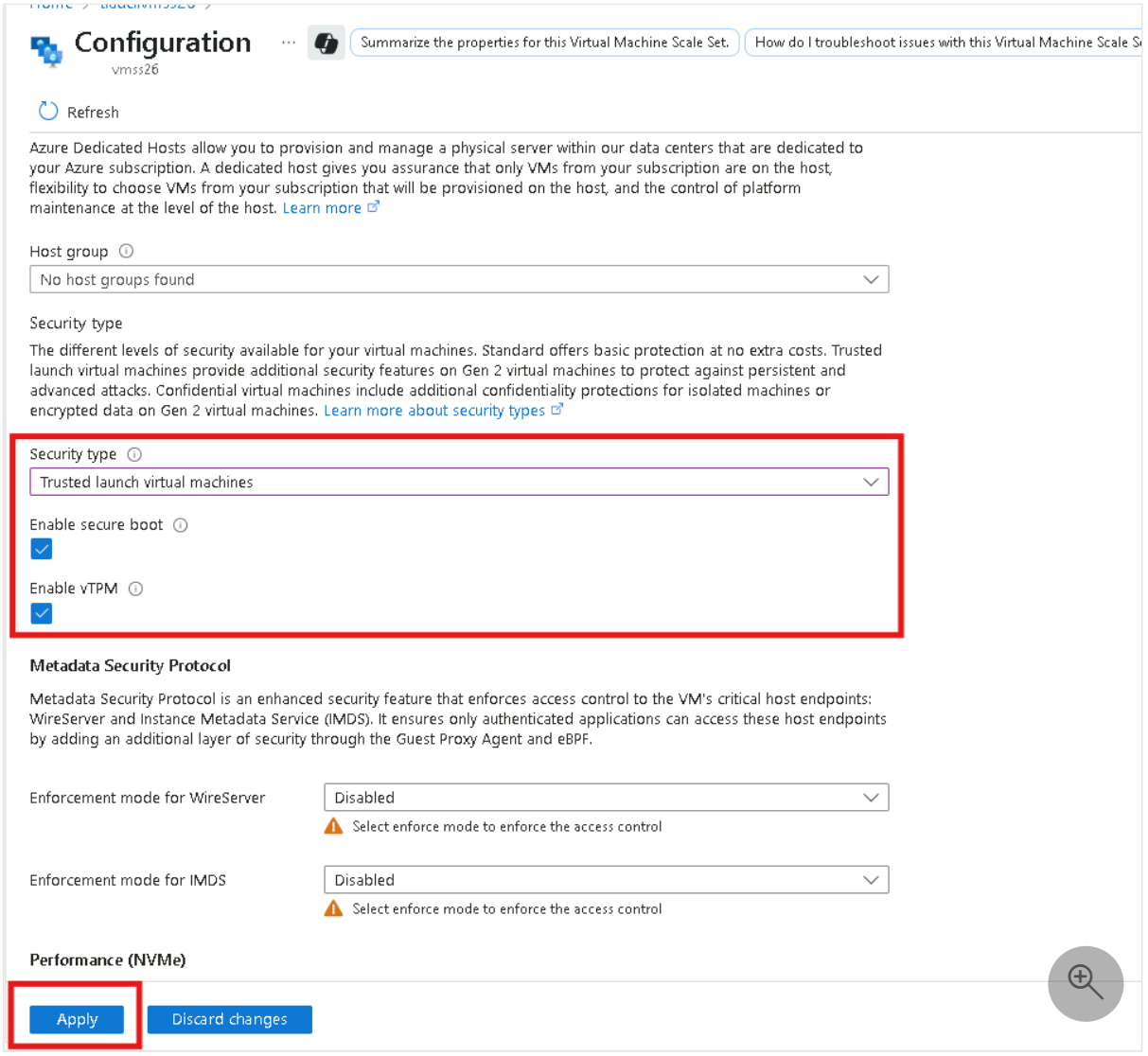


4. **Security type**: Click on **Standard** `Security type` on `Overview` page of scale set OR navigate to `Configuration` under `Settings`.

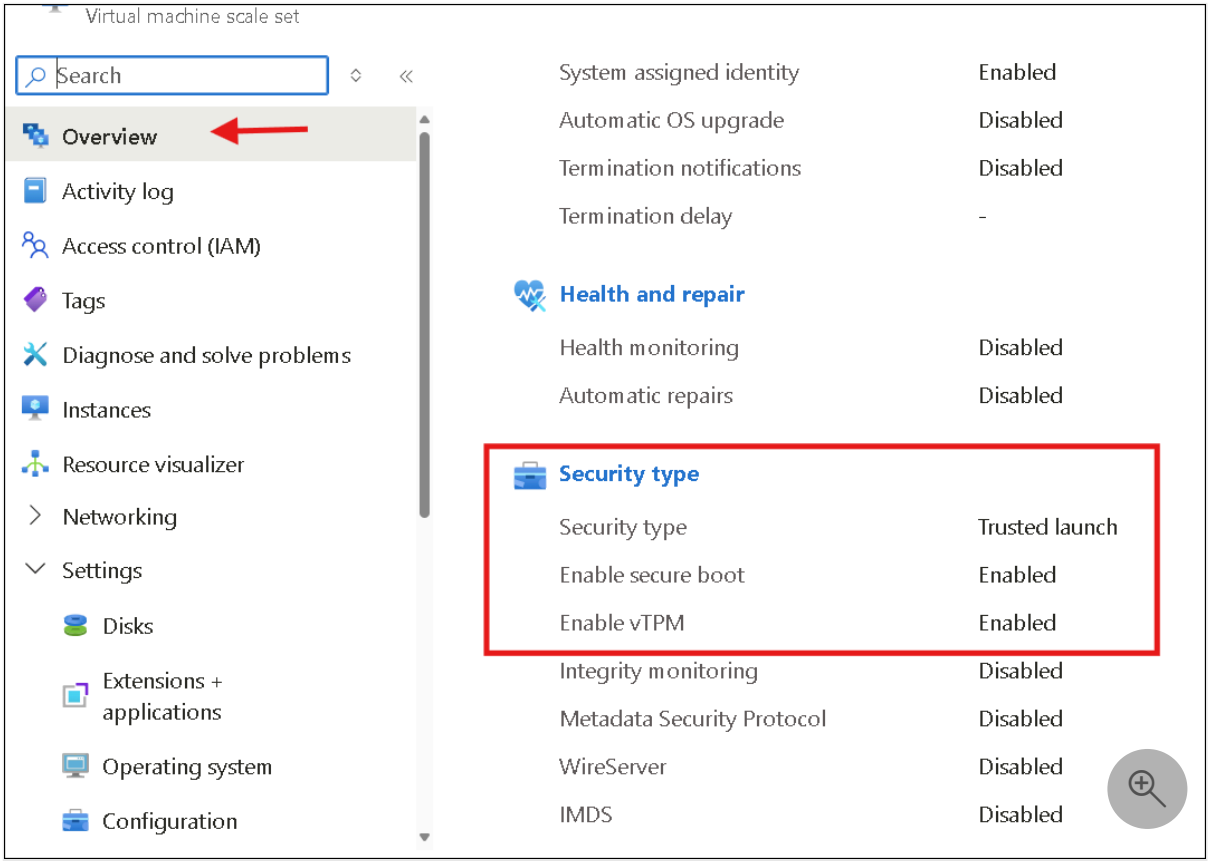


5. Update the security type drop-down on `Configuration` page from `Standard` to `Trusted launch` with `Enable secure boot` and `Enable vTPM` checked to enable Trusted Launch security configuration. Click `Yes` to confirm changes.

> ⓘ Note
>
> - **vTPM** is enabled by default.
> - **Secure Boot** should be enabled (not enabled by default) if you aren't using custom unsigned kernel or drivers. Secure Boot preserves boot integrity and enables foundational security for VM.

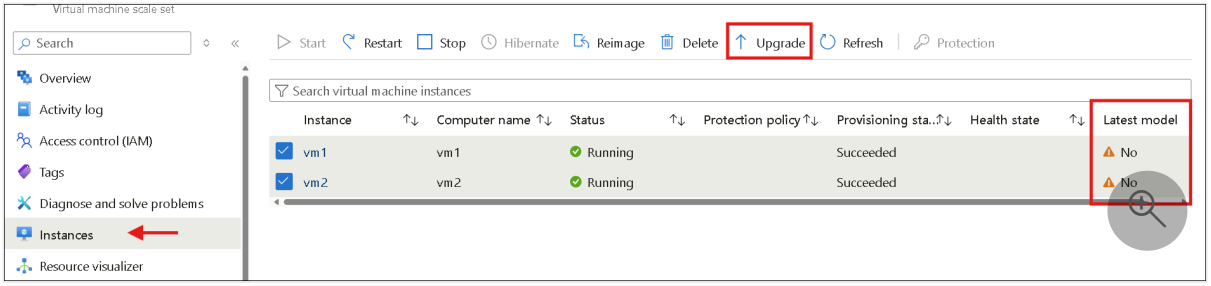


6. Validate the changes on the `Overview` page of scale set.



7. (Recommended) **Guest Attestation Extension**: Add Guest Attestation (GA) extension for Scale set resource, which enables Boot integrity monitoring for Scale set.

8. Update the VM instances manually if Scale set uniform upgrade mode is set to `Manual`.

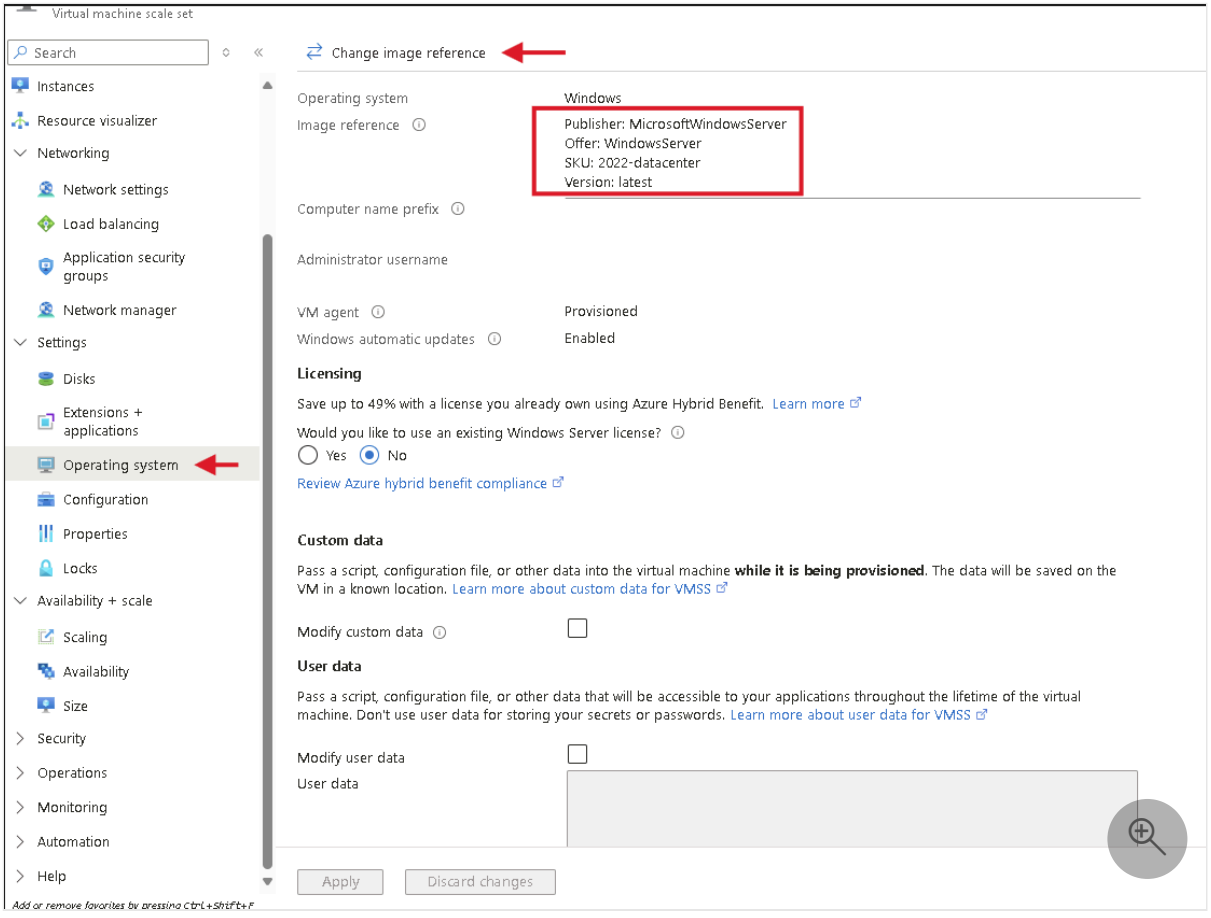


Roll-back

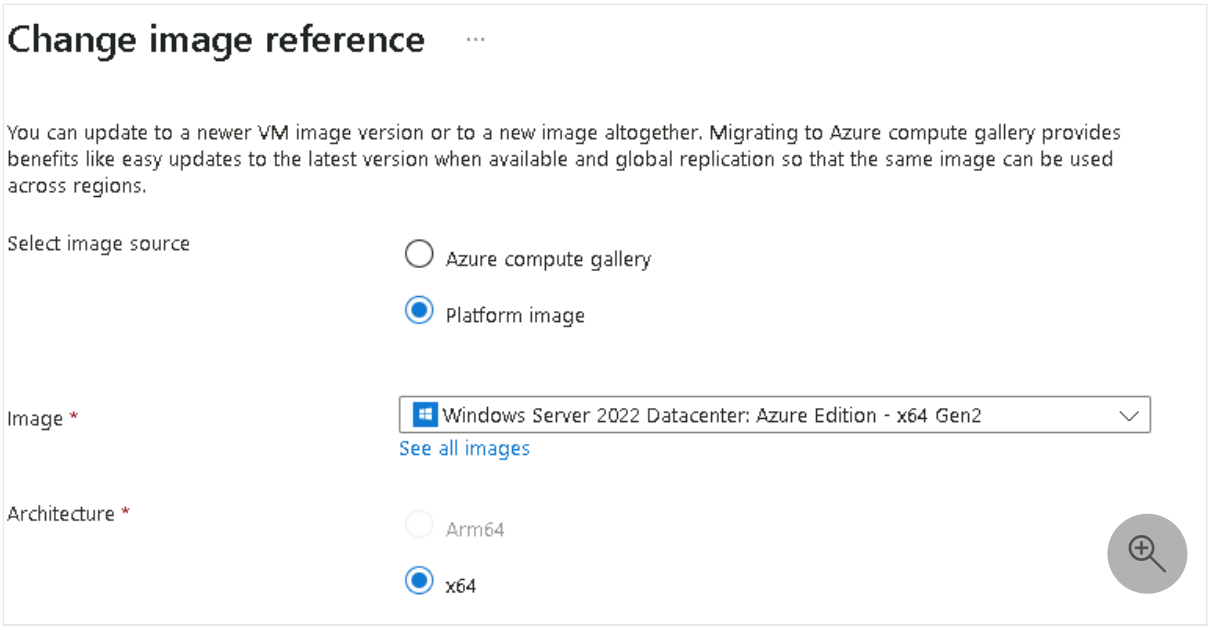
To roll-back changes from Trusted launch to previous known good configuration, you need to set `securityType` of Scale set to **Standard**.

Portal

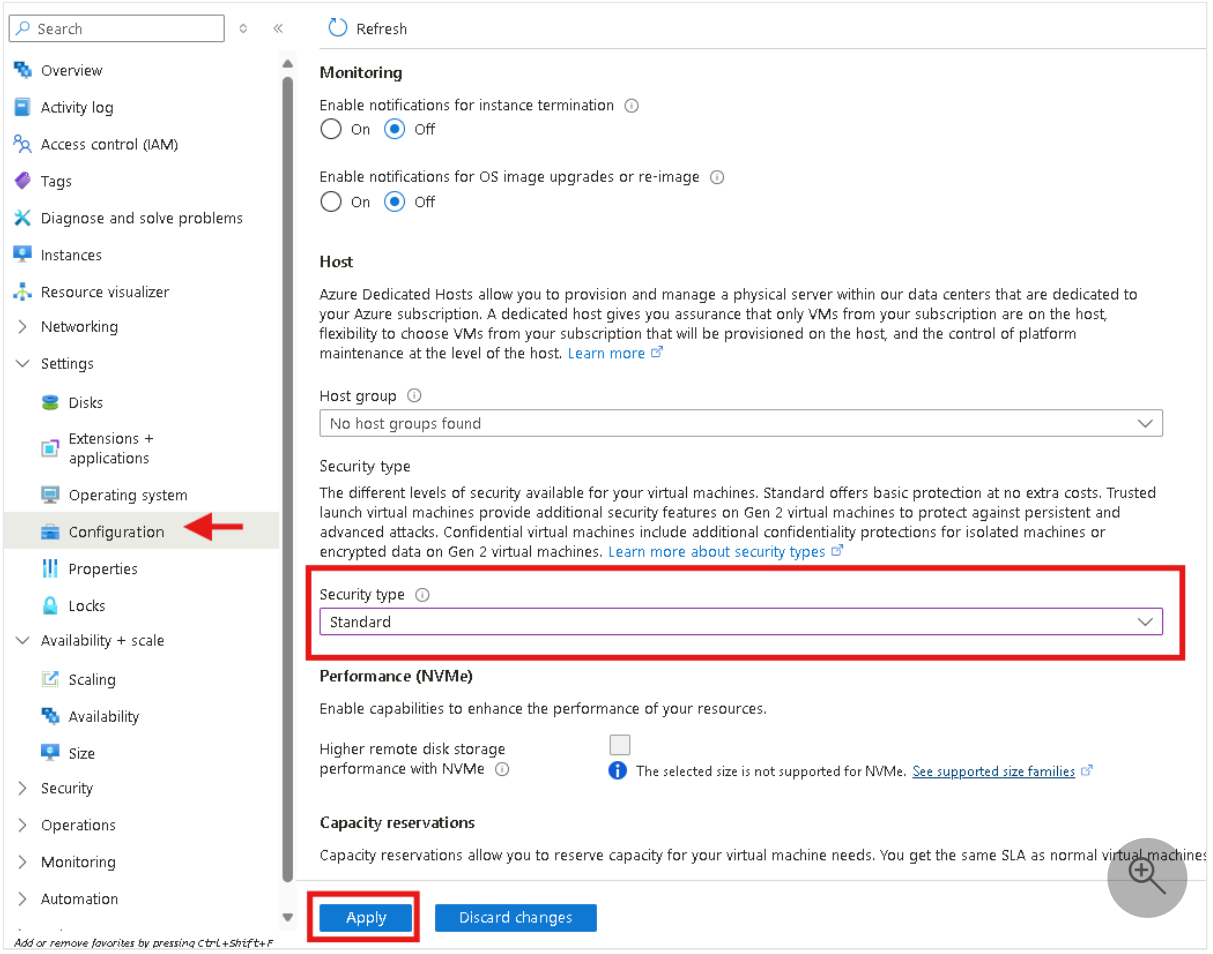
1. **OS Image**: Navigate to `Operating system` under `Settings`. Click on `Change image reference`.



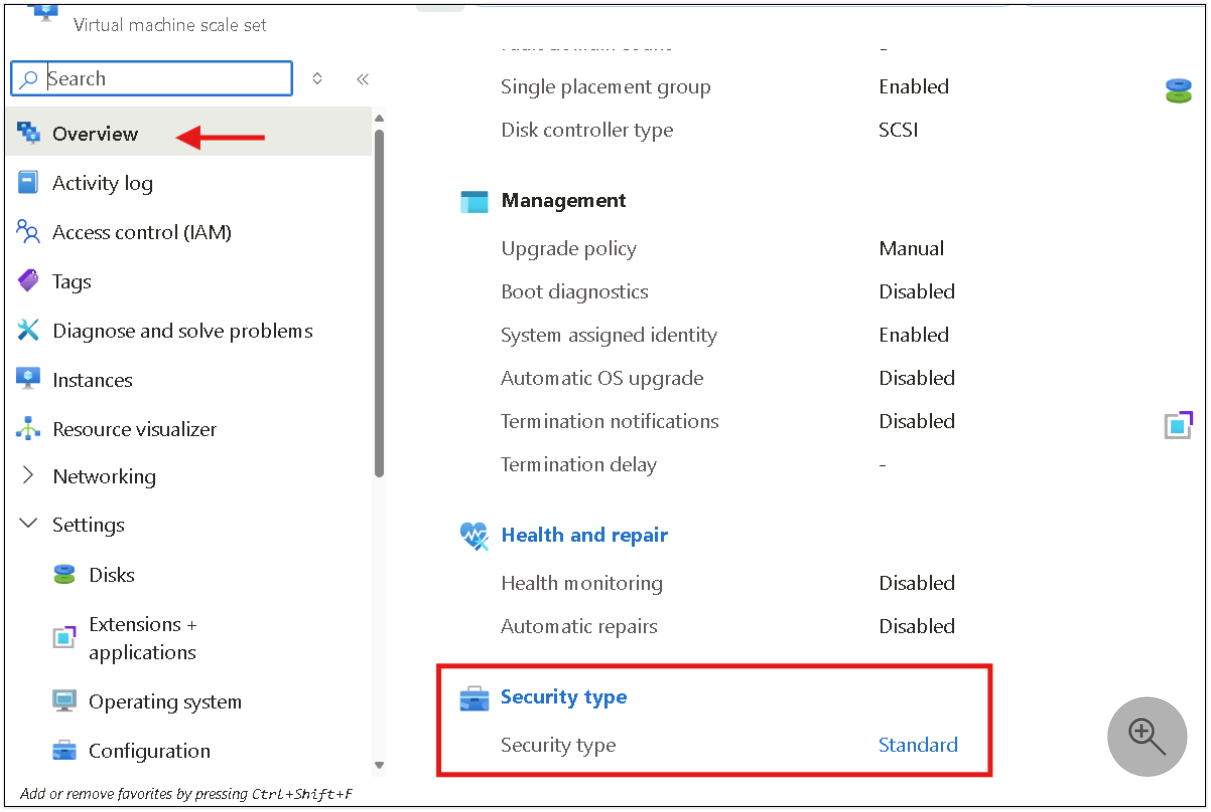
2. Update the OS Image reference to last known good configuration -> Click **Apply**.



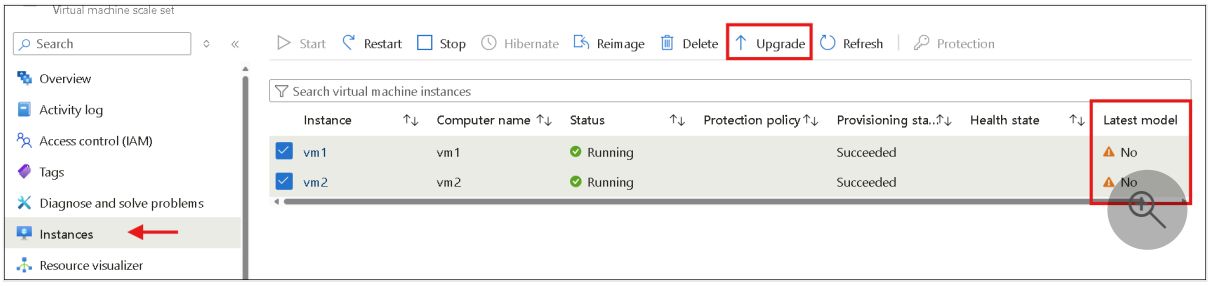
3. **Security type**: Navigate to `Configuration` page under `Settings` -> Update the security type drop-down on `Configuration` page from `Trusted launch` to `Standard` for disabling Trusted Launch security configuration. Click `Yes` to confirm changes.



4. Validate the changes on the `Overview` page of scale set.



5. Update the VM instances manually if Scale set uniform upgrade mode is set to `Manual`.



Next steps

(Recommended) Post-Upgrades enable Boot integrity monitoring to monitor the health of the VM using Microsoft Defender for Cloud.

Learn more about Trusted launch and review frequently asked questions.

Boot integrity monitoring overview

Article • 08/22/2024

To help Azure Trusted Launch better prevent malicious rootkit attacks on virtual machines (VMs), guest attestation through an Azure Attestation endpoint is used to monitor the boot sequence integrity. This attestation is critical to provide the validity of a platform's states.

Your Trusted Launch VM needs Secure Boot and virtual Trusted Platform Module (vTPM) to be enabled so that the attestation extensions can be installed. Microsoft Defender for Cloud offers reports based on Guest Attestation verifying status and that the boot integrity of your VM is set up correctly. To learn more about Microsoft Defender for Cloud integration, see Trusted Launch integration with Microsoft Defender for Cloud.

> **Important**
>
> Automatic Extension Upgrade is now available for the Boot Integrity Monitoring - Guest Attestation extension. For more information, see **Automatic Extension Upgrade**.

Prerequisites

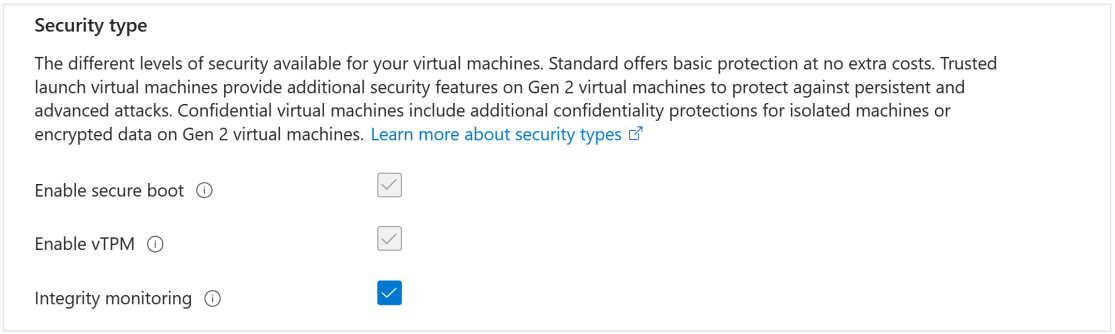
You need an active Azure subscription and a Trusted Launch VM.

Enable integrity monitoring

To enable integrity monitoring, follow the steps in this section.

Azure portal

1. Sign in to the Azure portal.
2. Select the resource (**Virtual Machines**).
3. Under **Settings**, select **Configuration**. On the **Security type** pane, select **Integrity monitoring**.



4. Save the changes.

On the VM **Overview** page, the security type for integrity monitoring should appear as **Enabled**.

This action installs the Guest Attestation extension, which you can refer to via the settings on the **Extensions + Applications** tab.

Troubleshooting guide for Guest Attestation extension installation

This section addresses attestation errors and solutions.

Symptoms

The Azure Attestation extension won't work properly when you set up a network security group (NSG) or a proxy. An error appears that looks similar to "`Microsoft.Azure.Security.WindowsAttestation.GuestAttestation` provisioning failed."



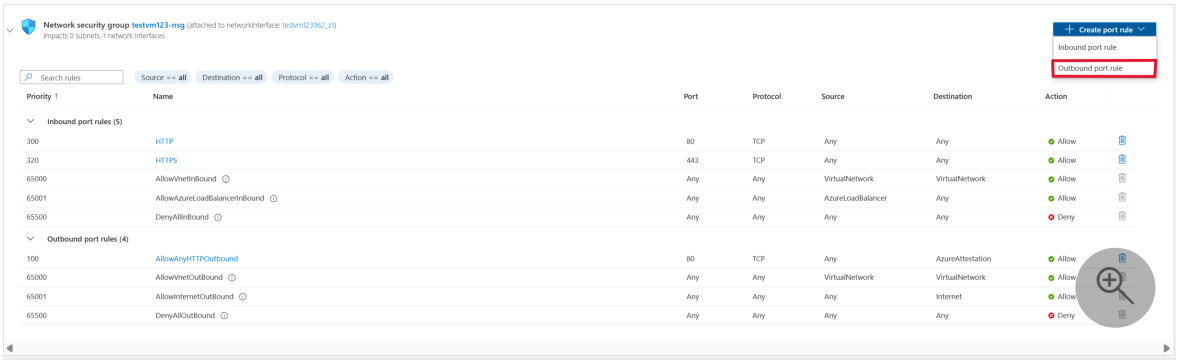
Solutions

In Azure, NSGs are used to help filter network traffic between Azure resources. NSGs contain security rules that either allow or deny inbound network traffic, or outbound network traffic from several types of Azure resources. The Azure Attestation endpoint should be able to communicate with the Guest Attestation extension. Without this

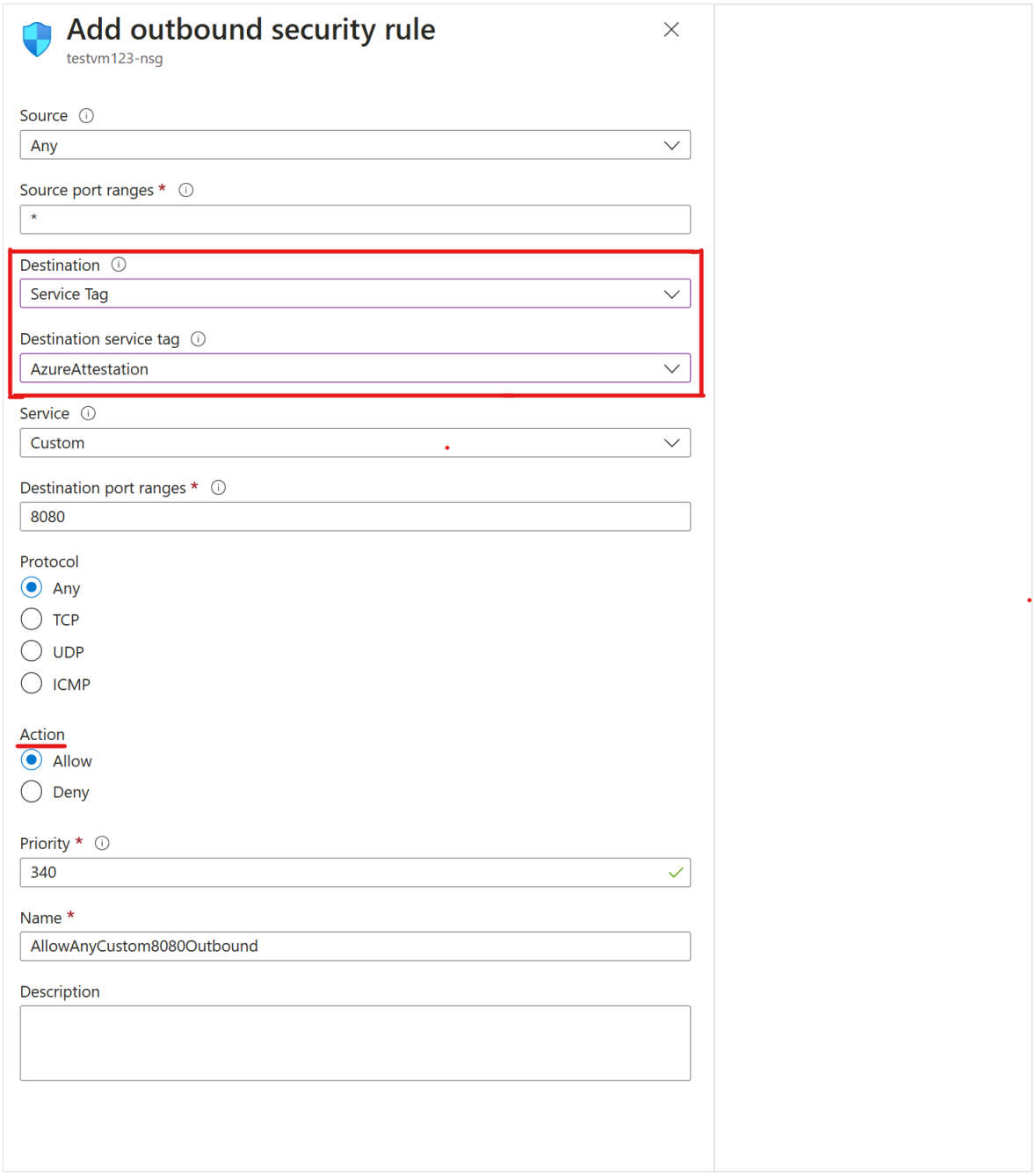
endpoint, Trusted Launch can't access guest attestation, which allows Microsoft Defender for Cloud to monitor the integrity of the boot sequence of your VMs.

To unblock Azure Attestation traffic in NSGs by using service tags:

1. Go to the VM that you want to allow outbound traffic.
2. On the leftmost pane, under **Networking**, select **Networking settings**.
3. Then select **Create port rule** > **Outbound port rule**.



4. To allow Azure Attestation, you make the destination a service tag. This setting allows for the range of IP addresses to update and automatically set rules that allow Azure Attestation. Set **Destination service tag** to **AzureAttestation** and set **Action** to **Allow**.

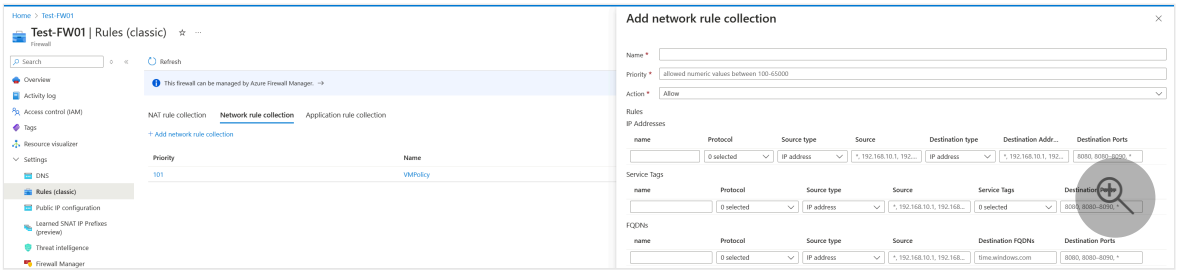


Firewalls protect a virtual network, which contains multiple Trusted Launch VMs. To unblock Azure Attestation traffic in a firewall by using an application rule collection:

1. Go to the Azure Firewall instance that has traffic blocked from the Trusted Launch VM resource.

2. Under **Settings**, select **Rules (classic)** to begin unblocking guest attestation behind the firewall.

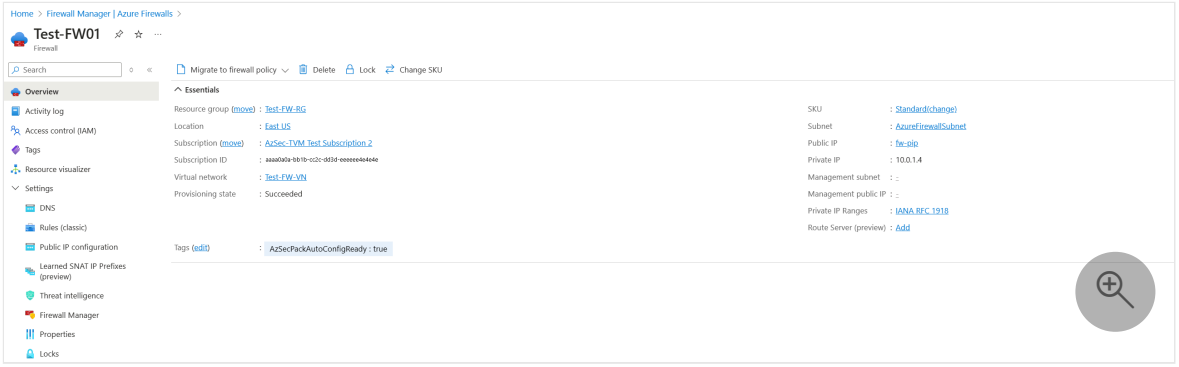
3. Under **Network rule collection**, select **Add network rule collection**.



4. Configure the name, priority, source type, and destination ports based on your needs. Set **Service tag name** to **AzureAttestation** and set **Action** to **Allow**.

To unblock Azure Attestation traffic in a firewall by using an application rule collection:

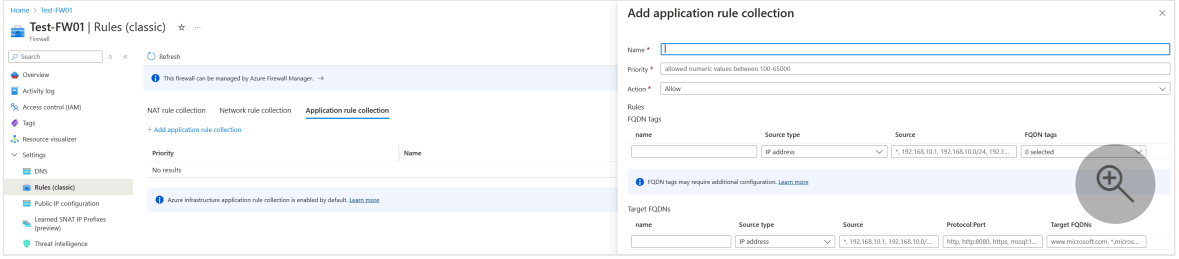
1. Go to the Azure Firewall instance that has traffic blocked from the Trusted Launch VM resource.



   The rules collection must contain at least one rule that targets fully qualified domain names (FQDNs).

2. Select the application rule collection and add an application rule.

3. Select a name and a numeric priority for your application rules. Set **Action** for the rule collection to **Allow**.



4. Configure the name, source, and protocol. The source type is for a single IP address. Select the IP group to allow multiple IP addresses through the firewall.

Regional shared providers

Azure Attestation provides a regional shared provider in each available region. You can choose to use the regional shared provider for attestation or create your own

providers with custom policies. Any Microsoft Entra user can access shared providers. The policy associated with it can't be changed.

> **Note**
>
> You can configure the source type, service, destination port ranges, protocol, priority, and name.

Related content

Learn more about Trusted Launch and deploying a Trusted Launch VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Trusted launch FAQs

Summarize this article for me

Frequently asked questions (FAQs) about Azure Trusted Launch feature use cases, support for other Azure features, and fixes for common errors.

Use cases

This section answers questions about use cases for Trusted Launch.

Trusted Launch is supported on both x64 and Arm64 Generation 2 Azure VMs, including Cobalt 100-based Arm64 sizes when using supported images.

Why should I use Trusted Launch? What does Trusted Launch guard against?

Trusted Launch guards against boot kits, rootkits, and kernel-level malware. These sophisticated types of malware run in kernel mode and remain hidden from users. For example:

- **Firmware rootkits**: These kits overwrite the firmware of the virtual machine (VM) BIOS, so the rootkit can start before the operating system (OS).
- **Boot kits**: These kits replace the OS's bootloader so that the VM loads the boot kit before the OS.
- **Kernel rootkits**: These kits replace a portion of the OS kernel, so the rootkit can start automatically when the OS loads.
- **Driver rootkits**: These kits pretend to be one of the trusted drivers that the OS uses to communicate with the VM's components.

What are the differences between Secure Boot and measured boot?

In a Secure Boot chain, each step in the boot process checks a cryptographic signature of the subsequent steps. For example, the BIOS checks a signature on the loader, and the loader checks signatures on all the kernel objects that it loads, and so on. If any of the objects are compromised, the signature doesn't match and the VM doesn't boot. For more information, see Secure Boot.

How does Trusted Launch compare to Hyper-V Shielded VM?

Hyper-V Shielded VM is currently available on Hyper-V only. Hyper-V Shielded VM is typically deployed with Guarded Fabric. A Guarded Fabric consists of a Host Guardian Service (HGS), one or more guarded hosts, and a set of Shielded VMs. Hyper-V Shielded VMs are used in fabrics where the data and state of the VM must be protected from various actors. These actors are both fabric administrators and untrusted software that might be running on the Hyper-V hosts.

Trusted launch can be deployed as a standalone VM or as virtual machine scale sets on Azure without other deployment and management of HGS. All of the Trusted Launch features can be enabled with a simple change in deployment or a checkbox on the Azure portal.

What is VM Guest State (VMGS)?

VM Guest State (VMGS) is specific to Trusted Launch VMs. It's a blob managed by Azure and contains the unified extensible firmware interface (UEFI) Secure Boot signature databases and other security information. The lifecycle of the VMGS blob is tied to that of the OS disk.

Supported features and deployments

This section discusses Trusted Launch supported features and deployments.

Are Arm64 VM sizes and images supported with Trusted Launch?

Yes, Trusted Launch is supported for Arm64 VMs. Arm64 images with Trusted Launch are available in Azure Marketplace. The Cobalt 100-based Arm64 VM series support Trusted Launch. You can deploy Trusted Launch on these Arm64 sizes using validated marketplace images.

Is Azure Compute Gallery supported by Trusted Launch?

Trusted Launch now allows images to be created and shared through the Azure Compute Gallery (formerly Shared Image Gallery). The image source can be:

- An existing Azure VM that is either generalized or specialized.
- An existing managed disk or a snapshot.
- A virtual hard disk (VHD) or an image version from another gallery.

For more information about deploying a Trusted Launch VM by using the Azure Compute Gallery, see Deploy Trusted Launch VMs.

Is Azure Backup supported by Trusted Launch?

Trusted Launch now supports Azure Backup. For more information, see Support matrix for Azure VM backup.

Will Azure Backup continue working after I enable Trusted Launch?

Backups configured with the Enhanced policy continue to take backups of VMs after you enable Trusted Launch.

Are ephemeral OS disks supported by Trusted Launch?

Trusted Launch supports ephemeral OS disks. For more information, see Trusted Launch for ephemeral OS disks.

> **Note**
>
> For Trusted launch VMs created with ephemeral disks, Keys and secrets generated or sealed by the virtual Trusted Platform Module (vTPM) after the creation of the VM might not be persisted across operations like reimaging and platform events like service healing.

Are security features available with Trusted launch applicable to data disks as well?

Trusted launch provides foundational security for Operating system hosted in virtual machine by attesting its boot integrity. Trusted launch security features are applicable for running OS and OS disks only, they aren't applicable to data disks or OS binaries stored in data disks. For more information, see Trusted launch overview

Can a VM be restored by using backups taken before Trusted Launch was enabled?

Backups taken before you upgrade an existing Generation 2 VM to Trusted Launch can be used to restore the entire VM or individual data disks. They can't be used to restore or replace the OS disk only.

How can I find VM sizes that support Trusted Launch?

See the list of Generation 2 VM sizes that support Trusted Launch, which includes both x64 and Arm64 sizes.

Arm64 Cobalt 100-based VM sizes (Dpsv6, Dplsv6, Epsv6) support Trusted Launch. These Arm64 sizes are Gen2 and support Trusted Launch.

Use the following commands to check if a Generation 2 VM size doesn't support Trusted Launch.

CLI

Azure CLI

```
subscription="<yourSubID>"
region="westus"
vmSize="Standard_NC12s_v3"

az vm list-skus --resource-type virtualMachines  --location $region --query "[?
name=='$vmSize'].capabilities" --subscription $subscription
```

How can I validate that my OS image supports Trusted Launch?

See the list of OS versions supported with Trusted Launch.

Both x64 and Arm64 Gen2 images can support Trusted Launch. For Arm64 images, the CLI/PowerShell outputs show `architecture` as `Arm64` and the `features` include `SecurityType` with `TrustedLaunch` or `TrustedLaunchSupported`.

Marketplace OS images

Use the following commands to check if an Azure Marketplace OS image supports Trusted Launch.

CLI

Azure CLI

```
az vm image show --urn "MicrosoftWindowsServer:WindowsServer:2022-datacenter-
azure-edition:latest"
```

The response is similar to the following form. If `hyperVGeneration` is `V2` and `SecurityType` contains `TrustedLaunch` in the output, the Generation 2 OS image supports Trusted Launch.

JSON


```
{
  "architecture": "x64",
  "automaticOsUpgradeProperties": {
    "automaticOsUpgradeSupported": false
  },
  "dataDiskImages": [],
  "disallowed": {
    "vmDiskType": "Unmanaged"
  },
  "extendedLocation": null,
  "features": [
    {
      "name": "SecurityType",
      "value": "TrustedLaunchAndConfidentialVmSupported"
    },
    {
      "name": "IsAcceleratedNetworkSupported",
      "value": "True"
    },
    {
      "name": "DiskControllerTypes",
      "value": "SCSI, NVMe"
    },
    {
      "name": "IsHibernateSupported",
      "value": "True"
    }
  ],
  "hyperVGeneration": "V2",
  "id": "/Subscriptions/00000000-0000-0000-0000-
00000000000/Providers/Microsoft.Compute/Locations/westus/Publishers/MicrosoftWi
ndowsServer/ArtifactTypes/VMImage/Offers/WindowsServer/Skus/2022-datacenter-
azure-edition/Versions/20348.1906.230803",
  "imageDeprecationStatus": {
    "alternativeOption": null,
    "imageState": "Active",
    "scheduledDeprecationTime": null
  },
  "location": "westus",
  "name": "20348.1906.230803",
  "osDiskImage": {
    "operatingSystem": "Windows",
    "sizeInGb": 127
  },
  "plan": null,
```

```
  "tags": null
}
```

For Arm64 Marketplace images (for example, `Canonical:0001-com-ubuntu-server-jammy:22_04-lts-arm64:latest`), the output shows `"architecture": "Arm64"` with `SecurityType` including `TrustedLaunchSupported`.

Azure Compute Gallery OS image

Use the following commands to check if an Azure Compute Gallery OS image supports Trusted Launch.

CLI

Azure CLI

```
az sig image-definition show `
    --gallery-image-definition myImageDefinition `
    --gallery-name myImageGallery `
    --resource-group myImageGalleryRg
```

The response is similar to the following form. If `hyperVGeneration` is `V2` and `SecurityType` contains `TrustedLaunch` in the output, the Generation 2 OS image supports Trusted Launch.

JSON

```
{
  "architecture": "x64",
  "features": [
    {
      "name": "SecurityType",
      "value": "TrustedLaunchSupported"
    }
  ],
  "hyperVGeneration": "V2",
  "id": "/subscriptions/00000000-0000-0000-0000-
000000000000/resourceGroups/myImageGalleryRg/providers/Microsoft.Compute/galler
ies/myImageGallery/images/myImageDefinition",
  "identifier": {
    "offer": "myImageDefinition",
    "publisher": "myImageDefinition",
    "sku": "myImageDefinition"
  },
  "location": "westus3",
```

```
  "name": "myImageDefinition",
  "osState": "Generalized",
  "osType": "Windows",
  "provisioningState": "Succeeded",
  "recommended": {
    "memory": {
      "max": 32,
      "min": 1
    },
    "vCPUs": {
      "max": 16,
      "min": 1
    }
  },
  "resourceGroup": "myImageGalleryRg",
  "tags": {},
  "type": "Microsoft.Compute/galleries/images"
}
```


How can I validate Secure boot compatibility for Gen1 or Gen2 VM before upgrade to Trusted launch?

To self-evaluate the presence of untrusted boot components or kernel modules in your existing Gen1/Gen2 Linux VMs before migrating to Trusted Launch, you can use the `SBInfo` tool from the Linux Security Package

1. Connect to your Azure Linux Gen1/Gen2 VM.
2. Install the SBInfo tool for the distribution your VM is running. It resides within the Linux Security Package.

Refer to Linux Trusted launch Secure boot validation for distribution-based `SBInfo` tool installation commands.

How do external communication drivers work with Trusted Launch VMs?

Adding component object model (COM) ports require that you disable Secure Boot. COM ports are disabled by default in Trusted Launch VMs.

Trusted launch as default (TLaD)

What is Trusted Launch as default?

Trusted launch as default (TLaD) is currently in preview for all clients AND generally available for Azure portal, PowerShell, and CLI. For more information, see Trusted launch default (Preview).

Does TLaD impacts existing VM & scale sets?

Trusted launch as default doesn't change existing Azure VMs, scale sets already running in your environment

Do I need to update my automation scripts or deployment templates?

You need to update the API versions for following resource providers to validate the Trusted launch default end to end experience as part of the preview:

- `Microsoft.Compute/virtualMachines` – API version `2021-11-01` or higher.
- `Microsoft.Compute/virtualMachineScaleSets` – API version `2021-11-01` or higher.

I'm currently using Gen2 VM or scale set without Trusted Launch and would like to continue using Non-Trusted Launch configuration after the TLaD general availability

Trusted Launch VMs provide you with foundational compute security at no extra cost. We strongly recommend that you don't disable them for new VM or scale set deployments. Reach out to us at Trusted launch default feedback.

See Can I disable Trusted launch for new deployment if you explicitly need to disable Trusted launch.

What could be possible situations where I must bypass Trusted launch defaults for VM or scale set?

You need to explicitly bypass Trusted launch default if one of the following scenarios applies to your Gen2 VM or scale set deployments:

- Gen2 VM is used to generate `TrustedLaunchSupported` or `TrustedLaunchAndConfidentialVMSupported` or `ConfidentialVMSupported` Azure compute gallery images via Azure image builder (AIB) or Packer. OR,
- Gen2 VM is used to create managed images*. OR,
- Gen2 Linux VM requires Hibernation enabled.

> **Note**
>
> For managed images, use Azure Compute Gallery for the latest capabilities. All new features, like ARM64, Trusted Launch, and Confidential VM are only supported through Azure Compute Gallery.

Can I disable Trusted Launch for a new VM deployment?

Trusted Launch VMs provide you with foundational compute security. We **strongly** recommend that you don't disable them for new VM or scale set deployments except if your deployments have dependency on:

- A VM size currently not supported
- Unsupported features with Trusted Launch
- An OS that doesn't support Trusted Launch
- VM used to generate TrustedLaunchSupported Azure compute gallery image

You can use the `securityType` parameter with the `Standard` value to disable Trusted Launch in new VM or scale set deployments by using Azure PowerShell (v10.3.0+) and the Azure CLI (v2.53.0+).

> **Note**
>
> - Parameter `securityType` with value `Standard` can be used if subscription has feature flag `UseStandardSecurityType` registered under `Microsoft.Compute` namespace. Refer to Setup feature in Azure subscription for steps to enable required feature.
> - We don't recommend disabling Secure Boot unless you're using custom unsigned kernel or drivers.

If you need to disable Secure Boot, under the VM's configuration, clear the **Enable Secure Boot** option.

Template

Sample `securityProfile` element of ARM template for using `securityType` parameter with value `Standard`

JSON

```
"securityProfile": {
    "securityType": "Standard",
    "uefiSettings": "[null()]"
}
```

Troubleshooting issues

This section answers questions about specific states, boot types, and common boot issues.

What should I do when my Trusted Launch VM has deployment failures?

This section provides more details on Trusted Launch deployment failures for you to take proper action to prevent them.

JSON

```
Virtual machine <vm name> failed to create from the selected snapshot because the
virtual Trusted Platform Module (vTPM) state is locked.
To proceed with the VM creation, please select a different snapshot without a
locked vTPM state.
For more assistance, please refer to “Troubleshooting locked vTPM state” in FAQ
page at https://aka.ms/TrustedLaunch-FAQ.
```

This deployment error happens when the snapshot or restore point provided is inaccessible or unusable for the following reasons:

1. Corrupt virtual machine guest state (VMGS).
2. vTPM in a locked state.
3. One or more critical vTPM indices are in an invalid state.

The listed reasons can happen if a user or workload running on the virtual machine sets the lock on vTPM or modifies critical vTPM indices that leave the vTPM in an invalid state.

Retrying with the same snapshot or restore point results in the same failure.

Resolution:

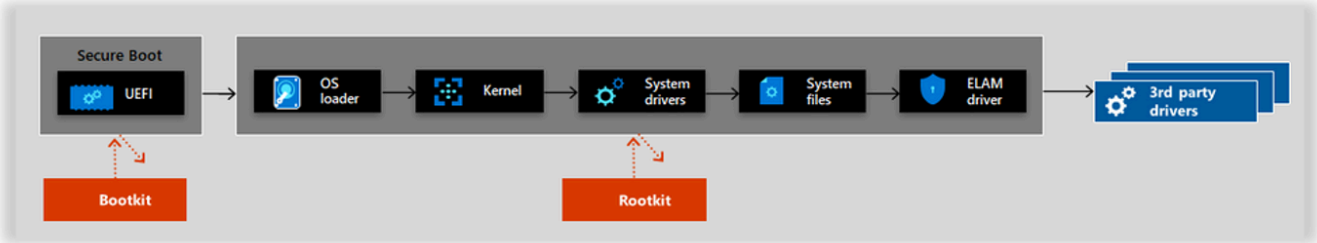
1. On the source Trusted Launch VM where the snapshot or restore point was generated, the vTPM errors must be rectified.
   a. For vTPM state modified by a workload on the virtual machine, you need to use the same to check the error states and bring the vTPM to a non-error state.

b. For vTPM state modified using trusted platform module (TPM) tools, then you should use the same tools to check the error states and bring the vTPM to a non-error state.

Once the snapshot or restore point is free from these errors, you can use same to create a new Trusted Launch VM.

Why is the Trusted Launch VM not booting correctly?

If unsigned components are detected from the UEFI (guest firmware), bootloader, OS, or boot drivers, a Trusted Launch VM doesn't boots. The Secure Boot setting in the Trusted Launch VM fails to boot if unsigned or untrusted boot components are encountered during the boot process and reports as a Secure Boot failure.

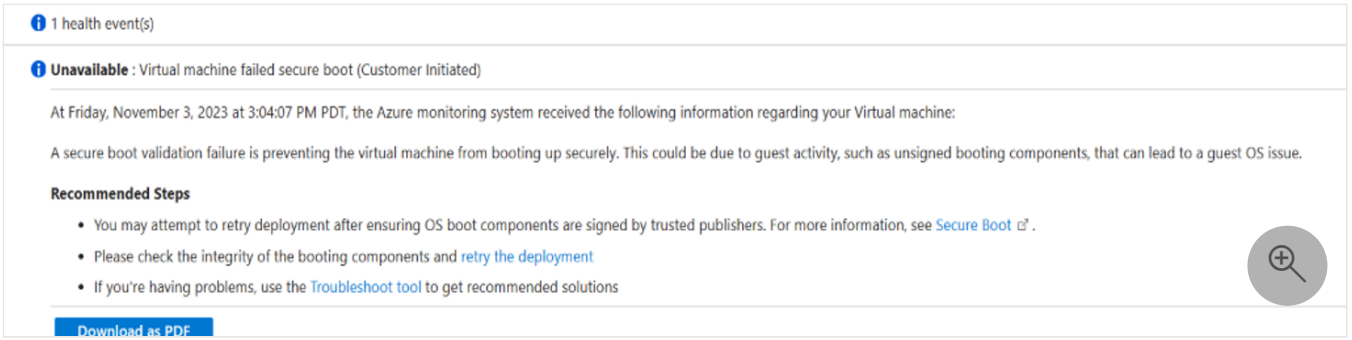


> **Note**
>
> Trusted Launch VMs that are created directly from an Azure Marketplace image shouldn't encounter Secure Boot failures. Azure Compute Gallery images with an original image source of Azure Marketplace and snapshots created from Trusted Launch VMs should also not encounter these errors.

How would I verify a no-boot scenario in the Azure portal?

When a VM becomes unavailable from a Secure Boot failure, `no-boot` means that VM has an OS component not signed by a trusted authority, which blocks booting a Trusted Launch VM. On VM deployment, you might see information from resource health within the Azure portal stating that there's a validation error in Secure Boot.

To access resource health from the VM configuration page, go to **Resource Health** under the **Help** pane.



If you verified that the no-boot is caused due to Secure Boot failure:

1. The image you're using is an older version that might have one or more untrusted boot components and is on a deprecation path. To remedy an outdated image, update to a supported newer image version.
2. The image you're using might be built outside of a marketplace source or the boot components modified and contain unsigned or untrusted boot components. To verify whether your image has unsigned or untrusted boot components, see the following section, "Verify Secure Boot failures."
3. If the preceding two scenarios don't apply, the VM is potentially infected with malware (bootkit/rootkit). Consider deleting the VM and re-creating a new VM from the same source image while you evaluate all the software being installed.

Why does my Trusted Launch VM show 50 MB less memory?

With Trusted Launch, an execution environment commonly known as "the paravisor" is created and runs inside the VM. Typically, about 50MB of memory is used by the paravisor and would show as "reserved" within the guest Operating System.

Verify Secure Boot failures

This section helps you verify Secure Boot failures.

Linux Trusted launch virtual machines

To verify which boot components are responsible for Secure Boot failures within an Azure Linux Trusted Launch VM, you can use the `SBInfo` tool from the Linux Security Package

1. Turn off Secure Boot.
2. Connect to your Azure Linux Trusted Launch VM.
3. Install the SBInfo tool for the distribution your VM is running. It resides within the Linux Security Package

Debian-based distributions

These commands apply to Ubuntu, Debian, and other Debian-based distributions.

Bash

```
echo "deb [arch=amd64] http://packages.microsoft.com/repos/azurecore/ trusty
main" | sudo tee -a /etc/apt/sources.list.d/azure.list

echo "deb [arch=amd64] http://packages.microsoft.com/repos/azurecore/ xenial
main" | sudo tee -a /etc/apt/sources.list.d/azure.list

echo "deb [arch=amd64] http://packages.microsoft.com/repos/azurecore/ bionic
main" | sudo tee -a /etc/apt/sources.list.d/azure.list

wget https://packages.microsoft.com/keys/microsoft.asc

wget https://packages.microsoft.com/keys/msopentech.asc

sudo apt-key add microsoft.asc && sudo apt-key add msopentech.asc

sudo apt update && sudo apt install azure-security
```

After you install the Linux Security Package for your distribution, run the `sbinfo` command to verify which boot components are responsible for Secure Boot failures by displaying all unsigned modules, kernels, and bootloaders.

Bash

```
sudo sbinfo -u -m -k -b
```

To learn more about the SBInfo diagnostic tool, you can run `sudo sbinfo -help`.

Why am I getting a boot integrity monitoring fault?

Trusted Launch for Azure VMs is monitored for advanced threats. If such threats are detected, an alert is triggered. Alerts are only available if enhanced security features in Microsoft Defender for Cloud are enabled.

Microsoft Defender for Cloud periodically performs attestation. If the attestation fails, a medium-severity alert is triggered. Trusted Launch attestation can fail for the following reasons:

- The attested information, which includes a log of the Trusted Computing Base (TCB), deviates from a trusted baseline (like when Secure Boot is enabled). Any deviation

indicates that untrusted modules were loaded and the OS might be compromised.
- The attestation quote couldn't be verified to originate from the vTPM of the attested VM. The verification failure indicates malware is present and might be intercepting traffic to the TPM.
- The attestation extension on the VM isn't responding. An unresponsive extension indicates a denial-of-service attack by malware or an OS admin.

Certificates

This section provides information on certificates.

How can I establish root of trust with Trusted Launch VMs?

The virtual TPM AK public certificate provides you with visibility for information on the full certificate chain (Root and Intermediate Certificates) to help you validate trust in the certificate and root chain. It provides information on instance properties so that you can trace back to the full chain. It ensures that you continually have the highest security posture for Trusted Launch

Download instructions

Package certificates, composed of .p7b (Full Certificate Authority) and .cer (Intermediate CA), reveal the signing and certificate authority. Copy the relevant content and use certificate tooling to inspect and assess details of certificates.

▼ Select to download the .crt that follows for the Azure Virtual TPM Root Certificate Authority 2023

Azure Virtual TPM Root Certificate Authority 2023

▼ Select to view the .cert content for the Root Certificate

Azure Virtual TPM Root Certificate Authority 2023

```
-----BEGIN CERTIFICATE-----
MIIFsDCCA5igAwIBAgIQUfQx2iySCIpOKeDZKd5KpzANBgkqhkiG9w0BAQwFADBp
MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9uMTow
OAYDVQQDEzFBenVyZSBWaXJ0dWFsIFRQTSBSb290IENlcnRpZmljYXRlIEF1dGhv
cml0eSAyMDIzMB4XDTIzMDYwMTE4MDg1M1oXDTQ4MDYwMTE4MTU0MVowaTELMAkG
A1UEBhMCVVMxHjAcBgNVBAoTFU1pY3Jvc29mdCBDb3Jwb3JhdGlvbjE6MDgGA1UE
AxMxQXp1cmUgVmlydHVhbCBUUE0gUm9vdCBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkg
MjAyMzCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALoMMwvdRJ7+bW00
adKE1VemNqJS+268Ure8QcfZXVOsVO22+PL9WRoPnWo0r5dVoomYGbobh4HC72s9
sGY6BGRe+Ui2LMwuWnirBtOjaJ34r1ZieNMcVNJT/dXW5HN/HLlm/gSKlWzqCEx6
gFFAQTvyYl/5jYI4Oe05zJ7ojgjK/6ZHXpFysXnyUITJ9qgjn546IJh/G5OMC3mD
```

```
fFU7A/GAi+LYaOHSzXj69Lk1vCftNq9DcQHtB7otO0VxFkRLaULcfu/AYHM7FC/S
q6cJb9Au8K/IUhw/5lJSXZawLJwHpcEYzETm2blad0VHsACaLNucZL5wBi8GEusQ
9Wo8W1p1rUCMp89pufxa3Ar9sYZvWeJlvKggWcQVUlhvvIZEnT+fteEvwTdoajl5
qSvZbDPGCPjb91rSznoiLq8XqgQBBFjnEiTL+ViaZmyZPYUsBvBY3lKXB1l2hgga
hfBIag4j0wcgqlL82SL7pAdGjq0Fou6SKgHnkkrV5CNxUBBVMNCwUoj5mvEjd5mF
7XPgfM98qNABb2Aqtfl+VuCkU/G1XvFoTqS9AkwbLTGFMS9+jCEU2rw6wnKuGv1T
x9iuSdNvsXt8stx4fkVeJvnFpJeAIwBZVgKRSTa3w3099k0mW8qGiMnwCI5SfdZ2
SJyD4uEmszsnieE6wAWd1tLLg1jvAgMBAAGjVDBSMA4GA1UdDwEB/wQEAwIBhjAP
BgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBRL/iZalMH2M8ODSCbd8+WwZLKqlTAQ
BgkrBgEEAYI3FQEEAwIBADANBgkqhkiG9w0BAQwFAAOCAgEALgNAyg8I0ANNO/8I
2BhpTOsbywN2YSmShAmig5h4sCtaJSM1dRXwA+keY6PCXQEt/PRAQAiHNcOF5zbu
OU1Bw/Z5Z7k9okt04eu8CsS2Bpc+POg9js6lBtmigM5LWJCH1goMD0kJYpzkaCzx
1TdD3yjo0xSxgGhabk5Iu1soD3OxhUyIFcxaluhwkiVINt3Jhy7G7VJTlEwkk21A
oOrQxUsJH0f2GXjYShS1r9qLPzLf7ykcOm62jHGmLZVZujBzLIdNk1bljP9VuGW+
cISBwzkNeEMMFufcL2xh6s/oiUnXicFWvG7E6ioPnayYXrHy3Rh68XLnhfpzeCzv
bz/I4yMV38qGo/cAY2OJpXUuuD/ZbI5rT+lRBEkDW1kxHP8cpwkRwGopV8+gX2KS
UucIIN4l8/rrNDEX8T0b5U+BUqiO7Z5YnxCya/H0ZIwmQnTlLRTU2fW+OGG+xyIr
jMi/0l6/yWPUkIAkNtvS/yO7USRVLPbtGVk3Qre6HcqacCXzEjINcJhGEVg83Y8n
M+Y+a9J0lUnHytMSFZE85h88OseRS2QwqjozUo2j1DowmhSSUv9Na5Ae22ycciBk
EZSq8a4rSlwqthaELNpeoTLUk6iVoUkK/iLvaMvrkdj9yJY1O/gvlfN2aiNTST/2
bd+PA4RBToG9rXn6vNkUWdbLibU=
-----END CERTIFICATE-----
```

▼ Select to view the .p7b content (Root + ICA-01)

Global Virtual TPM CA - 01:

```
-----BEGIN CERTIFICATE-----
MIIRKQYJKoZIhvcNAQcCoIIRGjCCERYCAQExADCCBbMGCSqGSIb3DQEHAaCCBaQE
ggWgMIIFnDCCA4SgAwIBAgITMwAAAALA0XtLj5ecNQAAAAAAAjANBgkqhkiG9w0B
AQwFADBpMQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0
aW9uMTowOAYDVQQDEzFBenVyZSBWaXJ0dWFsIFRQTSBSb290IENlcnRpZmljYXRl
IEF1dGhvcml0eSAyMDIzMB4XDTIzMDYwODE3NTMwNFoXDTI1MTEwMzE3NTMwNFow
JTEjMCEGA1UEAxMaR2xvYmFsIFZpcnR1YWwgVFBNIENBIC0gMDEwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQC/NqouHGBovTadw1GLMQYNxWrEciPSh7gE
7VxsfbbPCE0SfTO80OF2raLdRYN2gEWE2Dr8+TDUuAb/WBFyczhU1aHFPssg8B3/
DT6pTXlDlohLLPWkjYU+OuT1/Ls7RzjQBe2se/MJyPaIJXI6KgCwePw7EcWFChe8
aCTUMHYBG0Ju4xNlMTUd/tcu6a53CXn6Nq48umwlaJelRh+i1f0vcwB2CY/v+Rli
wb/8DM5Ed9FUHZOKyp5Vnaw9GWloM46sLQT/fdHB0jmugfNZzafkkhQAYiNL3jYN
YFZH5/IgUfYJ/yybwnwoxOdV2NV0Q2i+P5Pcb0WNGaJY47aqOj8BAgMBAAGjggF/
MIIBezASBgNVHRMBAf8ECDAGAQH/AgEAMA4GA1UdDwEB/wQEAwICBDAXBgNVHSUE
EDAOBgVngQUIAQYFZ4EFCAMwHQYDVR0OBBYEFP/2zueowUhpKMuKS/LYgYG1bYCB
MB8GA1UdIwQYMBaAFEv+JlqUwfYzw4NIJt3z5bBksqqVMHYGA1UdHwRvMG0wa6Bp
oGeGZWh0dHA6Ly93d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvY3JsL0F6dXJlJTIw
VmlydHVhbCUyMFRQTSUyMFJvb3QlMjBDZXJ0aWZpY2F0ZSUyMEF1dGhvcml0eSUy
MDIwMjMuY3JsMIGDBggrBgEFBQcBAQR3MHUwcwYIKwYBBQUHMAKGZ2h0dHA6Ly93
d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvY2VydHMvQXp1cmUlMjBWaXJ0dWFsJTIw
VFBNJTIwUm9vdCUyMENlcnRpZmljYXRlJTIwQXV0aG9yaXR5JTIwMjAyMy5jcnQw
DQYJKoZIhvcNAQEMBQADggIBAEhTwx26W+Xap7zXExbAwnHYtN6kB4dIGXdgQIiQ
y5OQltsSj2jx7qZ/af5n5OnTBQ+gXM8siVipgaAIdBkbGgOjCb6b6MTe1YpFAH4f
Qv8eVwTVDziBMD0EKI30h0JbFKLdSdKe48O9Lw+T2b0PBXvMFJOSdZT7meGIkpNx
SqmAZ+RNyreLxil9knNjF5ymPT0RcGK52+MwGlElBb/jc+snhr+ZJZ1grjFky9Nz
```

```
jCTiE5SG+6H3YgiHCqfXr0L3NRt/QZ5IgkuGkNPeMvn4JjevFwAhXFxBqJYJ7mY6
1MJuWTdyhhoUJgzmZo1hS+GNyuMKaKBLreUwtc1y7LRH3YGGed57HbQ9bmyMdhO7
x8KZNrBDX2/cRLzrCmpSUldmKMu9G4dpzXpde4pFMObiVFrGRq8/9HMOJwZlQvzh
wQp0PUMY/gIU5rf23n1M1M36tM5g5CEzxQUGtVaG9ABTJQ2zijD5wDo840vbznyK
t3ihimrUs+LqpPDNXyxbwvibcZidwSdhu0QmUoyYsgSP2Zff5E8ks53h2xQSM3zz
2qaWVS1vVqG4zC0EfRnO65ogPPfrtK6ZiFmVHSWP9vPkFcUNYDnYQXW/TArO/JCe
2I++GClM7AcDQwWLxcopzskGQdHNM1zMsprRRwYaVpTJH67xeNda6+Y7IOPJYTvy
oXHPoIILVDCCBZwwggOEoAMCAQICEzMAAAACwNF7S4+XnDUAAAAAAAIwDQYJKoZI
hvcNAQEMBQAwaTELMAkGA1UEBhMCVVMxHjAcBgNVBAoTFU1pY3Jvc29mdCBDb3Jw
b3JhdGlvbjE6MDgGA1UEAxMxQXp1cmUgVmlydHVhbCBUUE0gUm9vdCBDZXJ0aWZp
Y2F0ZSBBdXRob3JpdHkgMjAyMzAeFw0yMzA2MDgxNzUzMDRaFw0yNTExMDMxNzUz
MDRaMCUxIzAhBgNVBAMTGkdsb2JhbCBWaXJ0dWFsIFRQTSBDQSAtIDAxMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvzaqLhxgaL02ncNRizEGDcVqxHIj
0oe4BO1cbH22zwhNEn0zvNDhdq2i3UWDdoBFhNg6/Pkw1LgG/1gRcnM4VNWhxT7L
IPAd/w0+qU15Q5aISyz1pI2FPjrk9fy7O0c40AXtrHvzCcj2iCVyOioAsHj8OxHF
hQoXvGgk1DB2ARtCbuMTZTE1Hf7XLumudwl5+jauPLpsJWiXpUYfotX9L3MAdgmP
7/kZYsG//AzORHfRVB2TisqeVZ2sPRlpaDOOrC0E/33RwdI5roHzWc2n5JIUAGIj
S942DWBWR+fyIFH2Cf8sm8J8KMTnVdjVdENovj+T3G9FjRmiWOO2qjo/AQIDAQAB
o4IBfzCCAXswEgYDVR0TAQH/BAgwBgEB/wIBADAOBgNVHQ8BAf8EBAMCAgQwFwYD
VR0lBBAwDgYFZ4EFCAEGBWeBBQgDMB0GA1UdDgQWBBT/9s7nqMFIaSjLikvy2IGB
tW2AgTAfBgNVHSMEGDAWgBRL/iZalMH2M8ODSCbd8+WwZLKqlTB2BgNVHR8EbzBt
MGugaaBnhmVodHRwOi8vd3d3Lm1pY3Jvc29mdC5jb20vcGtpb3BzL2NybC9BenVy
ZSUyMFZpcnR1YWwlMjBUUE0lMjBSb290JTIwQ2VydGlmaWNhdGUlMjBBdXRob3Jp
dHklMjAyMDIzLmNybDCBgwYIKwYBBQUHAQEEdzB1MHMGCCsGAQUFBzAChmdodHRw
Oi8vd3d3Lm1pY3Jvc29mdC5jb20vcGtpb3BzL2NlcnRzL0F6dXJlJTIwVmlydHVh
bCUyMFRQTSUyMFJvb3QlMjBDZXJ0aWZpY2F0ZSUyMEF1dGhvcml0eSUyMDIwMjMu
Y3J0MA0GCSqGSIb3DQEBDAUAA4ICAQBIU8Mdulvl2qe81xMWwMJx2LTepAeHSBl3
YECIkMuTkJbbEo9o8e6mf2n+Z+Tp0wUPoFzPLIlYqYGgCHQZGxoDowm+m+jE3tWK
RQB+H0L/HlcE1Q84gTA9BCiN9IdCWxSi3UnSnuPDvS8Pk9m9DwV7zBSTknWU+5nh
iJKTcUqpgGfkTcq3i8YpfZJzYxecpj09EXBiudvjMBpRJQW/43PrJ4a/mSWdYK4x
ZMvTc4wk4hOUhvuh92IIhwqn169C9zUbf0GeSIJLhpDT3jL5+CY3rxcAIVxcQaiW
Ce5mOtTCblk3coYaFCYM5maNYUvhjcrjCmigS63lMLXNcuy0R92Bhnneex20PW5s
jHYTu8fCmTawQ19v3ES86wpqUlJXZijLvRuHac16XXuKRTDm4lRaxkavP/RzDicG
ZUL84cEKdD1DGP4CFOa39t59TNTN+rTOYOQhM8UFBrVWhvQAUyUNs4ow+cA6PONL
2858ird4oYpq1LPi6qTwzV8sW8L4m3GYncEnYbtEJlKMmLIEj9mX3+RPJLOd4dsU
EjN889qmllUtb1ahuMwtBH0ZzuuaIDz367SumYhZlR0lj/bz5BXFDWA52EF1v0wK
zvyQntiPvhgpTOwHA0MFi8XKKc7JBkHRzTNczLKa0UcGGlaUyR+u8XjXWuvmOyDj
yWE78qFxzzCCBbAwggOYoAMCAQICEFH0MdoskgiKTing2SneSqcwDQYJKoZIhvcN
AQEMBQAwaTELMAkGA1UEBhMCVVMxHjAcBgNVBAoTFU1pY3Jvc29mdCBDb3Jwb3Jh
dGlvbjE6MDgGA1UEAxMxQXp1cmUgVmlydHVhbCBUUE0gUm9vdCBDZXJ0aWZpY2F0
ZSBBdXRob3JpdHkgMjAyMzAeFw0yMzA2MDExODA4NTNaFw00ODA2MDExODE1NDFa
MGkxCzAJBgNVBAYTAlVTMR4wHAYDVQQKExVNaWNyb3NvZnQgQ29ycG9yYXRpb24x
OjA4BgNVBAMTMUF6dXJlIFZpcnR1YWwgVFBNIFJvb3QgQ2VydGlmaWNhdGUgQXV0
aG9yaXR5IDIwMjMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC6DDML
3USe/m1tNGnShNVXpjaiUvtuvFK3vEHH2V1TrFTttvjy/VkaD51qNK+XVaKJmBm6
G4eBwu9rPbBmOgRkXvlItizMLlp4qwbTo2id+K9WYnjTHFTSU/3V1uRzfxy5Zv4E
ipVs6ghMeoBRQEE78mJf+Y2CODntOcye6I4Iyv+mR16RcrF58lCEyfaoI5+eOiCY
fxuTjAt5g3xVOwPxgIvi2Gjh0s14+vS5Nbwn7TavQ3EB7Qe6LTtFcRZES2lC3H7v
wGBzOxQv0qunCW/QLvCvyFIcP+ZSUl2WsCycB6XBGMxE5tm5WndFR7AAmizbnGS+
cAYvBhLrEPVqPFtada1AjKfPabn8WtwK/bGGb1niZbyoIFnEFVJYb7yGRJ0/n7Xh
L8E3aGo5eakr2Wwzxgj42/da0s56Ii6vF6oEAQRY5xIky/lYmmZsmT2FLAbwWN5S
lwdZdoYIGoXwSGoOI9MHIKpS/Nki+6QHRo6tBaLukioB55JK1eQjcVAQVTDQsFKI
+ZrxI3eZhe1z4HzPfKjQAW9gKrX5flbgpFPxtV7xaE6kvQJMGy0xhTEvfowhFNq8
OsJyrhr9U8fYrknTb7F7fLLceH5FXib5xaSXgCMAWVYCkUk2t8N9PfZNJlvKhojJ
8AiOUn3Wdkicg+LhJrM7J4nhOsAFndbSy4NY7wIDAQABo1QwUjAOBgNVHQ8BAf8E
```

```
BAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUS/4mWpTB9jPDg0gm3fPl
sGSyqpUwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEMBQADggIBAC4DQMoP
CNADTTv/CNgYaUzrG8sDdmEpkoQJooOYeLArWiUjNXUV8APpHmOjwl0BLfz0QEAI
hzXDhec27jlNQcP2eWe5PaJLdOHrvArEtgaXPjzoPY7OpQbZooDOS1iQh9YKDA9J
CWKc5Ggs8dU3Q98o6NMUsYBoWm5OSLtbKA9zsYVMiBXMWpbocJIlSDbdyYcuxu1S
U5RMJJNtQKDq0MVLCR9H9hl42EoUta/aiz8y3+8pHDputoxxpi2VWbowcyyHTZNW
5Yz/VbhlvnCEgcM5DXhDDBbn3C9sYerP6IlJ14nBVrxuxOoqD52smF6x8t0YevFy
54X6c3gs728/yOMjFd/KhqP3AGNjiaV1Lrg/2WyOa0/pUQRJA1tZMRz/HKcJEcBq
KVfPoF9iklLnCCDeJfP66zQxF/E9G+VPgVKoju2eWJ8Qsmvx9GSMJkJ05S0U1Nn1
vjhhvsciK4zIv9Jev8lj1JCAJDbb0v8ju1EkVSz27RlZN0K3uh3KmnAl8xIyDXCY
RhFYPN2PJzPmPmvSdJVJx8rTEhWRPOYfPDrHkUtkMKo6M1KNo9Q6MJoUklL/TWuQ
HttsnHIgZBGUqvGuK0pcKrYWhCzaXqEy1JOolaFJCv4i72jL65HY/ciWNTv4L5Xz
dmojU0k/9m3fjwOEQU6Bva15+rzZFFnWy4m1MQA=
-----END CERTIFICATE-----
```

▼ Select to view the intermediate CA content (ICA-01)

"Global Virtual TPM CA - 01" (intermediate CA) [.cer]:

```
-----BEGIN CERTIFICATE-----
MIIFnDCCA4SgAwIBAgITMwAAAALA0XtLj5ecNQAAAAAAAjANBgkqhkiG9w0BAQwF
ADBpMQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9u
MTowOAYDVQQDEzFBenVyZSBWaXJ0dWFsIFRQTSBSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eSAyMDIzMB4XDTIzMDYwODE3NTMwNFoXDTI1MTEwMzE3NTMwNFowJTEj
MCEGA1UEAxMaR2xvYmFsIFZpcnR1YWwgVFBNIENBIC0gMDEwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQC/NqouHGBovTadw1GLMQYNxWrEciPSh7gE7Vxs
fbbPCE0SfTO80OF2raLdRYN2gEWE2Dr8+TDUuAb/WBFyczhU1aHFPssg8B3/DT6p
TXlDlohLLPWkjYU+OuT1/Ls7RzjQBe2se/MJyPaIJXI6KgCwePw7EcWFChe8aCTU
MHYBG0Ju4xNlMTUd/tcu6a53CXn6Nq48umwlaJelRh+i1f0vcwB2CY/v+Rliwb/8
DM5Ed9FUHZOKyp5Vnaw9GWloM46sLQT/fdHB0jmugfNZzafkkhQAYiNL3jYNYFZH
5/IgUfYJ/yybwnwoxOdV2NV0Q2i+P5Pcb0WNGaJY47aqOj8BAgMBAAGjggF/MIIB
ezASBgNVHRMBAf8ECDAGAQH/AgEAMA4GA1UdDwEB/wQEAwICBDAXBgNVHSUEEDAO
BgVngQUIAQYFZ4EFCAMwHQYDVR0OBBYEFP/2zueowUhpKMuKS/LYgYG1bYCBMB8G
A1UdIwQYMBaAFEv+JlqUwfYzw4NIJt3z5bBksqqVMHYGA1UdHwRvMG0wa6BpoGeG
ZWh0dHA6Ly93d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvY3JsL0F6dXJlJTIwVmly
dHVhbCUyMFRQTSUyMFJvb3QlMjBDZXJ0aWZpY2F0ZSUyMEF1dGhvcml0eSUyMDIw
MjMuY3JsMIGDBggrBgEFBQcBAQR3MHUwcwYIKwYBBQUHMAKGZ2h0dHA6Ly93d3cu
bWljcm9zb2Z0LmNvbS9wa2lvcHMvY2VydHMvQXp1cmUlMjBWaXJ0dWFsJTIwVFBN
JTIwUm9vdCUyMENlcnRpZmljYXRlJTIwQXV0aG9yaXR5JTIwMjAyMy5jcnQwDQYJ
KoZIhvcNAQEMBQADggIBAEhTwx26W+Xap7zXExbAwnHYtN6kB4dIGXdgQIiQy5OQ
ltsSj2jx7qZ/af5n5OnTBQ+gXM8siVipgaAIdBkbGgOjCb6b6MTe1YpFAH4fQv8e
VwTVDziBMD0EKI30h0JbFKLdSdKe48O9Lw+T2b0PBXvMFJOSdZT7meGIkpNxSqmA
Z+RNyreLxil9knNjF5ymPT0RcGK52+MwGlElBb/jc+snhr+ZJZ1grjFky9NzjCTi
E5SG+6H3YgiHCqfXr0L3NRt/QZ5IgkuGkNPeMvn4JjevFwAhXFxBqJYJ7mY61MJu
WTdyhhoUJgzmZo1hS+GNyuMKaKBLreUwtc1y7LRH3YGGed57HbQ9bmyMdhO7x8KZ
NrBDX2/cRLzrCmpSUldmKMu9G4dpzXpde4pFMObiVFrGRq8/9HMOJwZlQvzhwQp0
PUMY/gIU5rf23n1M1M36tM5g5CEzxQUGtVaG9ABTJQ2zijD5wDo840vbznyKt3ih
imrUs+LqpPDNXyxbwvibcZidwSdhu0QmUoyYsgSP2Zff5E8ks53h2xQSM3zz2qaW
VS1vVqG4zC0EfRnO65ogPPfrtK6ZiFmVHSWP9vPkFcUNYDnYQXW/TArO/JCe2I++
```

```
GClM7AcDQwWLxcopzskGQdHNM1zMsprRRwYaVpTJH67xeNda6+Y7IOPJYTvyoXHP
-----END CERTIFICATE-----
```


▼ Select to view the .p7b content (Root + ICA-03)

Global Virtual TPM CA - 03:


```
-----BEGIN CERTIFICATE-----
MIIRKQYJKoZIhvcNAQcCoIIRGjCCERYCAQExADCCBbMGCSqGSIb3DQEHAaCCBaQE
ggWgMIIFnDCCA4SgAwIBAgITMwAAAAknQOWscnsOpgAAAAAACTANBgkqhkiG9w0B
AQwFADBpMQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0
aW9uMTowOAYDVQQDEzFBenVyZSBWaXJ0dWFsIFRQTSBSb290IENlcnRpZmljYXRl
IEF1dGhvcml0eSAyMDIzMB4XDTI1MDQyNDE4MDExN1oXDTI3MDQyNDE4MDExN1ow
JTEjMCEGA1UEAxMaR2xvYmFsIFZpcnR1YWwgVFBNIENBIC0gMDMwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDYGYtis5ka0cxQkhU11jslgX6wzjR/UXQI
FdUn8juTUMJl91VokwUPX3WfXeog7mtbWyYWD8SI0BSnchRGlV8u3AhcW61/HetH
qmILtD0c75UATi+gsTQnpwKPA/m38MGGyXFETr3xHXjilUPfIhmxO4ImuNJ0R95b
ZYhxbLYmOZpVUcj8oz980An8HlIqSzrskQR6NiuEmikHkHc1/CpoNunrr8kQNPF6
gxexIrvXsKLUAuUqnNtcQWc/8Er5EN9+TdX6AOjUmKriVGbCInP1m/aC+DWH/+aJ
/8aDpKze6fe7OHh2BL9hxqIsmJAStIh4siRdLYTt8hKGmkdzOWnRAgMBAAGjggF/
MIIBezASBgNVHRMBAf8ECDAGAQH/AgEAMA4GA1UdDwEB/wQEAwICBDAXBgNVHSUE
EDAOBgVngQUIAQYFZ4EFCAMwHQYDVR0OBBYEFGcJhvj5gV6TrfnJZOcUCtqZywot
MB8GA1UdIwQYMBaAFEv+JlqUwfYzw4NIJt3z5bBksqqVMHYGA1UdHwRvMG0wa6Bp
oGeGZWh0dHA6Ly93d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvY3JsL0F6dXJlJTIw
VmlydHVhbCUyMFRQTSUyMFJvb3QlMjBDZXJ0aWZpY2F0ZSUyMEF1dGhvcml0eSUy
MDIwMjMuY3JsMIGDBggrBgEFBQcBAQR3MHUwcwYIKwYBBQUHMAKGZ2h0dHA6Ly93
d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvY2VydHMvQXp1cmUlMjBWaXJ0dWFsJTIw
VFBNJTIwUm9vdCUyMENlcnRpZmljYXRlJTIwQXV0aG9yaXR5JTIwMjAyMy5jcnQw
DQYJKoZIhvcNAQEMBQADggIBAJPP3Z2z1zhzUS3qSRVgyoUVnaxCGuMHzPQAZuoP
BVpzwKnv4HqyjMgT8pBtQqxkqAsg7KiqbPfO97bMCHcuqkkfHjw8yg6IYt01RjUj
VPKqlrsY2iw7hFWNWr8SGMa10JdNYNyf5dxob5+mKAwEOhLzKNwq9rM/uIvZky77
pNlyRLt55XEPfBMYdI9I8uQ5Uqmrw7mVJfERMfTBhSQF9BrcajAsaLcs7qEUyj0y
UdJfcgZkfCoUEUSPr3OwLHaYeV1J6VidhIYsYo53sXXal91d60NspYgei2nJFei/
+R3ESWnGbPBW+EQ4FbvZXxu57zUMX9mM7lC+GoXLvA6/vtKShEi9ZXl2PSnBQ/R2
A7b3AXyg4fmMLFausEk6OiuU8E/bvp+gPLOJ8YrX7SAJVuEn+koJaK5G7os5DMIh
7/KMl9cI9WxPwqoWjp4VBfrF4hDOCmKWrqtFUDQCML8qD8RTxlQKQtgeGAcNDfoA
uL9KVtSG5/iIhuyBEFYEHa3vRWbSaHCUzaHJsTmLcz4cp1VDdepzqZRVuErBzJKF
nBXbzRNW32EFmcAUKZImIsE5dgB7y7eiijf33VWNfWmK05fxzQziWFWRYlET4SVc
3jMnPBiY3N8BfK8EBOYbLvzo0qn2n3SAmPhYX3Ag6vbbIHd4Qc8DQKHRV0PB8D3j
PGmDoIILVDCCBZwwggOEoAMCAQICEzMAAAAJJ0DlrHJ7DqYAAAAAAAkwDQYJKoZI
hvcNAQEMBQAwaTELMAkGA1UEBhMCVVMxHjAcBgNVBAoTFU1pY3Jvc29mdCBDb3Jw
b3JhdGlvbjE6MDgGA1UEAxMxQXp1cmUgVmlydHVhbCBUUE0gUm9vdCBDZXJ0aWZp
Y2F0ZSBBdXRob3JpdHkgMjAyMzAeFw0yNTA0MjQxODAxMTdaFw0yNzA0MjQxODAx
MTdaMCUxIzAhBgNVBAMTGkdsb2JhbCBWaXJ0dWFsIFRQTSBDQSAtIDAzMIIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2BmLYrOZGtHMUJIVNdY7JYF+sM40
f1F0CBXVJ/I7k1DCZfdVaJMFD191n13qIO5rW1smFg/EiNAUp3IURpVfLtwIXFut
fx3rR6piC7Q9HO+VAE4voLE0J6cCjwP5t/DBhslxRE698R144pVD3yIZsTuCJrjS
dEfeW2WIcWy2JjmaVVHI/KM/fNAJ/B5SKks67JEEejYrhJopB5B3NfwqaDbp66/J
EDTxeoMXsSK717Ci1ALlKpzbXEFnP/BK+RDffk3V+gDo1Jiq4lRmwiJz9Zv2gvg1
h//mif/Gg6Ss3un3uzh4dgS/YcaiLJiQErSIeLIkXS2E7fIShppHczlp0QIDAQAB
o4IBfzCCAXswEgYDVR0TAQH/BAgwBgEB/wIBADAOBgNVHQ8BAf8EBAMCAgQwFwYD
```

```
VR0lBBAwDgYFZ4EFCAEGBWeBBQgDMB0GA1UdDgQWBBRnCYb4+YFek635yWTnFAra
mcsKLTAfBgNVHSMEGDAWgBRL/iZalMH2M8ODSCbd8+WwZLKqlTB2BgNVHR8EbzBt
MGugaaBnhmVodHRwOi8vd3d3Lm1pY3Jvc29mdC5jb20vcGtpb3BzL2NybC9BenVy
ZSUyMFZpcnR1YWwlMjBUUE0lMjBSb290JTIwQ2VydGlmaWNhdGUlMjBBdXRob3Jp
dHklMjAyMDIzLmNybDCBgwYIKwYBBQUHAQEEdzB1MHMGCCsGAQUFBzAChmdodHRw
Oi8vd3d3Lm1pY3Jvc29mdC5jb20vcGtpb3BzL2NlcnRzL0F6dXJlJTIwVmlydHVh
bCUyMFRQTSUyMFJvb3QlMjBDZXJ0aWZpY2F0ZSUyMEF1dGhvcml0eSUyMDIwMjMu
Y3J0MA0GCSqGSIb3DQEBDAUAA4ICAQCTz92ds9c4c1Et6kkVYMqFFZ2sQhrjB8z0
AGbqDwVac8Cp7+B6sozIE/KQbUKsZKgLIOyoqmz3zve2zAh3LqpJHx48PMoOiGLd
NUY1I1Tyqpa7GNosO4RVjVq/EhjGtdCXTWDcn+XcaG+fpigMBDoS8yjcKvazP7iL
2ZMu+6TZckS7eeVxD3wTGHSPSPLkOVKpq8O5lSXxETH0wYUkBfQa3GowLGi3LO6h
FMo9MlHSX3IGZHwqFBFEj69zsCx2mHldSelYnYSGLGKOd7F12pfdXetDbKWIHotp
yRXov/kdxElpxmzwVvhEOBW72V8bue81DF/ZjO5QvhqFy7wOv77SkoRIvWV5dj0p
wUP0dgO29wF8oOH5jCxWrrBJOjorlPBP276foDyzifGK1+0gCVbhJ/pKCWiuRu6L
OQzCIe/yjJfXCPVsT8KqFo6eFQX6xeIQzgpilq6rRVA0AjC/Kg/EU8ZUCkLYHhgH
DQ36ALi/SlbUhuf4iIbsgRBWBB2t70Vm0mhwlM2hybE5i3M+HKdVQ3Xqc6mUVbhK
wcyShZwV280TVt9hBZnAFCmSJiLBOXYAe8u3ooo3991VjX1pitOX8c0M4lhVkWJR
E+ElXN4zJzwYmNzfAXyvBATmGy786NKp9p90gJj4WF9wIOr22yB3eEHPA0Ch0VdD
wfA94zxpgzCCBbAwggOYoAMCAQICEFH0MdoskgiKTing2SneSqcwDQYJKoZIhvcN
AQEMBQAwaTELMAkGA1UEBhMCVVMxHjAcBgNVBAoTFU1pY3Jvc29mdCBDb3Jwb3Jh
dGlvbjE6MDgGA1UEAxMxQXp1cmUgVmlydHVhbCBUUE0gUm9vdCBDZXJ0aWZpY2F0
ZSBBdXRob3JpdHkgMjAyMzAeFw0yMzA2MDExODA4NTNaFw00ODA2MDExODE1NDFa
MGkxCzAJBgNVBAYTAlVTMR4wHAYDVQQKExVNaWNyb3NvZnQgQ29ycG9yYXRpb24x
OjA4BgNVBAMTMUF6dXJlIFZpcnR1YWwgVFBNIFJvb3QgQ2VydGlmaWNhdGUgQXV0
aG9yaXR5IDIwMjMwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQC6DDML
3USe/m1tNGnShNVXpjaiUvtuvFK3vEHH2V1TrFTttvjy/VkaD51qNK+XVaKJmBm6
G4eBwu9rPbBmOgRkXvlItizMLlp4qwbTo2id+K9WYnjTHFTSU/3V1uRzfxy5Zv4E
ipVs6ghMeoBRQEE78mJf+Y2CODntOcye6I4Iyv+mR16RcrF58lCEyfaoI5+eOiCY
fxuTjAt5g3xVOwPxgIvi2Gjh0s14+vS5Nbwn7TavQ3EB7Qe6LTtFcRZES2lC3H7v
wGBzOxQv0qunCW/QLvCvyFIcP+ZSUl2WsCycB6XBGMxE5tm5WndFR7AAmizbnGS+
cAYvBhLrEPVqPFtada1AjKfPabn8WtwK/bGGb1niZbyoIFnEFVJYb7yGRJ0/n7Xh
L8E3aGo5eakr2Wwzxgj42/da0s56Ii6vF6oEAQRY5xIky/lYmmZsmT2FLAbwWN5S
lwdZdoYIGoXwSGoOI9MHIKpS/Nki+6QHRo6tBaLukioB55JK1eQjcVAQVTDQsFKI
+ZrxI3eZhe1z4HzPfKjQAW9gKrX5flbgpFPxtV7xaE6kvQJMGy0xhTEvfowhFNq8
OsJyrhr9U8fYrknTb7F7fLLceH5FXib5xaSXgCMAWVYCkUk2t8N9PfZNJlvKhojJ
8AiOUn3Wdkicg+LhJrM7J4nhOsAFndbSy4NY7wIDAQABo1QwUjAOBgNVHQ8BAf8E
BAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUS/4mWpTB9jPDg0gm3fPl
sGSyqpUwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEMBQADggIBAC4DQMoP
CNADTTv/CNgYaUzrG8sDdmEpkoQJooOYeLArWiUjNXUV8APpHmOjwl0BLfz0QEAI
hzXDhec27jlNQcP2eWe5PaJLdOHrvArEtgaXPjzoPY7OpQbZooDOS1iQh9YKDA9J
CWKc5Ggs8dU3Q98o6NMUsYBoWm5OSLtbKA9zsYVMiBXMWpbocJIlSDbdyYcuxu1S
U5RMJJNtQKDq0MVLCR9H9hl42EoUta/aiz8y3+8pHDputoxxpi2VWbowcyyHTZNW
5Yz/VbhlvnCEgcM5DXhDDBbn3C9sYerP6IlJ14nBVrxuxOoqD52smF6x8t0YevFy
54X6c3gs728/yOMjFd/KhqP3AGNjiaV1Lrg/2WyOa0/pUQRJA1tZMRz/HKcJEcBq
KVfPoF9iklLnCCDeJfP66zQxF/E9G+VPgVKoju2eWJ8Qsmvx9GSMJkJ05S0U1Nn1
vjhhvsciK4zIv9Jev8lj1JCAJDbb0v8ju1EkVSz27RlZN0K3uh3KmnAl8xIyDXCY
RhFYPN2PJzPmPmvSdJVJx8rTEhWRPOYfPDrHkUtkMKo6M1KNo9Q6MJoUklL/TWuQ
HttsnHIgZBGUqvGuK0pcKrYWhCzaXqEy1JOolaFJCv4i72jL65HY/ciWNTv4L5Xz
dmojU0k/9m3fjwOEQU6Bva15+rzZFFnWy4m1MQA=
-----END CERTIFICATE-----
```

▼ Select to view the intermediate CA content (ICA-03)

"Global Virtual TPM CA - 03" (intermediate CA) [.cer]:

```
-----BEGIN CERTIFICATE-----
MIIFnDCCA4SgAwIBAgITMwAAAAknQOWscnsOpgAAAAAACTANBgkqhkiG9w0BAQwF
ADBpMQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9u
MTowOAYDVQQDEzFBenVyZSBWaXJ0dWFsIFRQTSBSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eSAyMDIzMB4XDTI1MDQyNDE4MDExN1oXDTI3MDQyNDE4MDExN1owJTEj
MCEGA1UEAxMaR2xvYmFsIFZpcnR1YWwgVFBNIENBIC0gMDMwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDYGYtis5ka0cxQkhU11jslgX6wzjR/UXQIFdUn
8juTUMJl91VokwUPX3WfXeog7mtbWyYWD8SI0BSnchRGlV8u3AhcW61/HetHqmIL
tD0c75UATi+gsTQnpwKPA/m38MGGyXFETr3xHXjilUPfIhmxO4ImuNJ0R95bZYhx
bLYmOZpVUcj8oz980An8HlIqSzrskQR6NiuEmikHkHc1/CpoNunrr8kQNPF6gxex
IrvXsKLUAuUqnNtcQWc/8Er5EN9+TdX6AOjUmKriVGbCInP1m/aC+DWH/+aJ/8aD
pKze6fe7OHh2BL9hxqIsmJAStIh4siRdLYTt8hKGmkdzOWnRAgMBAAGjggF/MIIB
ezASBgNVHRMBAf8ECDAGAQH/AgEAMA4GA1UdDwEB/wQEAwICBDAXBgNVHSUEEDAO
BgVngQUIAQYFZ4EFCAMwHQYDVR0OBBYEFGcJhvj5gV6TrfnJZOcUCtqZywotMB8G
A1UdIwQYMBaAFEv+JlqUwfYzw4NIJt3z5bBksqqVMHYGA1UdHwRvMG0wa6BpoGeG
ZWh0dHA6Ly93d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvY3JsL0F6dXJlJTIwVmly
dHVhbCUyMFRQTSUyMFJvb3QlMjBDZXJ0aWZpY2F0ZSUyMEF1dGhvcml0eSUyMDIw
MjMuY3JsMIGDBggrBgEFBQcBAQR3MHUwcwYIKwYBBQUHMAKGZ2h0dHA6Ly93d3cu
bWljcm9zb2Z0LmNvbS9wa2lvcHMvY2VydHMvQXp1cmUlMjBWaXJ0dWFsJTIwVFBN
JTIwUm9vdCUyMENlcnRpZmljYXRlJTIwQXV0aG9yaXR5JTIwMjAyMy5jcnQwDQYJ
KoZIhvcNAQEMBQADggIBAJPP3Z2z1zhzUS3qSRVgyoUVnaxCGuMHzPQAZuoPBVpz
wKnv4HqyjMgT8pBtQqxkqAsg7KiqbPfO97bMCHcuqkkfHjw8yg6IYt01RjUjVPKq
lrsY2iw7hFWNWr8SGMa10JdNYNyf5dxob5+mKAwEOhLzKNwq9rM/uIvZky77pNly
RLt55XEPfBMYdI9I8uQ5Uqmrw7mVJfERMfTBhSQF9BrcajAsaLcs7qEUyj0yUdJf
cgZkfCoUEUSPr3OwLHaYeV1J6VidhIYsYo53sXXal91d60NspYgei2nJFei/+R3E
SWnGbPBW+EQ4FbvZXxu57zUMX9mM7lC+GoXLvA6/vtKShEi9ZXl2PSnBQ/R2A7b3
AXyg4fmMLFausEk6OiuU8E/bvp+gPLOJ8YrX7SAJVuEn+koJaK5G7os5DMIh7/KM
l9cI9WxPwqoWjp4VBfrF4hDOCmKWrqtFUDQCML8qD8RTxlQKQtgeGAcNDfoAuL9K
VtSG5/iIhuyBEFYEHa3vRWbSaHCUzaHJsTmLcz4cp1VDdepzqZRVuErBzJKFnBXb
zRNW32EFmcAUKZImIsE5dgB7y7eiijf33VWNfWmK05fxzQziWFWRYlET4SVc3jMn
PBiY3N8BfK8EBOYbLvzo0qn2n3SAmPhYX3Ag6vbbIHd4Qc8DQKHRV0PB8D3jPGmD
-----END CERTIFICATE-----
```

What Microsoft owned trusted certificates are built into Azure VMs?

For Windows VMs, Windows CA certificate is built in UEFI firmware. For Linux VMs, Microsoft UEFI CA certificate is built in UEFI firmware.

For Azure Linux VMs only, `Azure Services Linux Kmod PCA` certificate is also added in UEFI firmware for all Linux distributions. Linux Kmod PCA is used to sign Microsoft owned kernel modules.

Linux Kmod PCA certificate is added to make customer experience smoother when using Microsoft solutions like Azure Site Recovery (Site recovery) which installs a kernel module. The

Site recovery kernel module loads without any customer action to supply a key as Site recovery kernel module is signed using the trusted 'Azure Services Linux Kmod PCA' certificate.

Download instructions

Package certificates, composed of .p7b and .cer reveal the signing and certificate authority. Copy the relevant content and use certificate tooling to inspect and assess details of certificates.

▼ Select to download the .crt that follows for the Microsoft Windows CA

Microsoft Windows CA

▼ Select to download the .crt that follows for the Microsoft UEFI CA

Microsoft UEFI CA

▼ Select to view the .p7b content

Azure services Linux kmod PCA certificate:

```
-----BEGIN CERTIFICATE-----
MIIGrjCCBJagAwIBAgITMwAAAATCM9cMfybr0QAAAAAABDANBgkqhkiG9w0BAQwF
ADBbMQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9u
MSwwKgYDVQQDEyNNaWNyb3NvZnQgUlNBIFNlcnZpY2VzIFJvb3QgQ0EgMjAyMTAe
Fw0yMzA4MDgxODE0NTVaFw0zODA4MDgxODI0NTVaMFUxCzAJBgNVBAYTAlVTMR4w
HAYDVQQKExVNaWNyb3NvZnQgQ29ycG9yYXRpb24xJjAkBgNVBAMTHUF6dXJlIFNl
cnZpY2VzIExpbnV4IEttb2QgUENBMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIIC
CgKCAgEAwQAMp1T5lFW9RKdeuXVts2Wcim44ObsCNa4PVMfdpiPNOCPCYFBkyB2o
k/s/8Is5pYxNkjcvNdYPKW+8E8IC1HU6Vj+jR+sdtuFX1mbYV9I4LWNZEWHr/FHn
A2lk8QLGwj9HfElQxKNjEtgkJPfvtp5B3XlPkhMzxZCdzqWZk9qNd8l9PaccSidC
m/BB8dBbf7MirXAphT9FPn5gNAUqmc2Sz2/HcGPp0n1X3VMf/9gemri/MEKScO6r
byLT7rpLnVUNWfSVARM35e0cFkyGfYtDh4LgrNUnl2lpZg/nvCdeR4k4mgYbYGWQ
EppENAJ8Hh3gxKiG7phYShxxG+x5NZdIqhBa71VGgrlqys/9ybZNsqW5iBjleflQ
L7SWz4vbZGkVNDQ1tpWF/UrM4rHfmLiXhofDmN3/lYZ4veeyMkktvmLk9RUcO9X4
MzKVipZGr9a6gDIU5obNAuD5enny3ejD1ny6azSbRY6YYgJx/zgxg93wbVgVMljy
ke8y0QDtZfDi078AuOWhrUzw4t87NfdlZ/NmAJIcildRaICDes6/kW5AOyCfZqV4
vXVD7dokC8pbt7hvmTZeWrGBTSPvo8PiJvRdhQYE2lDiOjtXFElcDG6/xs4XLrUy
P/U1L2Q7F7GgA46KhQILqYkJhmEgrSVc4EsZ2xDFmdHRpSM5EjsCAwEAAaOCAW8w
ggFrMA4GA1UdDwEB/wQEAwIBhjAQBgkrBgEEAYI3FQEEAwIBADAdBgNVHQ4EFgQU
aXzE9BoiCaer0GBMHxlvTYxXNdkwGQYJKwYBBAGCNxQCBAweCgBTAHUAYgBDAEEw
DwYDVR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBQODLFkab0tsdVrJqZH6lZOgMPt
ijBmBgNVHR8EXzBdMFugWaBXhlVodHRwOi8vd3d3Lm1pY3Jvc29mdC5jb20vcGtp
b3BzL2NybC9NaWNyb3NvZnQlMjBSU0ElMjBTZXJ2aWNlcyUyMFJvb3QlMjBDQSUy
MDIwMjEuY3JsMHMGCCsGAQUFBwEBBGcwZTBjBggrBgEFBQcwAoZXaHR0cDovL3d3
dy5taWNyb3NvZnQuY29tL3BraW9wcy9jZXJ0cy9NaWNyb3NvZnQlMjBSU0ElMjBT
ZXJ2aWNlcyUyMFJvb3QlMjBDQSUyMDIwMjEuY3J0MA0GCSqGSIb3DQEBDAUAA4IC
AQB2eZw8jfATH/wiLEpIA4Npc3+f6KhmsMlbC5o+ud3mkKMy7O3IgUP3nITvtFPV
ekyfGQZB3Hm0dOVCNaGZ6BLYD4iEXD6I1Z2XqUayKitGZaagOjAjr2piRpcwGSql
V8lVq1EfKH/iJYf/408D/hkH8M/6TQqZRSjpsmeX/PxYXKZrEKr6XQsUy1dGq7oR
UTc6WU0iy7WMagrqQAQlGpZpSehhoGvodwJoSGhPM9/GDIiEiwXT2hkswJX8/MQQ
```

```
T8O9Is0aLAgf+bwuk9Ng8TDNr/m8B2VXYrfcW2OTlJy7kXh8LmiQfxV7JtS8UKSx
OL+AXJtcAn3MBMscG+Lb3SGoQywGeNqCeeglIvMeOYhFrQ5WT3Ob04ZHAbt+aQ/r
psceMSMsE3RwcCANZhq98/6kh8cUsblmJbBgUV4pFJtEMjmUByZa8aRfABM7FUZ4
My/GJL6iPgcqzCTofkc7Z50Fa3NjXEyGzWMae3mS/djRePlr3RaTBTELCdtHVG/H
OFIyldD/wdlzvoOIkNc7UoIQPbjvxM0qCb6ruQiifCjvo8KFXjdmft/Dh3h60idH
g2Zz8q2u48vpBVBwxv5mJOF6ioYTpJpMtLfFgAOdTxIma89Ibja866Sr73Sg/J0b
KoeRpZWk8vqiJUl9PAk+JMqy8Fe/DoKq5OYT3zeZuVxGUQ==
-----END CERTIFICATE-----

Linux kmod PCA Details

Serial Number # 3300000004c233d70c7f26ebd1000000000004
Valid Until # 08/Aug/2038
```

Last updated on 02/10/2026

Secure Boot UEFI keys

Article • 02/28/2025

For certain advanced scenarios, it's necessary for customizing secure boot keys. The secure boot UEFI keys customization allows you to modify unified extensible firmware interface (UEFI) keys – PK, KEK, DB, DBX – in your image for a secure boot capable Azure virtual machine (VM) (Trusted Launch and Confidential VM). With this feature, UEFI keys can be fully replaced or appended to default key databases.

When a secure boot Azure VM is deployed, signatures of all the boot components such as UEFI, shim/bootloader, kernel, and kernel modules/drivers are verified during the boot process. Verification fails if the boot component signatures don't match with a key in the trusted key databases, and the VM fails to boot. This failure can occur if: a component is signed by a key not found in the trusted key databases, the signing key is listed in the revoked key database, or a component is unsigned. Through custom secure boot UEFI keys, the necessary signatures can be supplied for a successful VM boot.

UEFI key customization is offered through Azure compute gallery resources. If you currently use a Marketplace image to create a Trusted Launch VM, you must first create Azure compute gallery resources using the Marketplace image to modify UEFI keys.

> **Note**
>
> Customizing UEFI keys is for advanced customers intimately familiar with inner workings of secure boot technology.

Prerequisites

The following Azure Compute Gallery resources are necessary for customizing UEFI keys:

- Gallery
- Image definition
- Image version

Currently, UEFI key customization is available only through the REST API.

You must attach a *VM image version* resource to a *VM image definition* resource with security-type set to *TrustedLaunchSupported* or *TrustedLaunchAndConfidentialVmSupported*. For detailed instructions on how to create an image definition resource, see Create an image definition and an image version. To create a VM image definition resource, you must also create a gallery resource. You can

choose the Azure portal, PowerShell, CLI, or REST API to create gallery and image definition resources.

ARM template

A generic ARM template is provided as a sample for creating an image version resource. For customizing UEFI keys, you also need to include *security-profile* block in the image version resource. Each scenario in the Scenarios section provides details on what goes inside the *securityProfile* block.

JSON

```
{
    "apiVersion": "2022-03-03",
    "type": "Microsoft.Compute/galleries/images/versions",
    "dependsOn": [
        "[concat('Microsoft.Compute/galleries/', parameters('galleryName'),
'/images/', parameters('imageDefinitionName'))]"
    ],
    "name": "[concat(parameters('galleryName'), '/',
parameters('imageDefinitionName'), '/', parameters('versionName'))]",
    "location": "[parameters('location')]",
    "properties": {
        "publishingProfile": {
            "replicaCount": "[parameters('defaultReplicaCount')]",
            "targetRegions": "[parameters('regionReplications')]",
            "excludeFromLatest": "[parameters('excludedFromLatest')]",
            "replicationMode": "[parameters('replicationMode')]"
        },
        "storageProfile": {
            "source": {
                "id": "[parameters('sourceVmId')]"
            }
        },
        "safetyProfile": {
            "allowDeletionOfReplicatedLocations": "
[parameters('allowDeletionOfReplicatedLocations')]"
        },
        "securityProfile": {}
    },
    "tags": {}
}
```

Scenarios

This section contains instructions on how to customize UEFI keys for common scenarios.

Scenario 1: Replace all UEFI keys

In this scenario, you fully replace all default keys (PK, KEK, DB, DBX) available in a Trusted Launch VM.

1. Create certificates and/or keys for each or PK, KEK, DB, and DBX. Refer to the Useful links section for detailed instructions for various operating systems.
2. Create a new image version. Use the following *securityProfile* ARM template block with the image version creation Generic ARM template. Replace `Base64 formatted certificate` and `Base64 formatted sha256 hash` JSON values with the real values.

JSON

```
"securityProfile": {
    "uefiSettings": {
        "signatureTemplateNames": [
            "NoSignatureTemplate"
        ],
        "additionalSignatures": {
            "pk": {
                "type": "x509",
                "value": [
                    "Base64 formatted certificate"
                ]
            },
            "kek": [
                {
                    "type": "x509",
                    "value": [
                        "Base64 formatted certificate"
                    ]
                }
            ],
            "db": [
                {
                    "type": "x509",
                    "value": [
                        "Base64 formatted certificate",
                        "Base64 formatted certificate"
                    ]
                }
            ],
            "dbx": [
                {
                    "type": "x509",
                    "value": [
                        "Base64 formatted certificate",
                        "Base64 formatted certificate"
                    ]
                },
                {
```

```
                    "type": "sha256",
                    "value": [
                        "Base64 formatted sha256 hash",
                        "Base64 formatted sha256 hash"
                    ]
                }
            ]
        }
    }
}
```

3. Create a Trusted Launch VM using Azure portal, PowerShell, CLI, or REST API with the same source image as the one you created using Azure compute gallery resources.
4. Once the Trusted Launch VM is in the *running* state, verify that your custom UEFI keys are in the UEFI key databases.

Scenario 2: Append keys to DB and DBX to a signature template

In this scenario, you append DB and DBX keys to the default values available in a Trusted Launch VM.

1. Create certificates and/or keys for each or DB and DBX. Refer to the Useful links section for detailed instructions for various operating systems.
2. Create a new image version. Use the following *securityProfile* ARM template block with the image version creation Generic ARM template. Replace `Base64 formatted certificate` and `Base64 formatted sha256 hash` JSON values with the real values.

JSON

```
"securityProfile": {
    "uefiSettings": {
        "signatureTemplateNames": [
            "MicrosoftUefiCertificateAuthorityTemplate"
        ],
        "additionalSignatures": {
            "db": [
                {
                    "type": "x509",
                    "value": [
                        "Base64 formatted certificate",
                        "Base64 formatted certificate"
                    ]
                }
            ],
            "dbx": [
                {
```

```
                    "type": "sha256",
                    "value": [
                        "Base64 formatted sha256 hash",
                        "Base64 formatted sha256 hash"
                    ]
                }
            ]
        }
    }
}
```

3. Create a Trusted Launch VM using Azure portal, PowerShell, CLI, or REST API with the same source image as the one you created using Azure compute gallery resources.
4. Once the Trusted Launch VM is in the *running* state, verify that your custom UEFI keys are in the UEFI key databases.

Scenario 3: Apply multiple signature templates

In this scenario, you replace default signature templates and inject multiple signature templates (UEFI, Windows) into a Trusted Launch VM.

1. Create a new image version. Use the following *securityProfile* ARM template block with the image version creation Generic ARM template.

JSON

```
"securityProfile": {
    "uefiSettings": {
        "signatureTemplateNames": [
            "MicrosoftUefiCertificateAuthorityTemplate",
            "MicrosoftWindowsTemplate"
        ]
    }
}
```

2. Create a Trusted Launch VM using Azure portal, PowerShell, CLI, or REST API with the same source image as the one you created using Azure compute gallery resources.
3. Once the Trusted Launch VM is in the *running* state, verify that your custom UEFI keys are in the UEFI key databases.

Verify keys in UEFI databases

To verify the keys stored in UEFI databases, use the following commands in the command line (SSH) on the desired VM.

```Bash
// For key(s) added to db
$ mokutil --db

// For keys(s) added to dbx
$ mokutil --dbx
```

Verify that the displayed keys are the ones supplied with the image version.

Considerations

- Clouds and regions are supported
- For public cloud, all regions are supported except the following regions: Austria East, Belgium Central, Chile Central, Indonesia Central, Israel Northwest, Jio India Central, Jio India West, Malaysia South, Malaysia West, Mexico Central, Qatar Central, South Central US 2, Southeast US, Southeast US 3, Switzerland West, Taiwan North, Taiwan Northwest, UK West, and West India
- USGov and China clouds are currently not supported

Useful links

- How to convert Base64 certificates: Base64 certificate encoding
- How to convert X.509 certificate to Base64: X.509 Certificate Public Key in Base64
- How to sign things for secure boot: Ubuntu, Redhat, Debian, Oracle

---

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Enable just-in-time access

Defender for Servers in Microsoft Defender for Cloud provides a just-in-time machine access feature.

You can use Microsoft Defender for Cloud's just-in-time access to protect your Azure VMs from unauthorized network access. Many times firewalls contain allow rules that leave your VMs vulnerable to attack. JIT lets you allow access to your VMs only when the access is needed, on the ports needed, and for the period of time needed.

In this article, you learn how to set up and use just-in-time access, including how to:

- Enable just-in-time on VMs from the Azure portal or programmatically
- Request access to a VM that has just-in-time access enabled from the Azure portal or programmatically
- Audit just-in-time access activity to make sure your VMs are secured appropriately

Prerequisites

- Microsoft Defender for Servers Plan 2 must be enabled on the subscription.
- Supported VMs: VMs deployed through Azure Resource Manager, VMs protected by Azure Firewalls on the same VNET as the VM, AWS EC2 instances (Preview).
- Unsupported VMs: VMs deployed with classic deployment models, VMs protected by Azure Firewalls controlled by Azure Firewall Manager.
- To set up just-in-time access on your AWS VMs, you need to connect your AWS account to Microsoft Defender for Cloud.
- To create a JIT policy, the policy name, together with the targeted VM name, must not exceed a total of 56 characters.
- You need **Reader** and **SecurityReader** permissions, or a custom role can view the JIT status and parameters.
- For a custom role, assign the permissions summarized in the table. To create a least-privileged role for users that only need to request JIT access to a VM, use the Set-JitLeastPrivilegedRole script.

Expand table

User action	Permissions to set
Configure or edit a JIT policy for a VM	*Assign these actions to the role:* • On the scope of a subscription (or resource group when using API or PowerShell only) that is associated with the VM: `Microsoft.Security/locations/jitNetworkAccessPolicies/write` • On the scope of a subscription (or resource group when using API or PowerShell only) of VM: `Microsoft.Compute/virtualMachines/write`
Request JIT access to a VM	*Assign these actions to the user:* • `Microsoft.Security/locations/jitNetworkAccessPolicies/initiate/action` • `Microsoft.Security/locations/jitNetworkAccessPolicies/*/read` • `Microsoft.Compute/virtualMachines/read` • `Microsoft.Network/networkInterfaces/*/read` • `Microsoft.Network/publicIPAddresses/read`
Read JIT policies	*Assign these actions to the user:* • `Microsoft.Security/locations/jitNetworkAccessPolicies/read` • `Microsoft.Security/locations/jitNetworkAccessPolicies/initiate/action` • `Microsoft.Security/policies/read` • `Microsoft.Security/pricings/read` • `Microsoft.Compute/virtualMachines/read` • `Microsoft.Network/*/read`

> ⓘ **Note**
>
> Only the `Microsoft.Security` permissions are relevant for AWS. To create a least-privileged role for users that only need to request JIT access to a VM, use the Set-JitLeastPrivilegedRole script.

Work with JIT VM access using Microsoft Defender for Cloud

You can use Defender for Cloud or you can programmatically enable JIT VM access with your own custom options, or you can enable JIT with default, hard-coded parameters from Azure virtual machines.

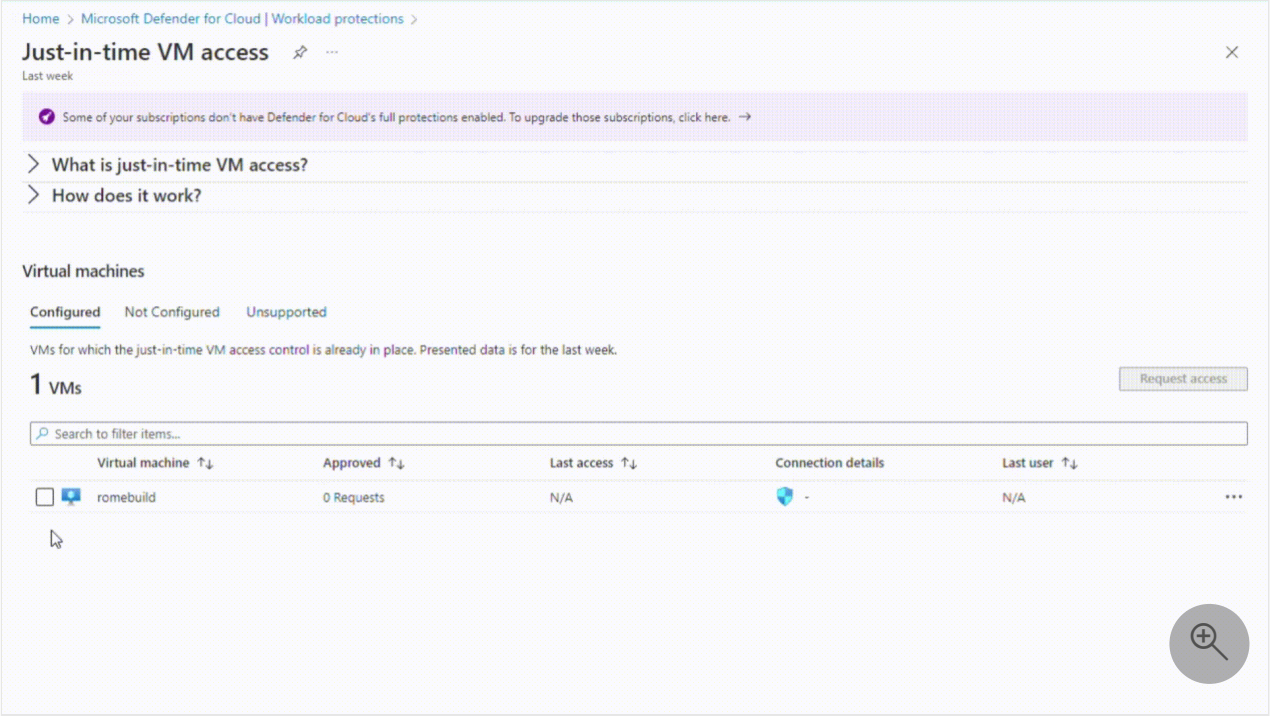
Just-in-time VM access shows your VMs grouped into:

- **Configured** - VMs configured to support just-in-time VM access, and shows:
  - the number of approved JIT requests in the last seven days
  - the last access date and time
  - the connection details configured
  - the last user
- **Not configured** - VMs without JIT enabled, but that can support JIT. We recommend that you enable JIT for these VMs.
- **Unsupported** - VMs that don't support JIT because:
  - Missing network security group (NSG) or Azure Firewall - JIT requires an NSG to be configured or a Firewall configuration (or both)
  - Classic VM - JIT supports VMs that are deployed through Azure Resource Manager.
  - Other - The JIT solution is disabled in the security policy of the subscription or the resource group.

Enable JIT on your VMs from Microsoft Defender for Cloud

From Defender for Cloud, you can enable and configure the JIT VM access.

1. Open **Workload protections** and, in the advanced protections, select **Just-in-time VM access**.



2. In the **Not configured** virtual machines tab, mark the VMs to protect with JIT and select **Enable JIT on VMs**.

   The JIT VM access page opens listing the ports that Defender for Cloud recommends protecting:

- 22 - SSH
- 3389 - RDP
- 5985 - WinRM
- 5986 - WinRM

To customize the JIT access:

a. Select **Add**.

b. Select one of the ports in the list to edit it or enter other ports. For each port, you can set the:

- **Protocol**
- **Allowed source IPs**
- **Maximum request time**

c. Select **OK**.

3. To save the port configuration, select **Save**.

Edit the JIT configuration on a JIT-enabled VM using Defender for Cloud

You can modify a VM's just-in-time configuration by adding and configuring a new port to protect for that VM, or by changing any other setting related to an already protected port.

To edit the existing JIT rules for a VM:

1. Open **Workload protections** and, in the advanced protections, select **Just-in-time VM access**.
2. In the **Configured** virtual machines tab, right-click on a VM and select **Edit**.
3. In the **JIT VM access configuration**, you can either edit the list of port or select Add a new custom port.
4. When you finish editing the ports, select **Save**.

Request access to a JIT-enabled VM from Microsoft Defender for Cloud

When a VM has JIT enabled, you have to request access to connect to it. You can request access in any of the supported ways, regardless of how you enabled JIT.

1. From the **Just-in-time VM access** page, select the **Configured** tab.
2. Select the VMs you want to access.
    - The icon in the **Connection Details** column indicates whether JIT is enabled on the network security group or firewall. If it's enabled on both, only the firewall icon appears.
    - The **Connection Details** column shows the user and ports that can access the VM.
3. Select **Request access**. The **Request access** window opens.
4. Under **Request access**, select the ports that you want to open for each VM, the source IP addresses that you want the port opened on, and the time window to open the ports.
5. Select **Open ports**.

> **Note**
>
> If a user who is requesting access is behind a proxy, you can enter the IP address range of the proxy.

Other ways to work with JIT VM access

Azure virtual machines

Enable JIT on your VMs from Azure virtual machines

You can enable JIT on a VM from the Azure virtual machines pages of the Azure portal.

> **Tip**
>
> If a VM already has JIT enabled, the VM configuration page shows that JIT is enabled. You can use the link to open the JIT VM access page in Defender for Cloud to view and change the settings.

1. From the Azure portal, search for and select **Virtual machines**.
2. Select the virtual machine you want to protect with JIT.
3. In the menu, select **Configuration**.

4. Under **Just-in-time access**, select **Enable just-in-time**.

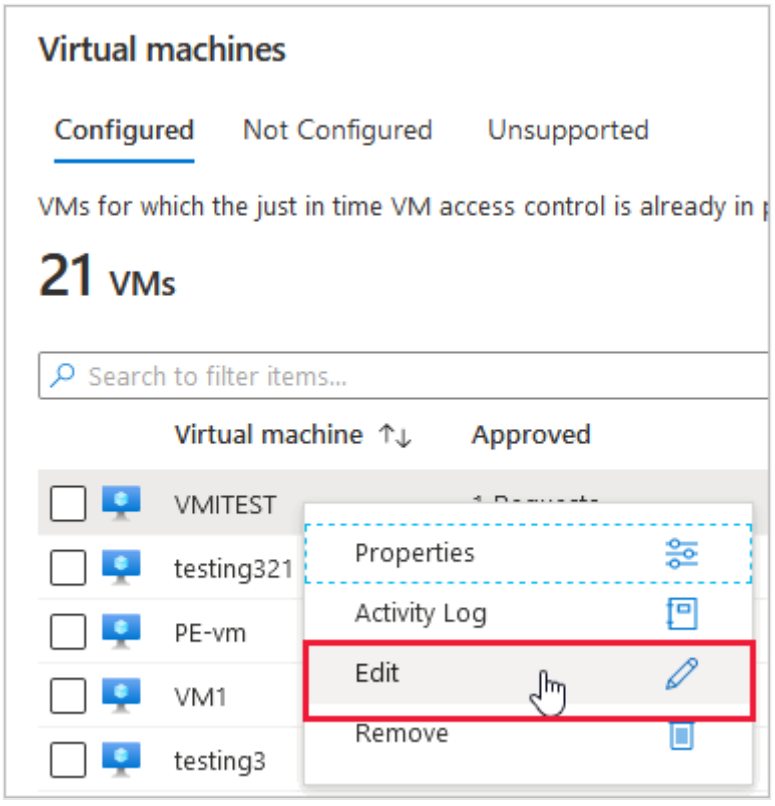
   By default, just-in-time access for the VM uses these settings:

   - Windows machines:
     - RDP port: 3389
     - Maximum allowed access: 3 hours
     - Allowed source IP addresses: Any
   - Linux machines:
     - SSH port: 22
     - Maximum allowed access: 3 hours
     - Allowed source IP addresses: Any

5. To edit any of these values or add more ports to your JIT configuration, use Microsoft Defender for Cloud's just-in-time page:

   a. From Defender for Cloud's menu, select **Just-in-time VM access**.

   b. From the **Configured** tab, right-click on the VM to which you want to add a port, and select **Edit**.

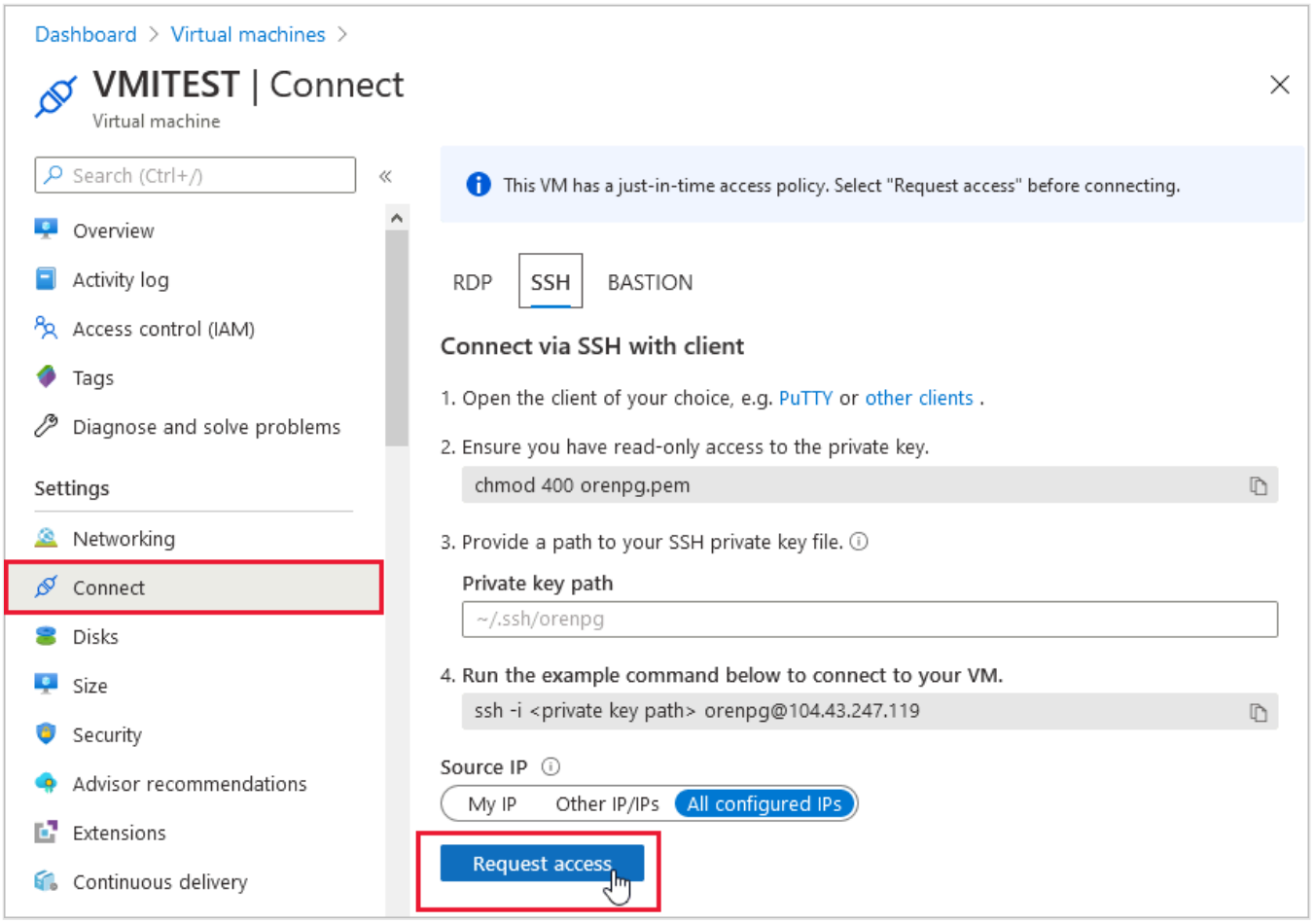


   c. Under **JIT VM access configuration**, you can either edit the existing settings of an already protected port or add a new custom port.

   d. When you've finished editing the ports, select **Save**.

Request access to a JIT-enabled VM from the Azure virtual machine's connect page

When a VM has a JIT enabled, you have to request access to connect to it. You can request access in any of the supported ways, regardless of how you enabled JIT.



To request access from Azure virtual machines:

1. In the Azure portal, open the virtual machines pages.
2. Select the VM to which you want to connect, and open the **Connect** page.

   Azure checks to see if JIT is enabled on that VM.

   - If JIT isn't enabled for the VM, you're prompted to enable it.
   - If JIT is enabled, select **Request access** to pass an access request with the requesting IP, time range, and ports that were configured for that VM.

> ⓘ **Note**
>
> After a request is approved for a VM protected by Azure Firewall, Defender for Cloud provides the user with the proper connection details (the port mapping from the DNAT table) to use to connect to the VM.

PowerShell

Enable JIT on your VMs using PowerShell

To enable just-in-time VM access from PowerShell, use the official Microsoft Defender for Cloud PowerShell cmdlet `Set-AzJitNetworkAccessPolicy`.

Example - Enable just-in-time VM access on a specific VM with the following rules:

- Close ports 22 and 3389
- Set a maximum time window of 3 hours for each so they can be opened per approved request
- Allow the user who is requesting access to control the source IP addresses
- Allow the user who is requesting access to establish a successful session upon an approved just-in-time access request

The following PowerShell commands create this JIT configuration:

1. Assign a variable that holds the just-in-time VM access rules for a VM:

   Azure PowerShell

   ```
   $JitPolicy = (@{

   id="/subscriptions/SUBSCRIPTIONID/resourceGroups/RESOURCEGROUP/providers/Microsoft.Compute/virtualMachines/VMNAME";
       ports=(@{
           number=22;
           protocol="*";
           allowedSourceAddressPrefix=@("*");
           maxRequestAccessDuration="PT3H"},
           @{
           number=3389;
           protocol="*";
           allowedSourceAddressPrefix=@("*");
           maxRequestAccessDuration="PT3H"})})
   ```

2. Insert the VM just-in-time VM access rules into an array:

   Azure PowerShell

   ```
   $JitPolicyArr=@($JitPolicy)
   ```

3. Configure the just-in-time VM access rules on the selected VM:

   Azure PowerShell

```
Set-AzJitNetworkAccessPolicy -Kind "Basic" -Location "LOCATION" -Name "default"
-ResourceGroupName "RESOURCEGROUP" -VirtualMachine $JitPolicyArr
```

Use the -Name parameter to specify a VM. For example, to establish the JIT configuration for two different VMs, VM1 and VM2, use: `Set-AzJitNetworkAccessPolicy -Name VM1` and `Set-AzJitNetworkAccessPolicy -Name VM2`.

Request access to a JIT-enabled VM using PowerShell

In the following example, you can see a just-in-time VM access request to a specific VM for port 22, for a specific IP address, and for a specific amount of time:

Run the following commands in PowerShell:

1. Configure the VM request access properties:

Azure PowerShell

```
$JitPolicyVm1 = (@{

id="/subscriptions/SUBSCRIPTIONID/resourceGroups/RESOURCEGROUP/providers/Micros
oft.Compute/virtualMachines/VMNAME";
    ports=(@{
        number=22;
        endTimeUtc="2020-07-15T17:00:00.3658798Z";
        allowedSourceAddressPrefix=@("IPV4ADDRESS")})})
```

2. Insert the VM access request parameters in an array:

Azure PowerShell

```
$JitPolicyArr=@($JitPolicyVm1)
```

3. Send the request access (use the resource ID from step 1)

Azure PowerShell

```
Start-AzJitNetworkAccessPolicy -ResourceId
"/subscriptions/SUBSCRIPTIONID/resourceGroups/RESOURCEGROUP/providers/Microsoft
.Security/locations/LOCATION/jitNetworkAccessPolicies/default" -VirtualMachine
$JitPolicyArr
```

Learn more in the PowerShell cmdlet documentation.

REST API

Enable JIT on your VMs using the REST API

The just-in-time VM access feature can be used via the Microsoft Defender for Cloud API. Use this API to get information about configured VMs, add new ones, request access to a VM, and more.

Learn more at JIT network access policies.

Request access to a JIT-enabled VM using the REST API

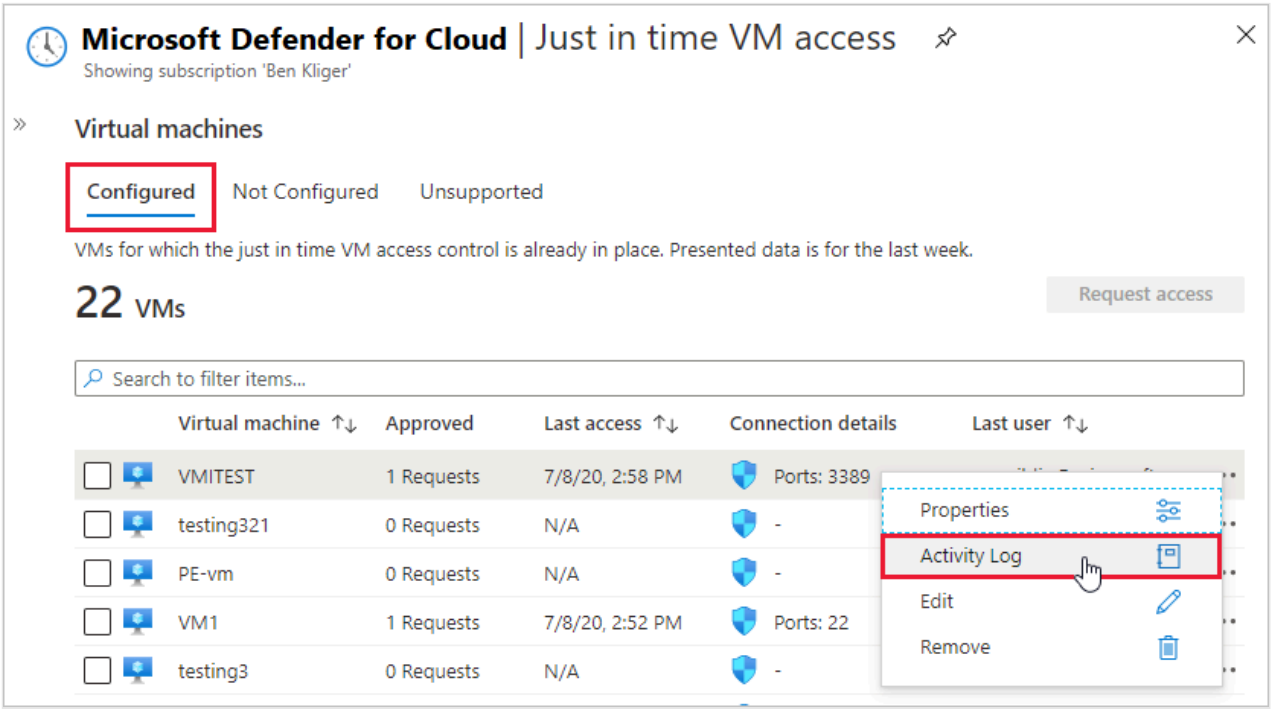
The just-in-time VM access feature can be used via the Microsoft Defender for Cloud API. Use this API to get information about configured VMs, add new ones, request access to a VM, and more.

Learn more at JIT network access policies.

Audit JIT access activity in Defender for Cloud

You can gain insights into VM activities using log search. To view the logs:

1. From **Just-in-time VM access**, select the **Configured** tab.
2. For the VM that you want to audit, open the ellipsis menu at the end of the row.
3. Select **Activity Log** from the menu.



The activity log provides a filtered view of previous operations for that VM along with time, date, and subscription.

4. To download the log information, select **Download as CSV**.

Next step

- Understand just-in-time VM access management

Last updated on 10/28/2025

Secure and use policies on virtual machines in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

It's important to keep your virtual machine (VM) secure for the applications that you run. Securing your VMs can include one or more Azure services and features that cover secure access to your VMs and secure storage of your data. This article provides information that enables you to keep your VM and applications secure.

Antimalware

The modern threat landscape for cloud environments is dynamic, increasing the pressure to maintain effective protection in order to meet compliance and security requirements. Microsoft Antimalware for Azure is a free real-time protection capability that helps identify and remove viruses, spyware, and other malicious software. Alerts can be configured to notify you when known malicious or unwanted software attempts to install itself or run on your VM. It is not supported on VMs running Linux or Windows Server 2008.

Microsoft Defender for Cloud

Microsoft Defender for Cloud helps you prevent, detect, and respond to threats to your VMs. Defender for Cloud provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Defender for Cloud's just-in-time access can be applied across your VM deployment to lock down inbound traffic to your Azure VMs, reducing exposure to attacks while providing easy access to connect to VMs when needed. When just-in-time is enabled and a user requests access to a VM, Defender for Cloud checks what permissions the user has for the VM. If they have the correct permissions, the request is approved and Defender for Cloud automatically configures the Network Security Groups (NSGs) to allow inbound traffic to the selected ports for a limited amount of time. After the time has expired, Defender for Cloud restores the NSGs to their previous states.

Encryption

Two encryption methods are offered for managed disks. Encryption at the OS-level, which is Azure Disk Encryption, and encryption at the platform-level, which is server-side encryption.

Server-side encryption

Azure managed disks automatically encrypt your data by default when persisting it to the cloud. Server-side encryption protects your data and helps you meet your organizational security and compliance commitments. Data in Azure managed disks is encrypted transparently using 256-bit AES encryption, one of the strongest block ciphers available, and is FIPS 140-2 compliant.

Encryption does not impact the performance of managed disks. There is no additional cost for the encryption.

You can rely on platform-managed keys for the encryption of your managed disk, or you can manage encryption using your own keys. If you choose to manage encryption with your own keys, you can specify a *customer-managed key* to use for encrypting and decrypting all data in managed disks.

To learn more about server-side encryption, refer to either the articles for Windows or Linux.

Azure Disk Encryption

For enhanced Windows VM and Linux VM security and compliance, virtual disks in Azure can be encrypted. Virtual disks on Windows VMs are encrypted at rest using BitLocker. Virtual disks on Linux VMs are encrypted at rest using dm-crypt.

There is no charge for encrypting virtual disks in Azure. Cryptographic keys are stored in Azure Key Vault using software-protection, or you can import or generate your keys in Hardware Security Modules (HSMs) certified to FIPS 140 validated standards. These cryptographic keys are used to encrypt and decrypt virtual disks attached to your VM. You retain control of these cryptographic keys and can audit their use. A Microsoft Entra service principal provides a secure mechanism for issuing these cryptographic keys as VMs are powered on and off.

Key Vault and SSH Keys

Secrets and certificates can be modeled as resources and provided by Key Vault. You can use Azure PowerShell to create key vaults for Windows VMs and the Azure CLI for Linux VMs. You can also create keys for encryption.

Key vault access policies grant permissions to keys, secrets, and certificates separately. For example, you can give a user access to only keys, but no permissions for secrets. However, permissions to access keys or secrets or certificates are at the vault level. In other words, key vault access policy does not support object level permissions.

When you connect to VMs, you should use public-key cryptography to provide a more secure way to sign in to them. This process involves a public and private key exchange using the secure shell (SSH) command to authenticate yourself rather than a username and password. Passwords are vulnerable to brute-force attacks, especially on Internet-facing VMs such as web servers. With a secure shell (SSH) key pair, you can create a Linux VM that uses SSH keys for authentication, eliminating the need for passwords to sign-in. You can also use SSH keys to connect from a Windows VM to a Linux VM.

Managed identities for Azure resources

A common challenge when building cloud applications is how to manage the credentials in your code for authenticating to cloud services. Keeping the credentials secure is an important task. Ideally, the credentials never appear on developer workstations and aren't checked into source control. Azure Key Vault provides a way to securely store credentials, secrets, and other keys, but your code has to authenticate to Key Vault to retrieve them.

The managed identities for Azure resources feature in Microsoft Entra solves this problem. The feature provides Azure services with an automatically managed identity in Microsoft Entra ID. You can use the identity to authenticate to any service that supports Microsoft Entra authentication, including Key Vault, without any credentials in your code. Your code that's running on a VM can request a token from two endpoints that are accessible only from within the VM. For more detailed information about this service, review the managed identities for Azure resources overview page.

Policies

Azure policies can be used to define the desired behavior for your organization's VMs. By using policies, an organization can enforce various conventions and rules throughout the enterprise. Enforcement of the desired behavior can help mitigate risk while contributing to the success of the organization.

Azure role-based access control

Using Azure role-based access control (Azure RBAC), you can segregate duties within your team and grant only the amount of access to users on your VM that they need to perform their jobs. Instead of giving everybody unrestricted permissions on the VM, you can allow only certain actions. You can configure access control for the VM in the Azure portal, using the Azure CLI, orAzure PowerShell.

Next steps

- Walk through the steps to monitor virtual machine security by using Microsoft Defender for Cloud for Linux or Windows.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Policy Regulatory Compliance controls for Azure Virtual Machines

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Regulatory Compliance in Azure Policy provides Microsoft created and managed initiative definitions, known as *built-ins*, for the **compliance domains** and **security controls** related to different compliance standards. This page lists the **compliance domains** and **security controls** for Azure Virtual Machines . You can assign the built-ins for a **security control** individually to help make your Azure resources compliant with the specific standard.

The title of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Policy Version** column to view the source on the Azure Policy GitHub repo.

> **Important**
>
> Each control is associated with one or more **Azure Policy** definitions. These policies might help you **assess compliance** with the control. However, there often isn't a one-to-one or complete match between a control and one or more policies. As such, **Compliant** in Azure Policy refers only to the policies themselves. This doesn't ensure that you're fully compliant with all requirements of a control. In addition, the compliance standard includes controls that aren't addressed by any Azure Policy definitions at this time. Therefore, compliance in Azure Policy is only a partial view of your overall compliance status. The associations between controls and Azure Policy Regulatory Compliance definitions for these compliance standards can change over time.

Australian Government ISM PROTECTED

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - Australian Government ISM PROTECTED. For more information about this compliance standard, see Australian Government ISM PROTECTED.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for Personnel Security - Access to systems and their resources	415	User identification - 415	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Personnel Security -	415	User identification - 415	Add system-assigned managed identity to enable Guest	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access to systems and their resources			Configuration assignments on VMs with a user-assigned identity	
Guidelines for Personnel Security - Access to systems and their resources	415	User identification - 415	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Guidelines for Personnel Security - Access to systems and their resources	415	User identification - 415	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Guidelines for System Hardening - Authentication hardening	421	Single-factor authentication - 421	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for System Hardening - Authentication hardening	421	Single-factor authentication - 421	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for System Hardening - Authentication hardening	421	Single-factor authentication - 421	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Guidelines for System Hardening - Authentication hardening	421	Single-factor authentication - 421	Windows machines should meet requirements for 'Security Settings - Account Policies'	3.0.0
Guidelines for Personnel Security - Access to systems and their resources	445	Privileged access to systems - 445	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	445	Privileged access to systems - 445	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	445	Privileged access to systems - 445	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Guidelines for Personnel Security - Access to systems and their resources	445	Privileged access to systems - 445	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for System Monitoring - Event logging and auditing	582	Events to be logged - 582	Virtual machines should be connected to a specified workspace	1.1.0
Guidelines for System Management - System patching	940	When to patch security vulnerabilities - 940	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Guidelines for System Management - System patching	940	When to patch security vulnerabilities - 940	Vulnerabilities in container security configurations should be remediated	3.0.0
Guidelines for System Management - System patching	940	When to patch security vulnerabilities - 940	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Guidelines for System Management - System patching	940	When to patch security vulnerabilities - 940	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Guidelines for Cryptography - Transport Layer Security	1139	Using Transport Layer Security - 1139	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Cryptography - Transport Layer Security	1139	Using Transport Layer Security - 1139	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for Cryptography - Transport Layer Security	1139	Using Transport Layer Security - 1139	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Guidelines for Cryptography - Transport Layer Security	1139	Using Transport Layer Security - 1139	Windows machines should be configured to use secure communication protocols	4.1.1
Guidelines for System Management - System patching	1144	When to patch security vulnerabilities - 1144	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Guidelines for System Management - System patching	1144	When to patch security vulnerabilities - 1144	Vulnerabilities in container security configurations should be remediated	3.0.0
Guidelines for System Management - System patching	1144	When to patch security vulnerabilities - 1144	Vulnerabilities in security configuration on your machines should be remediated	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for System Management - System patching	1144	When to patch security vulnerabilities - 1144	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Guidelines for Networking - Network design and configuration	1182	Network access controls - 1182	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Guidelines for Networking - Network design and configuration	1182	Network access controls - 1182	Internet-facing virtual machines should be protected with network security groups	3.0.0
Guidelines for Database Systems - Database servers	1277	Communications between database servers and web servers - 1277	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Database Systems - Database servers	1277	Communications between database servers and web servers - 1277	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for Database Systems - Database servers	1277	Communications between database servers and web servers - 1277	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Guidelines for Database Systems - Database servers	1277	Communications between database servers and web servers - 1277	Windows machines should be configured to use secure communication protocols	4.1.1
Guidelines for Gateways - Content filtering	1288	Antivirus scanning - 1288	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Guidelines for Gateways - Content filtering	1288	Antivirus scanning - 1288	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
Guidelines for Gateways - Content filtering	1288	Antivirus scanning - 1288	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Guidelines for System Management - System administration	1386	Restriction of management traffic flows - 1386	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Guidelines for System Hardening - Operating system hardening	1407	Operating system versions - 1407	System updates on virtual machine scale sets should be installed	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for System Hardening - Operating system hardening	1407	Operating system versions - 1407	System updates should be installed on your machines	4.0.0
Guidelines for System Hardening - Operating system hardening	1417	Antivirus software - 1417	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Guidelines for System Hardening - Operating system hardening	1417	Antivirus software - 1417	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
Guidelines for System Hardening - Operating system hardening	1417	Antivirus software - 1417	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Guidelines for System Management - System patching	1472	When to patch security vulnerabilities - 1472	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Guidelines for System Management - System patching	1472	When to patch security vulnerabilities - 1472	Vulnerabilities in container security configurations should be remediated	3.0.0
Guidelines for System Management - System patching	1472	When to patch security vulnerabilities - 1472	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Guidelines for System Management - System patching	1472	When to patch security vulnerabilities - 1472	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Guidelines for System Hardening - Operating system hardening	1490	Application control - 1490	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Guidelines for System Management - System patching	1494	When to patch security vulnerabilities - 1494	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Guidelines for System Management - System patching	1494	When to patch security vulnerabilities - 1494	Vulnerabilities in container security configurations should be remediated	3.0.0
Guidelines for System Management - System patching	1494	When to patch security vulnerabilities - 1494	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Guidelines for System Management - System patching	1494	When to patch security vulnerabilities - 1494	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for System Management - System patching	1495	When to patch security vulnerabilities - 1495	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Guidelines for System Management - System patching	1495	When to patch security vulnerabilities - 1495	Vulnerabilities in container security configurations should be remediated	3.0.0
Guidelines for System Management - System patching	1495	When to patch security vulnerabilities - 1495	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Guidelines for System Management - System patching	1495	When to patch security vulnerabilities - 1495	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Guidelines for System Management - System patching	1496	When to patch security vulnerabilities - 1496	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Guidelines for System Management - System patching	1496	When to patch security vulnerabilities - 1496	Vulnerabilities in container security configurations should be remediated	3.0.0
Guidelines for System Management - System patching	1496	When to patch security vulnerabilities - 1496	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Guidelines for System Management - System patching	1496	When to patch security vulnerabilities - 1496	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Guidelines for Personnel Security - Access to systems and their resources	1503	Standard access to systems - 1503	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	1503	Standard access to systems - 1503	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	1503	Standard access to systems - 1503	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Guidelines for Personnel Security - Access to systems and their resources	1503	Standard access to systems - 1503	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for Personnel Security - Access to systems and their resources	1507	Privileged access to systems - 1507	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	1507	Privileged access to systems - 1507	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	1507	Privileged access to systems - 1507	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Guidelines for Personnel Security - Access to systems and their resources	1507	Privileged access to systems - 1507	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Guidelines for Personnel Security - Access to systems and their resources	1508	Privileged access to systems - 1508	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	1508	Privileged access to systems - 1508	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for Personnel Security - Access to systems and their resources	1508	Privileged access to systems - 1508	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Guidelines for Personnel Security - Access to systems and their resources	1508	Privileged access to systems - 1508	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Guidelines for Personnel Security - Access to systems and their resources	1508	Privileged access to systems - 1508	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Guidelines for System Management - Data backup and restoration	1511	Performing backups - 1511	Audit virtual machines without disaster recovery configured	1.0.0
Guidelines for System Hardening - Authentication hardening	1546	Authenticating to systems - 1546	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Guidelines for System Hardening - Authentication hardening	1546	Authenticating to systems - 1546	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Guidelines for System Hardening - Authentication hardening	1546	Authenticating to systems - 1546	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Guidelines for System Hardening - Authentication hardening	1546	Authenticating to systems - 1546	Audit Linux machines that have accounts without passwords	3.1.0
Guidelines for System Hardening - Authentication hardening	1546	Authenticating to systems - 1546	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0

Canada Federal PBMM

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - Canada Federal PBMM. For more information about this compliance standard, see Canada Federal PBMM.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-5	Separation of Duties	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-5	Separation of Duties	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-5	Separation of Duties	Audit Windows machines missing any of specified members in the Administrators group	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-5	Separation of Duties	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Access Control	AC-5	Separation of Duties	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-6	Least Privilege	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-6	Least Privilege	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-6	Least Privilege	Audit Windows machines missing any of specified members in the Administrators group	2.0.0
Access Control	AC-6	Least Privilege	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Access Control	AC-6	Least Privilege	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17(1)	Remote Access | Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17(1)	Remote Access | Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17(1)	Remote Access | Automated Monitoring / Control	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17(1)	Remote Access | Automated Monitoring / Control	Deploy the Linux Guest Configuration extension to	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			enable Guest Configuration assignments on Linux VMs	
Audit and Accountability	AU-3	Content of Audit Records	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Audit and Accountability	AU-3	Content of Audit Records	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Audit and Accountability	AU-3	Content of Audit Records	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	AU-12	Audit Generation	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Audit and Accountability	AU-12	Audit Generation	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Audit and Accountability	AU-12	Audit Generation	Virtual machines should be connected to a specified workspace	1.1.0
Configuration Management	CM-7(5)	Least Functionality | Authorized Software / Whitelisting	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-11	User-Installed Software	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Contingency Planning	CP-7	Alternative Processing Site	Audit virtual machines without disaster recovery configured	1.0.0
Identification and Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification and Authentication	IA-5	Authenticator Management	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification and Authentication	IA-5	Authenticator Management	Audit Linux machines that have accounts without passwords	3.1.0
Identification and Authentication	IA-5	Authenticator Management	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification and Authentication	IA-5(1)	Authenticator Management | Password-Based Authentication	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Risk Assessment	RA-5	Vulnerability Scanning	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Communications Protection	SC-7	Boundary Protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	SC-7	Boundary Protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	SC-7(3)	Boundary Protection | Access Points	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	SC-7(4)	Boundary Protection | External Telecommunications Services	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	SC-8(1)	Transmission Confidentiality and Integrity | Cryptographic or Alternate Physical Protection	Windows machines should be configured to use secure communication protocols	4.1.1
System and Information Integrity	SI-2	Flaw Remediation	System updates on virtual machine scale sets should be installed	3.0.0
System and Information Integrity	SI-2	Flaw Remediation	System updates should be installed on your machines	4.0.0
System and Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
System and Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your virtual	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			machine scale sets should be remediated	
System and Information Integrity	SI-3	Malicious Code Protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	SI-3	Malicious Code Protection	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	SI-3(1)	Malicious Code Protection | Central Management	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	SI-3(1)	Malicious Code Protection | Central Management	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	SI-4	Information System Monitoring	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
System and Information Integrity	SI-4	Information System Monitoring	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
System and Information Integrity	SI-4	Information System Monitoring	Virtual machines should be connected to a specified workspace	1.1.0

CIS Microsoft Azure Foundations Benchmark 1.1.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - CIS Microsoft Azure Foundations Benchmark 1.1.0. For more information about this compliance standard, see CIS Microsoft Azure Foundations Benchmark.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
2 Security Center	2.10	Ensure ASC Default policy setting "Monitor Vulnerability Assessment" is not "Disabled"	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
2 Security Center	2.12	Ensure ASC Default policy setting "Monitor JIT Network Access" is not "Disabled"	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
2 Security Center	2.13	Ensure ASC Default policy setting "Monitor Adaptive Application Whitelisting" is not "Disabled"	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
2 Security Center	2.3	Ensure ASC Default policy setting "Monitor System Updates" is not "Disabled"	System updates should be installed on your machines	4.0.0
2 Security Center	2.4	Ensure ASC Default policy setting "Monitor OS Vulnerabilities" is not "Disabled"	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
2 Security Center	2.5	Ensure ASC Default policy setting "Monitor Endpoint Protection" is not "Disabled"	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
2 Security Center	2.7	Ensure ASC Default policy setting "Monitor Network Security Groups" is not "Disabled"	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
2 Security Center	2.9	Ensure ASC Default policy setting "Enable Next Generation Firewall(NGFW) Monitoring" is not "Disabled"	Internet-facing virtual machines should be protected with network security groups	3.0.0
7 Virtual Machines	7.4	Ensure that only approved extensions are installed	Only approved VM extensions should be installed	1.0.0
7 Virtual Machines	7.5	Ensure that the latest OS Patches for all Virtual Machines are applied	System updates should be installed on your machines	4.0.0
7 Virtual Machines	7.6	Ensure that the endpoint protection for all Virtual Machines is installed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

CIS Microsoft Azure Foundations Benchmark 1.3.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - CIS Microsoft Azure Foundations Benchmark 1.3.0. For more information about this compliance standard, see CIS Microsoft Azure Foundations Benchmark.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
7 Virtual Machines	7.1	Ensure Virtual Machines are utilizing Managed Disks	Audit VMs that do not use managed disks	1.0.0
7 Virtual Machines	7.4	Ensure that only approved extensions are installed	Only approved VM extensions should be installed	1.0.0
7 Virtual Machines	7.5	Ensure that the latest OS Patches for all Virtual Machines are applied	System updates should be installed on your machines	4.0.0
7 Virtual Machines	7.6	Ensure that the endpoint protection for all Virtual Machines is installed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

CIS Microsoft Azure Foundations Benchmark 1.4.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for CIS v1.4.0. For more information about this compliance standard, see CIS Microsoft Azure Foundations Benchmark.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
7 Virtual Machines	7.1	Ensure Virtual Machines are utilizing Managed Disks	Audit VMs that do not use managed disks	1.0.0
7 Virtual Machines	7.4	Ensure that Only Approved Extensions Are Installed	Only approved VM extensions should be installed	1.0.0
7 Virtual Machines	7.5	Ensure that the latest OS Patches for all Virtual Machines are applied	System updates should be installed on your machines	4.0.0
7 Virtual Machines	7.6	Ensure that the endpoint protection for all Virtual Machines is installed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

CIS Microsoft Azure Foundations Benchmark 2.0.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for CIS v2.0.0. For more information about this compliance standard, see CIS Microsoft Azure Foundations Benchmark.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
2.1	2.1.13	Ensure that Microsoft Defender Recommendation for 'Apply system updates' status is 'Completed'	Machines should be configured to periodically check for missing system updates	3.7.0
6	6.1	Ensure that RDP access from the Internet is evaluated and restricted	Management ports should be closed on your virtual machines	3.0.0
6	6.2	Ensure that SSH access from the Internet is evaluated and restricted	Management ports should be closed on your virtual machines	3.0.0
7	7.2	Ensure Virtual Machines are utilizing Managed Disks	Audit VMs that do not use managed disks	1.0.0
7	7.4	Ensure that 'Unattached disks' are encrypted with 'Customer Managed Key' (CMK)	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
7	7.5	Ensure that Only Approved Extensions Are Installed	Only approved VM extensions should be installed	1.0.0
7	7.6	Ensure that Endpoint Protection for all Virtual Machines is installed	Endpoint protection should be installed on your machines	1.0.0

CMMC Level 3

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - CMMC Level 3. For more information about this compliance standard, see Cybersecurity Maturity Model Certification (CMMC).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Windows machines should meet requirements for 'Security Options - Network Access'	3.0.0
Access Control	AC.1.001	Limit information system access to authorized users, processes acting on behalf of authorized users, and devices (including other information systems).	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Windows machines should be configured to use secure communication protocols	4.1.1
Access Control	AC.1.002	Limit information system access to the types of transactions and functions that authorized users are permitted to execute.	Windows machines should meet requirements for 'Security Options - Network Access'	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC.1.003	Verify and control/limit connections to and use of external information systems.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC.1.003	Verify and control/limit connections to and use of external information systems.	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC.2.007	Employ the principle of least privilege, including for specific security functions and privileged accounts.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC.2.008	Use non-privileged accounts or roles when accessing nonsecurity functions.	Windows machines should meet requirements for 'Security Options - User Account Control'	3.0.0
Access Control	AC.2.008	Use non-privileged accounts or roles when accessing nonsecurity functions.	Windows machines should meet requirements for 'User Rights Assignment'	3.0.0
Access Control	AC.2.013	Monitor and control remote access sessions.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC.2.013	Monitor and control remote access sessions.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC.2.013	Monitor and control remote access sessions.	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC.2.013	Monitor and control remote access sessions.	Deploy the Windows Guest Configuration extension to enable Guest Configuration	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			assignments on Windows VMs	
Access Control	AC.2.013	Monitor and control remote access sessions.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC.2.013	Monitor and control remote access sessions.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC.2.016	Control the flow of CUI in accordance with approved authorizations.	Windows machines should meet requirements for 'Security Options - Network Access'	3.0.0
Access Control	AC.3.017	Separate the duties of individuals to reduce the risk of malevolent activity without collusion.	Audit Windows machines missing any of specified members in the Administrators group	2.0.0
Access Control	AC.3.017	Separate the duties of individuals to reduce the risk of malevolent activity without collusion.	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Access Control	AC.3.018	Prevent non-privileged users from executing privileged functions and capture the execution of such functions in audit logs.	Windows machines should meet requirements for 'System Audit Policies - Privilege Use'	3.0.0
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			machines with no identities	
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Guest Configuration extension should be installed on your machines	1.0.3
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Windows machines should meet requirements for 'Security Options - User Account Control'	3.0.0
Access Control	AC.3.021	Authorize remote execution of privileged commands and remote access to security-relevant information.	Windows machines should meet requirements for 'User Rights Assignment'	3.0.0
Audit and Accountability	AU.2.041	Ensure that the actions of individual system users can be uniquely traced to those users so they can be held accountable for their actions.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Audit and Accountability	AU.2.041	Ensure that the actions of individual system users can be uniquely traced	Log Analytics extension should be enabled in	2.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		to those users so they can be held accountable for their actions.	virtual machine scale sets for listed virtual machine images	
Audit and Accountability	AU.2.041	Ensure that the actions of individual system users can be uniquely traced to those users so they can be held accountable for their actions.	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Audit and Accountability	AU.2.041	Ensure that the actions of individual system users can be uniquely traced to those users so they can be held accountable for their actions.	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	AU.2.041	Ensure that the actions of individual system users can be uniquely traced to those users so they can be held accountable for their actions.	Virtual machines should have the Log Analytics extension installed	1.0.1
Audit and Accountability	AU.2.042	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Audit and Accountability	AU.2.042	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity.	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Audit and Accountability	AU.2.042	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity.	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Audit and Accountability	AU.2.042	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity.	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	AU.2.042	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity.	Virtual machines should have the Log Analytics extension installed	1.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit and Accountability	AU.3.046	Alert in the event of an audit logging process failure.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Audit and Accountability	AU.3.046	Alert in the event of an audit logging process failure.	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Audit and Accountability	AU.3.046	Alert in the event of an audit logging process failure.	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	AU.3.048	Collect audit information (e.g., logs) into one or more central repositories.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Audit and Accountability	AU.3.048	Collect audit information (e.g., logs) into one or more central repositories.	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Audit and Accountability	AU.3.048	Collect audit information (e.g., logs) into one or more central repositories.	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Audit and Accountability	AU.3.048	Collect audit information (e.g., logs) into one or more central repositories.	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	AU.3.048	Collect audit information (e.g., logs) into one or more central repositories.	Virtual machines should have the Log Analytics extension installed	1.0.1
Security Assessment	CA.2.158	Periodically assess the security controls in organizational systems to determine if the controls are effective in their application.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Security Assessment	CA.2.158	Periodically assess the security controls in organizational systems to determine if the controls are effective in their application.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Security Assessment	CA.2.158	Periodically assess the security controls in organizational systems to	Allowlist rules in your adaptive application	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		determine if the controls are effective in their application.	control policy should be updated	
Security Assessment	CA.2.158	Periodically assess the security controls in organizational systems to determine if the controls are effective in their application.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Security Assessment	CA.2.158	Periodically assess the security controls in organizational systems to determine if the controls are effective in their application.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Security Assessment	CA.3.161	Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Security Assessment	CA.3.161	Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Security Assessment	CA.3.161	Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Security Assessment	CA.3.161	Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Security Assessment	CA.3.161	Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Configuration Management	CM.2.061	Establish and maintain baseline configurations and inventories of organizational systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM.2.061	Establish and maintain baseline configurations and inventories of organizational systems (including hardware, software, firmware, and documentation) throughout the	Linux machines should meet requirements for the Azure compute security baseline	2.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		respective system development life cycles.		
Configuration Management	CM.2.062	Employ the principle of least functionality by configuring organizational systems to provide only essential capabilities.	Windows machines should meet requirements for 'System Audit Policies - Privilege Use'	3.0.0
Configuration Management	CM.2.063	Control and monitor user-installed software.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM.2.063	Control and monitor user-installed software.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM.2.063	Control and monitor user-installed software.	Windows machines should meet requirements for 'Security Options - User Account Control'	3.0.0
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Configuration Management	CM.2.064	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Configuration Management	CM.2.065	Track, review, approve or disapprove, and log changes to organizational systems.	Windows machines should meet requirements for 'System Audit Policies - Policy Change'	3.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Adaptive network hardening recommendations should	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			be applied on internet facing virtual machines	
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Internet-facing virtual machines should be protected with network security groups	3.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Configuration Management	CM.3.068	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Configuration Management	CM.3.069	Apply deny-by-exception (blacklisting) policy to prevent the use of unauthorized software or deny-all, permit-by-exception (whitelisting) policy to allow the execution of authorized software.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Identification and Authentication	IA.1.077	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA.1.077	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification and Authentication	IA.1.077	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification and Authentication	IA.1.077	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	Audit Linux machines that have accounts without passwords	3.1.0
Identification and Authentication	IA.1.077	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	IA.1.077	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Audit Linux machines that have accounts without passwords	3.1.0
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Audit Windows machines that do not restrict the minimum password	2.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			length to specified number of characters	
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	IA.2.078	Enforce a minimum password complexity and change of characters when new passwords are created.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Identification and Authentication	IA.2.079	Prohibit password reuse for a specified number of generations.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA.2.079	Prohibit password reuse for a specified number of generations.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	IA.2.079	Prohibit password reuse for a specified number of generations.	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification and Authentication	IA.2.079	Prohibit password reuse for a specified number of generations.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	IA.2.079	Prohibit password reuse for a specified number of generations.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Identification and Authentication	IA.2.081	Store and transmit only cryptographically-protected passwords.	Add system-assigned managed identity to enable Guest	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			Configuration assignments on virtual machines with no identities	
Identification and Authentication	IA.2.081	Store and transmit only cryptographically-protected passwords.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	IA.2.081	Store and transmit only cryptographically-protected passwords.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification and Authentication	IA.2.081	Store and transmit only cryptographically-protected passwords.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	IA.2.081	Store and transmit only cryptographically-protected passwords.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Identification and Authentication	IA.3.084	Employ replay-resistant authentication mechanisms for network access to privileged and nonprivileged accounts.	Windows machines should be configured to use secure communication protocols	4.1.1
Incident Response	IR.2.093	Detect and report events.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Recovery	RE.2.137	Regularly perform and test data back-ups.	Audit virtual machines without disaster recovery configured	1.0.0
Recovery	RE.2.137	Regularly perform and test data back-ups.	Azure Backup should be enabled for Virtual Machines	3.0.0
Recovery	RE.3.139	Regularly perform complete, comprehensive and resilient data backups as organizationally-defined.	Audit virtual machines without disaster recovery configured	1.0.0
Recovery	RE.3.139	Regularly perform complete, comprehensive and resilient data	Azure Backup should be enabled for Virtual	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		backups as organizationally-defined.	Machines	
Risk Assessment	RM.2.141	Periodically assess the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, and individuals, resulting from the operation of organizational systems and the associated processing, storage, or transmission of CUI.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RM.2.142	Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RM.2.143	Remediate vulnerabilities in accordance with risk assessments.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RM.2.143	Remediate vulnerabilities in accordance with risk assessments.	Vulnerabilities in container security configurations should be remediated	3.0.0
Risk Assessment	RM.2.143	Remediate vulnerabilities in accordance with risk assessments.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	RM.2.143	Remediate vulnerabilities in accordance with risk assessments.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Windows machines should be configured to use secure communication protocols	4.1.1
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Windows machines should meet requirements for 'Security Options - Network Access'	3.0.0
System and Communications Protection	SC.1.175	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
System and Communications Protection	SC.1.176	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Adaptive network hardening recommendations should	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			be applied on internet facing virtual machines	
System and Communications Protection	SC.1.176	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	SC.1.176	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC.2.179	Use encrypted sessions for the management of network devices.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	SC.3.177	Employ FIPS-validated cryptography when used to protect the confidentiality of CUI.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
System and Communications Protection	SC.3.181	Separate user functionality from system management functionality.	Audit Windows machines that have the specified members in the Administrators group	2.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Windows machines should meet requirements for 'Security Options - Network Access'	3.0.0
System and Communications Protection	SC.3.183	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
System and Communications Protection	SC.3.185	Implement cryptographic mechanisms to prevent unauthorized disclosure of CUI during transmission unless otherwise protected by alternative physical safeguards.	Windows machines should be configured to use secure communication protocols	4.1.1
System and Communications Protection	SC.3.190	Protect the authenticity of communications sessions.	Windows machines should be configured to use secure communication protocols	4.1.1
System and Information Integrity	SI.1.210	Identify, report, and correct information and information system flaws in a timely manner.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	SI.1.210	Identify, report, and correct information and information system flaws in a timely manner.	System updates on virtual machine scale sets should be installed	3.0.0
System and Information Integrity	SI.1.210	Identify, report, and correct information and information system flaws in a timely manner.	System updates should be installed on your machines	4.0.0
System and Information Integrity	SI.1.210	Identify, report, and correct information and information system flaws in a timely manner.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
System and Information Integrity	SI.1.210	Identify, report, and correct information and information system flaws in a timely manner.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Information Integrity	SI.1.211	Provide protection from malicious code at appropriate locations within organizational information systems.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	SI.1.211	Provide protection from malicious code at appropriate locations within organizational information systems.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	SI.1.211	Provide protection from malicious code at appropriate locations within organizational information systems.	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
System and Information Integrity	SI.1.211	Provide protection from malicious code at appropriate locations within organizational information systems.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	SI.1.212	Update malicious code protection mechanisms when new releases are available.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	SI.1.213	Perform periodic scans of the information system and real-time scans of files from external sources as files are downloaded, opened, or executed.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	SI.1.213	Perform periodic scans of the information system and real-time scans of files from external sources as files are downloaded, opened, or executed.	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
System and Information Integrity	SI.1.213	Perform periodic scans of the information system and real-time scans of files from external sources as files are downloaded, opened, or executed.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

FedRAMP High

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - FedRAMP High. For more information about this compliance standard, see FedRAMP High.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-2 (12)	Account Monitoring / Atypical Usage	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-3	Access Enforcement	Audit Linux machines that have accounts without passwords	3.1.0
Access Control	AC-3	Access Enforcement	Authentication to Linux machines should require SSH keys	3.2.0
Access Control	AC-3	Access Enforcement	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-3	Access Enforcement	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Access Control	AC-4	Information Flow Enforcement	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Access Control	AC-4	Information Flow Enforcement	Disk access resources should use private link	1.0.0
Access Control	AC-4	Information Flow Enforcement	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-4	Information Flow Enforcement	IP Forwarding on your virtual machine should be disabled	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	Management ports should be closed on your virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17	Remote Access	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17	Remote Access	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-17	Remote Access	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17	Remote Access	Disk access resources should use private link	1.0.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-17 (1)	Automated Monitoring / Control	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Disk access resources should use private link	1.0.0
Audit And Accountability	AU-6	Audit Review, Analysis, And Reporting	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6	Audit Review, Analysis, And Reporting	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (4)	Central Review And Analysis	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (4)	Central Review And Analysis	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (4)	Central Review And Analysis	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-6 (4)	Central Review And Analysis	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit And Accountability	AU-12	Audit Generation	[Preview]: Network traffic data collection agent should be installed	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			on Linux virtual machines	
Audit And Accountability	AU-12	Audit Generation	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-12	Audit Generation	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-12	Audit Generation	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Configuration Management	CM-6	Configuration Settings	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Configuration Management	CM-6	Configuration Settings	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Configuration Management	CM-7	Least Functionality	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7	Least Functionality	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (5)	Authorized Software / Whitelisting	Adaptive application controls for defining safe applications should be	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			enabled on your machines	
Configuration Management	CM-7 (5)	Authorized Software / Whitelisting	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-11	User-Installed Software	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-11	User-Installed Software	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Contingency Planning	CP-7	Alternate Processing Site	Audit virtual machines without disaster recovery configured	1.0.0
Contingency Planning	CP-9	Information System Backup	Azure Backup should be enabled for Virtual Machines	3.0.0
Identification And Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification And Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification And Authentication	IA-5	Authenticator Management	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification And Authentication	IA-5	Authenticator Management	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification And Authentication	IA-5	Authenticator Management	Authentication to Linux machines should require SSH keys	3.2.0
Identification And Authentication	IA-5	Authenticator Management	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification And Authentication	IA-5	Authenticator Management	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Risk Assessment	RA-5	Vulnerability Scanning	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	SQL servers on machines should have vulnerability findings resolved	1.0.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in container security configurations should be remediated	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System And Communications Protection	SC-3	Security Function Isolation	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Communications Protection	SC-3	Security Function Isolation	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Communications Protection	SC-3	Security Function Isolation	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Communications Protection	SC-5	Denial Of Service Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System And Communications Protection	SC-7	Boundary Protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Disk access resources should use private link	1.0.0
System And Communications Protection	SC-7	Boundary Protection	Internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7	Boundary Protection	IP Forwarding on your virtual machine should be disabled	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Communications Protection	SC-7	Boundary Protection	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Management ports should be closed on your virtual machines	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Disk access resources should use private link	1.0.0
System And Communications Protection	SC-7 (3)	Access Points	Internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Management ports should be closed on your virtual machines	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-8	Transmission Confidentiality And Integrity	Windows machines should be configured to use secure communication protocols	4.1.1
System And Communications Protection	SC-8 (1)	Cryptographic Or Alternate Physical Protection	Windows machines should be configured to use secure communication protocols	4.1.1
System And Communications	SC-12	Cryptographic Key Establishment And	Managed disks should be double encrypted with both platform-	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection		Management	managed and customer-managed keys	
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	OS and data disks should be encrypted with a customer-managed key	3.0.0
System And Communications Protection	SC-28	Protection Of Information At Rest	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System And Communications Protection	SC-28 (1)	Cryptographic Protection	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System And Information Integrity	SI-2	Flaw Remediation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System And Information Integrity	SI-2	Flaw Remediation	System updates on virtual machine scale sets should be installed	3.0.0
System And Information Integrity	SI-2	Flaw Remediation	System updates should be installed on your machines	4.0.0
System And Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
System And Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Information Integrity	SI-3 (1)	Central Management	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Information Integrity	SI-3 (1)	Central Management	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Information Integrity	SI-3 (1)	Central Management	Windows Defender Exploit Guard should be enabled on your machines	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Information Integrity	SI-4	Information System Monitoring	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
System And Information Integrity	SI-4	Information System Monitoring	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
System And Information Integrity	SI-4	Information System Monitoring	Guest Configuration extension should be installed on your machines	1.0.3
System And Information Integrity	SI-4	Information System Monitoring	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
System And Information Integrity	SI-16	Memory Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0

FedRAMP Moderate

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - FedRAMP Moderate. For more information about this compliance standard, see FedRAMP Moderate.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-2 (12)	Account Monitoring / Atypical Usage	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-3	Access Enforcement	Audit Linux machines that have accounts without passwords	3.1.0
Access Control	AC-3	Access Enforcement	Authentication to Linux machines should require SSH keys	3.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-3	Access Enforcement	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-3	Access Enforcement	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Access Control	AC-4	Information Flow Enforcement	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Access Control	AC-4	Information Flow Enforcement	Disk access resources should use private link	1.0.0
Access Control	AC-4	Information Flow Enforcement	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-4	Information Flow Enforcement	IP Forwarding on your virtual machine should be disabled	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports should be closed on your virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17	Remote Access	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17	Remote Access	Deploy the Linux Guest Configuration extension to enable Guest	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			Configuration assignments on Linux VMs	
Access Control	AC-17	Remote Access	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17	Remote Access	Disk access resources should use private link	1.0.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Disk access resources should use private link	1.0.0
Audit And Accountability	AU-6	Audit Review, Analysis, And Reporting	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6	Audit Review, Analysis, And Reporting	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-12	Audit Generation	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-12	Audit Generation	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit And Accountability	AU-12	Audit Generation	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-12	Audit Generation	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Configuration Management	CM-6	Configuration Settings	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Configuration Management	CM-6	Configuration Settings	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Configuration Management	CM-7	Least Functionality	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7	Least Functionality	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (5)	Authorized Software / Whitelisting	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (5)	Authorized Software / Whitelisting	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-11	User-Installed Software	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-11	User-Installed Software	Allowlist rules in your adaptive application control policy should be	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			updated	
Contingency Planning	CP-7	Alternate Processing Site	Audit virtual machines without disaster recovery configured	1.0.0
Contingency Planning	CP-9	Information System Backup	Azure Backup should be enabled for Virtual Machines	3.0.0
Identification And Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification And Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification And Authentication	IA-5	Authenticator Management	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification And Authentication	IA-5	Authenticator Management	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification And Authentication	IA-5	Authenticator Management	Authentication to Linux machines should require SSH keys	3.2.0
Identification And Authentication	IA-5	Authenticator Management	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification And Authentication	IA-5	Authenticator Management	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Risk Assessment	RA-5	Vulnerability Scanning	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	SQL servers on machines should have vulnerability findings resolved	1.0.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in container security configurations should be remediated	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Communications Protection	SC-5	Denial Of Service Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System And Communications Protection	SC-7	Boundary Protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Disk access resources should use private link	1.0.0
System And Communications Protection	SC-7	Boundary Protection	Internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7	Boundary Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Management ports should be closed on your virtual machines	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Disk access resources should use private link	1.0.0
System And Communications Protection	SC-7 (3)	Access Points	Internet-facing virtual machines should be protected with network security groups	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Communications Protection	SC-7 (3)	Access Points	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Management ports should be closed on your virtual machines	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-8	Transmission Confidentiality And Integrity	Windows machines should be configured to use secure communication protocols	4.1.1
System And Communications Protection	SC-8 (1)	Cryptographic Or Alternate Physical Protection	Windows machines should be configured to use secure communication protocols	4.1.1
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	OS and data disks should be encrypted with a customer-managed key	3.0.0
System And Communications Protection	SC-28	Protection Of Information At Rest	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System And Communications Protection	SC-28 (1)	Cryptographic Protection	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System And Information Integrity	SI-2	Flaw Remediation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System And Information Integrity	SI-2	Flaw Remediation	System updates on virtual machine scale sets should be installed	3.0.0
System And Information Integrity	SI-2	Flaw Remediation	System updates should be installed on your machines	4.0.0
System And Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your machines	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			should be remediated	
System And Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Information Integrity	SI-3 (1)	Central Management	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Information Integrity	SI-3 (1)	Central Management	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Information Integrity	SI-3 (1)	Central Management	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Information Integrity	SI-4	Information System Monitoring	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
System And Information Integrity	SI-4	Information System Monitoring	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
System And Information Integrity	SI-4	Information System Monitoring	Guest Configuration extension should be installed on your machines	1.0.3
System And Information Integrity	SI-4	Information System Monitoring	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
System And Information Integrity	SI-16	Memory Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0

HIPAA HITRUST 9.2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - HIPAA HITRUST 9.2. For more information about this compliance standard, see HIPAA HITRUST 9.2 .

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
User Identification and Authentication	11210.01q2Organizational.10 - 01.q	Electronic signatures and handwritten signatures executed to electronic records shall be linked to their respective electronic records.	Audit Windows machines that have the specified members in the Administrators group	2.0.0
User Identification and Authentication	11211.01q2Organizational.11 - 01.q	Signed electronic records shall contain information associated with the signing in human-readable format.	Audit Windows machines missing any of specified members in the Administrators group	2.0.0
02 Endpoint Protection	0201.09j1Organizational.124-09.j	0201.09j1Organizational.124-09.j 09.04 Protection Against Malicious and Mobile Code	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
02 Endpoint Protection	0201.09j1Organizational.124-09.j	0201.09j1Organizational.124-09.j 09.04 Protection Against Malicious and Mobile Code	Deploy default Microsoft IaaSAntimalware extension for Windows Server	1.1.0
02 Endpoint Protection	0201.09j1Organizational.124-09.j	0201.09j1Organizational.124-09.j 09.04 Protection Against Malicious and Mobile Code	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
02 Endpoint Protection	0201.09j1Organizational.124-09.j	0201.09j1Organizational.124-09.j 09.04 Protection Against Malicious and Mobile Code	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
02 Endpoint Protection	0201.09j1Organizational.124-09.j	0201.09j1Organizational.124-09.j 09.04 Protection Against Malicious and Mobile Code	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
02 Endpoint Protection	0201.09j1Organizational.124-09.j	0201.09j1Organizational.124-09.j 09.04 Protection Against	System updates should be	4.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		Malicious and Mobile Code	installed on your machines	
06 Configuration Management	0605.10h1System.12-10.h	0605.10h1System.12-10.h 10.04 Security of System Files	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
06 Configuration Management	0605.10h1System.12-10.h	0605.10h1System.12-10.h 10.04 Security of System Files	Windows machines should meet requirements for 'Security Options - Audit'	3.0.0
06 Configuration Management	0605.10h1System.12-10.h	0605.10h1System.12-10.h 10.04 Security of System Files	Windows machines should meet requirements for 'System Audit Policies - Account Management'	3.0.0
06 Configuration Management	0635.10k1Organizational.12-10.k	0635.10k1Organizational.12-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0636.10k2Organizational.1-10.k	0636.10k2Organizational.1-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0637.10k2Organizational.2-10.k	0637.10k2Organizational.2-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
06 Configuration Management	0638.10k2Organizational.34569-10.k	0638.10k2Organizational.34569-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0639.10k2Organizational.78-10.k	0639.10k2Organizational.78-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0640.10k2Organizational.1012-10.k	0640.10k2Organizational.1012-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0641.10k2Organizational.11-10.k	0641.10k2Organizational.11-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0642.10k3Organizational.12-10.k	0642.10k3Organizational.12-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0643.10k3Organizational.3-10.k	0643.10k3Organizational.3-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	3.0.0
06 Configuration Management	0644.10k3Organizational.4-10.k	0644.10k3Organizational.4-10.k 10.05 Security In Development and Support Processes	Windows machines should meet requirements for	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			'System Audit Policies - Detailed Tracking'	
07 Vulnerability Management	0709.10m1Organizational.1-10.m	0709.10m1Organizational.1-10.m 10.06 Technical Vulnerability Management	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
07 Vulnerability Management	0709.10m1Organizational.1-10.m	0709.10m1Organizational.1-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in container security configurations should be remediated	3.0.0
07 Vulnerability Management	0709.10m1Organizational.1-10.m	0709.10m1Organizational.1-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
07 Vulnerability Management	0709.10m1Organizational.1-10.m	0709.10m1Organizational.1-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
07 Vulnerability Management	0709.10m1Organizational.1-10.m	0709.10m1Organizational.1-10.m 10.06 Technical Vulnerability Management	Windows machines should meet requirements for 'Security Options - Microsoft Network Server'	3.0.0
07 Vulnerability Management	0711.10m2Organizational.23-10.m	0711.10m2Organizational.23-10.m 10.06 Technical Vulnerability Management	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
07 Vulnerability Management	0713.10m2Organizational.5-10.m	0713.10m2Organizational.5-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in security configuration on your machines	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			should be remediated	
07 Vulnerability Management	0714.10m2Organizational.7-10.m	0714.10m2Organizational.7-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
07 Vulnerability Management	0715.10m2Organizational.8-10.m	0715.10m2Organizational.8-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in container security configurations should be remediated	3.0.0
07 Vulnerability Management	0717.10m3Organizational.2-10.m	0717.10m3Organizational.2-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
07 Vulnerability Management	0718.10m3Organizational.34-10.m	0718.10m3Organizational.34-10.m 10.06 Technical Vulnerability Management	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
08 Network Protection	0805.01m1Organizational.12-01.m	0805.01m1Organizational.12-01.m 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0806.01m2Organizational.12356-01.m	0806.01m2Organizational.12356-01.m 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0809.01n2Organizational.1234-01.n	0809.01n2Organizational.1234-01.n 01.04 Network Access Control	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
08 Network Protection	0809.01n2Organizational.1234-01.n	0809.01n2Organizational.1234-01.n 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0810.01n2Organizational.5-01.n	0810.01n2Organizational.5-01.n 01.04 Network Access Control	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
08 Network Protection	0810.01n2Organizational.5-01.n	0810.01n2Organizational.5-01.n 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0811.01n2Organizational.6-01.n	0811.01n2Organizational.6-01.n 01.04 Network Access Control	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
08 Network Protection	0811.01n2Organizational.6-01.n	0811.01n2Organizational.6-01.n 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0812.01n2Organizational.8-01.n	0812.01n2Organizational.8-01.n 01.04 Network Access Control	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
08 Network Protection	0812.01n2Organizational.8-01.n	0812.01n2Organizational.8-01.n 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0814.01n1Organizational.12-01.n	0814.01n1Organizational.12-01.n 01.04 Network Access Control	Adaptive network hardening recommendations	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			should be applied on internet facing virtual machines	
08 Network Protection	0814.01n1Organizational.12-01.n	0814.01n1Organizational.12-01.n 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
08 Network Protection	0835.09n1Organizational.1-09.n	0835.09n1Organizational.1-09.n 09.06 Network Security Management	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
08 Network Protection	0835.09n1Organizational.1-09.n	0835.09n1Organizational.1-09.n 09.06 Network Security Management	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
08 Network Protection	0836.09.n2Organizational.1-09.n	0836.09.n2Organizational.1-09.n 09.06 Network Security Management	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
08 Network Protection	0858.09m1Organizational.4-09.m	0858.09m1Organizational.4-09.m 09.06 Network Security Management	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
08 Network Protection	0858.09m1Organizational.4-09.m	0858.09m1Organizational.4-09.m 09.06 Network Security Management	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
08 Network Protection	0858.09m1Organizational.4-09.m	0858.09m1Organizational.4-09.m 09.06 Network Security Management	Windows machines should meet requirements for	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			'Windows Firewall Properties'	
08 Network Protection	0859.09m1Organizational.78-09.m	0859.09m1Organizational.78-09.m 09.06 Network Security Management	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
08 Network Protection	0861.09m2Organizational.67-09.m	0861.09m2Organizational.67-09.m 09.06 Network Security Management	Windows machines should meet requirements for 'Security Options - Network Access'	3.0.0
08 Network Protection	0885.09n2Organizational.3-09.n	0885.09n2Organizational.3-09.n 09.06 Network Security Management	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
08 Network Protection	0887.09n2Organizational.5-09.n	0887.09n2Organizational.5-09.n 09.06 Network Security Management	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
08 Network Protection	0894.01m2Organizational.7-01.m	0894.01m2Organizational.7-01.m 01.04 Network Access Control	Internet-facing virtual machines should be protected with network security groups	3.0.0
Back-up	1699.09l1Organizational.10 - 09.l	Workforce members roles and responsibilities in the data backup process are identified and communicated to the workforce; in particular, Bring Your Own Device (BYOD) users are required to perform backups of organizational and/or client data on their devices.	Azure Backup should be enabled for Virtual Machines	3.0.0
09 Transmission Protection	0945.09y1Organizational.3-09.y	0945.09y1Organizational.3-09.y 09.09 Electronic Commerce Services	Audit Windows machines that do not contain the specified	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			certificates in Trusted Root	
Control of Operational Software	0606.10h2System.1 - 10.h	Applications and operating systems are successfully tested for usability, security and impact prior to production.	Vulnerabilities in container security configurations should be remediated	3.0.0
Control of Operational Software	0607.10h2System.23 - 10.h	The organization uses its configuration control program to maintain control of all implemented software and its system documentation and archive prior versions of implemented software and associated system documentation.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Control of Operational Software	0607.10h2System.23 - 10.h	The organization uses its configuration control program to maintain control of all implemented software and its system documentation and archive prior versions of implemented software and associated system documentation.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
11 Access Control	11180.01c3System.6-01.c	11180.01c3System.6-01.c 01.02 Authorized Access to Information Systems	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
11 Access Control	1119.01j2Organizational.3-01.j	1119.01j2Organizational.3-01.j 01.04 Network Access Control	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
11 Access Control	1123.01q1System.2-01.q	1123.01q1System.2-01.q 01.05 Operating System Access Control	Audit Windows machines that have extra accounts in the Administrators group	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
11 Access Control	1125.01q2System.1-01.q	1125.01q2System.1-01.q 01.05 Operating System Access Control	Audit Windows machines that have the specified members in the Administrators group	2.0.0
11 Access Control	1127.01q2System.3-01.q	1127.01q2System.3-01.q 01.05 Operating System Access Control	Audit Windows machines missing any of specified members in the Administrators group	2.0.0
11 Access Control	1143.01c1System.123-01.c	1143.01c1System.123-01.c 01.02 Authorized Access to Information Systems	Management ports should be closed on your virtual machines	3.0.0
11 Access Control	1148.01c2System.78-01.c	1148.01c2System.78-01.c 01.02 Authorized Access to Information Systems	Windows machines should meet requirements for 'Security Options - Accounts'	3.0.0
11 Access Control	1150.01c2System.10-01.c	1150.01c2System.10-01.c 01.02 Authorized Access to Information Systems	Management ports should be closed on your virtual machines	3.0.0
11 Access Control	1175.01j1Organizational.8-01.j	1175.01j1Organizational.8-01.j 01.04 Network Access Control	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
11 Access Control	1179.01j3Organizational.1-01.j	1179.01j3Organizational.1-01.j 01.04 Network Access Control	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
11 Access Control	1192.01l1Organizational.1-01.l	1192.01l1Organizational.1-01.l 01.04 Network Access Control	Management ports of virtual machines should be protected with just-in-time	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			network access control	
11 Access Control	1193.01l2Organizational.13-01.l	1193.01l2Organizational.13-01.l 01.04 Network Access Control	Management ports should be closed on your virtual machines	3.0.0
11 Access Control	1197.01l3Organizational.3-01.l	1197.01l3Organizational.3-01.l 01.04 Network Access Control	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
12 Audit Logging & Monitoring	1202.09aa1System.1-09.aa	1202.09aa1System.1-09.aa 09.10 Monitoring	System updates on virtual machine scale sets should be installed	3.0.0
12 Audit Logging & Monitoring	12100.09ab2System.15-09.ab	12100.09ab2System.15-09.ab 09.10 Monitoring	Virtual machines should have the Log Analytics extension installed	1.0.1
12 Audit Logging & Monitoring	12101.09ab1Organizational.3-09.ab	12101.09ab1Organizational.3-09.ab 09.10 Monitoring	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
12 Audit Logging & Monitoring	12102.09ab1Organizational.4-09.ab	12102.09ab1Organizational.4-09.ab 09.10 Monitoring	Audit Windows machines on which the Log Analytics agent is not connected as expected	2.0.0
12 Audit Logging & Monitoring	1215.09ab2System.7-09.ab	1215.09ab2System.7-09.ab 09.10 Monitoring	Virtual machines should have the Log Analytics extension installed	1.0.1
12 Audit Logging & Monitoring	1216.09ab3System.12-09.ab	1216.09ab3System.12-09.ab 09.10 Monitoring	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
12 Audit Logging & Monitoring	1217.09ab3System.3-09.ab	1217.09ab3System.3-09.ab 09.10 Monitoring	Audit Windows machines on which the Log Analytics agent is not connected as expected	2.0.0
12 Audit Logging & Monitoring	1232.09c3Organizational.12-09.c	1232.09c3Organizational.12-09.c 09.01 Documented Operating Procedures	Windows machines should meet requirements for 'User Rights Assignment'	3.0.0
12 Audit Logging & Monitoring	1277.09c2Organizational.4-09.c	1277.09c2Organizational.4-09.c 09.01 Documented Operating Procedures	Windows machines should meet requirements for 'Security Options - User Account Control'	3.0.0
16 Business Continuity & Disaster Recovery	1620.09l1Organizational.8-09.l	1620.09l1Organizational.8-09.l 09.05 Information Back-Up	Azure Backup should be enabled for Virtual Machines	3.0.0
16 Business Continuity & Disaster Recovery	1625.09l3Organizational.34-09.l	1625.09l3Organizational.34-09.l 09.05 Information Back-Up	Azure Backup should be enabled for Virtual Machines	3.0.0
16 Business Continuity & Disaster Recovery	1634.12b1Organizational.1-12.b	1634.12b1Organizational.1-12.b 12.01 Information Security Aspects of Business Continuity Management	Audit virtual machines without disaster recovery configured	1.0.0
16 Business Continuity & Disaster Recovery	1637.12b2Organizational.2-12.b	1637.12b2Organizational.2-12.b 12.01 Information Security Aspects of Business Continuity Management	Windows machines should meet requirements for 'Security Options - Recovery console'	3.0.0
16 Business Continuity & Disaster Recovery	1638.12b2Organizational.345-12.b	1638.12b2Organizational.345-12.b 12.01 Information Security Aspects of Business Continuity Management	Audit virtual machines without disaster recovery configured	1.0.0

IRS 1075 September 2016

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - IRS 1075 September 2016. For more information about this compliance standard, see IRS 1075 September 2016.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	9.3.1.12	Remote Access (AC-17)	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	9.3.1.12	Remote Access (AC-17)	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	9.3.1.12	Remote Access (AC-17)	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	9.3.1.12	Remote Access (AC-17)	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	9.3.1.2	Account Management (AC-2)	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	9.3.1.5	Separation of Duties (AC-5)	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	9.3.1.5	Separation of Duties (AC-5)	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	9.3.1.5	Separation of Duties (AC-5)	Audit Windows machines missing any of specified members in the Administrators group	2.0.0
Access Control	9.3.1.5	Separation of Duties (AC-5)	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Access Control	9.3.1.5	Separation of Duties (AC-5)	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	9.3.1.6	Least Privilege (AC-6)	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	9.3.1.6	Least Privilege (AC-6)	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	9.3.1.6	Least Privilege (AC-6)	Audit Windows machines missing any of specified members in the Administrators group	2.0.0
Access Control	9.3.1.6	Least Privilege (AC-6)	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Access Control	9.3.1.6	Least Privilege (AC-6)	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Risk Assessment	9.3.14.3	Vulnerability Scanning (RA-5)	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	9.3.14.3	Vulnerability Scanning (RA-5)	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	9.3.14.3	Vulnerability Scanning (RA-5)	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Communications Protection	9.3.16.5	Boundary Protection (SC-7)	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
System and Communications Protection	9.3.16.5	Boundary Protection (SC-7)	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	9.3.16.5	Boundary Protection (SC-7)	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	9.3.16.6	Transmission Confidentiality and Integrity (SC-8)	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	9.3.16.6	Transmission Confidentiality and Integrity (SC-8)	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
System and Communications Protection	9.3.16.6	Transmission Confidentiality and Integrity (SC-8)	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
System and Communications Protection	9.3.16.6	Transmission Confidentiality and Integrity (SC-8)	Windows machines should be configured to use secure communication protocols	4.1.1
System and Information Integrity	9.3.17.2	Flaw Remediation (SI-2)	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System and Information Integrity	9.3.17.2	Flaw Remediation (SI-2)	System updates on virtual machine scale sets should be installed	3.0.0
System and Information Integrity	9.3.17.2	Flaw Remediation (SI-2)	System updates should be installed on your machines	4.0.0
System and Information Integrity	9.3.17.2	Flaw Remediation (SI-2)	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
System and Information Integrity	9.3.17.2	Flaw Remediation (SI-2)	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Information Integrity	9.3.17.3	Malicious Code Protection (SI-3)	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	9.3.17.3	Malicious Code Protection (SI-3)	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	9.3.17.4	Information System Monitoring (SI-4)	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
System and Information Integrity	9.3.17.4	Information System Monitoring (SI-4)	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
System and Information Integrity	9.3.17.4	Information System Monitoring (SI-4)	Virtual machines should be connected to a specified workspace	1.1.0
Awareness and Training	9.3.3.11	Audit Generation (AU-12)	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Awareness and Training	9.3.3.11	Audit Generation (AU-12)	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Awareness and Training	9.3.3.11	Audit Generation (AU-12)	Virtual machines should be connected to a specified workspace	1.1.0
Awareness and Training	9.3.3.3	Content of Audit Records (AU-3)	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Awareness and Training	9.3.3.3	Content of Audit Records (AU-3)	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Awareness and Training	9.3.3.3	Content of Audit Records (AU-3)	Virtual machines should be connected to a specified workspace	1.1.0
Awareness and Training	9.3.3.6	Audit Review, Analysis, and Reporting (AU-6)	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Awareness and Training	9.3.3.6	Audit Review, Analysis, and Reporting (AU-6)	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Awareness and Training	9.3.3.6	Audit Review, Analysis, and Reporting (AU-6)	Virtual machines should be connected to a specified workspace	1.1.0
Configuration Management	9.3.5.11	User-Installed Software (CM-11)	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	9.3.5.7	Least Functionality (CM-7)	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Contingency Planning	9.3.6.6	Alternate Processing Site (CP-7)	Audit virtual machines without disaster recovery configured	1.0.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Linux machines that have accounts without passwords	3.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification and Authentication	9.3.7.5	Authenticator Management (IA-5)	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

ISO 27001:2013

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - ISO 27001:2013. For more information about this compliance standard, see ISO 27001:2013.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Cryptography	10.1.1	Policy on the use of cryptographic controls	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Cryptography	10.1.1	Policy on the use of cryptographic controls	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Cryptography	10.1.1	Policy on the use of cryptographic controls	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Cryptography	10.1.1	Policy on the use of cryptographic controls	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Operations Security	12.4.1	Event Logging	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Operations Security	12.4.1	Event Logging	Dependency agent should be enabled for listed virtual machine images	2.0.0
Operations Security	12.4.1	Event Logging	Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	2.0.0
Operations Security	12.4.1	Event Logging	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Operations Security	12.4.3	Administrator and operator logs	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Operations Security	12.4.3	Administrator and operator logs	Dependency agent should be enabled for listed virtual machine images	2.0.0
Operations Security	12.4.3	Administrator and operator logs	Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	2.0.0
Operations Security	12.4.3	Administrator and operator logs	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Operations Security	12.4.4	Clock Synchronization	[Preview]: Log Analytics Extension should be enabled for listed virtual	2.0.1-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			machine images	
Operations Security	12.4.4	Clock Synchronization	Dependency agent should be enabled for listed virtual machine images	2.0.0
Operations Security	12.4.4	Clock Synchronization	Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	2.0.0
Operations Security	12.4.4	Clock Synchronization	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Operations Security	12.5.1	Installation of software on operational systems	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Operations Security	12.6.1	Management of technical vulnerabilities	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operations Security	12.6.1	Management of technical vulnerabilities	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Operations Security	12.6.1	Management of technical vulnerabilities	System updates should be installed on your machines	4.0.0
Operations Security	12.6.1	Management of technical vulnerabilities	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Operations Security	12.6.2	Restrictions on software installation	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Communications Security	13.1.1	Network controls	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Access Control	9.1.2	Access to networks and network services	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	9.1.2	Access to networks and network services	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	9.1.2	Access to networks and network services	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	9.1.2	Access to networks and network services	Audit Linux machines that have accounts without passwords	3.1.0
Access Control	9.1.2	Access to networks and network services	Audit VMs that do not use managed disks	1.0.0
Access Control	9.1.2	Access to networks and network services	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	9.1.2	Access to networks and network services	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Access Control	9.2.4	Management of secret authentication information of users	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	9.2.4	Management of secret authentication information of users	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	9.2.4	Management of secret authentication information of users	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Access Control	9.2.4	Management of secret authentication information of users	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	9.4.3	Password management system	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	9.4.3	Password management system	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	9.4.3	Password management system	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Access Control	9.4.3	Password management system	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	9.4.3	Password management system	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Access Control	9.4.3	Password management system	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Access Control	9.4.3	Password management system	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Access Control	9.4.3	Password management system	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Microsoft Cloud for Sovereignty Baseline Confidential Policies

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for MCfS Sovereignty Baseline Confidential Policies. For more information about this compliance standard, see Microsoft Cloud for Sovereignty Policy portfolio.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SO.3 - Customer-Managed Keys	SO.3	Azure products must be configured to use Customer-Managed Keys when possible.	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
SO.4 - Azure Confidential Computing	SO.4	Azure products must be configured to use Azure Confidential Computing SKUs when possible.	Allowed virtual machine size SKUs	1.0.1

Microsoft Cloud for Sovereignty Baseline Global Policies

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for MCfS Sovereignty Baseline Global Policies. For more information about this compliance standard, see Microsoft Cloud for Sovereignty Policy portfolio.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SO.5 - Trusted Launch	SO.5	VMs should be configured with Trusted Launch SKUs and Trusted Launch enabled when possible.	Disks and OS image should support TrustedLaunch	1.0.0
SO.5 - Trusted Launch	SO.5	VMs should be configured with Trusted Launch SKUs and Trusted Launch enabled when possible.	Virtual Machine should have TrustedLaunch enabled	1.0.0

Microsoft cloud security benchmark

The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. To see how this service completely maps to the Microsoft cloud security benchmark, see the Azure Security Benchmark mapping files.

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - Microsoft cloud security benchmark.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Security	NS-1	Establish network segmentation boundaries	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Network Security	NS-1	Establish network segmentation boundaries	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Network Security	NS-1	Establish network segmentation boundaries	Internet-facing virtual machines should be protected with network security groups	3.0.0
Network Security	NS-1	Establish network segmentation boundaries	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Network Security	NS-3	Deploy firewall at the edge of enterprise	IP Forwarding on your virtual machine should be disabled	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		network		
Network Security	NS-3	Deploy firewall at the edge of enterprise network	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Network Security	NS-3	Deploy firewall at the edge of enterprise network	Management ports should be closed on your virtual machines	3.0.0
Network Security	NS-7	Simplify network security configuration	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Identity Management	IM-3	Manage application identities securely and automatically	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Identity Management	IM-6	Use strong authentication controls	Authentication to Linux machines should require SSH keys	3.2.0
Identity Management	IM-8	Restrict the exposure of credential and secrets	Machines should have secret findings resolved	1.0.2
Privileged Access	PA-2	Avoid standing access for accounts and permissions	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Data Protection	DP-3	Encrypt sensitive data in transit	Windows machines should be configured to use secure communication protocols	4.1.1
Data Protection	DP-4	Enable data at rest encryption by default	Linux virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	1.2.1
Data Protection	DP-4	Enable data at rest encryption by default	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
Data Protection	DP-4	Enable data at rest encryption by default	Windows virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	1.1.1
Asset Management	AM-2	Use only approved services	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Asset Management	AM-5	Use only approved applications in virtual machine	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Asset Management	AM-5	Use only approved applications in virtual	Allowlist rules in your adaptive application control policy should be	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		machine	updated	
Logging and Threat Detection	LT-1	Enable threat detection capabilities	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
Logging and Threat Detection	LT-2	Enable threat detection for identity and access management	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
Logging and Threat Detection	LT-4	Enable network logging for security investigation	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Logging and Threat Detection	LT-4	Enable network logging for security investigation	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	6.0.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines scale sets	5.1.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	4.0.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	3.1.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: Linux virtual machines should use only signed and trusted boot components	1.0.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	Guest Configuration extension should be installed on your machines	1.0.3
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	Linux machines should meet requirements for the Azure compute security baseline	2.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Posture and Vulnerability Management	PV-4	Audit and enforce secure configurations for compute resources	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Posture and Vulnerability Management	PV-5	Perform vulnerability assessments	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Posture and Vulnerability Management	PV-5	Perform vulnerability assessments	Machines should have secret findings resolved	1.0.2
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	[Preview]: System updates should be installed on your machines (powered by Update Center)	1.0.0-preview
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	Machines should be configured to periodically check for missing system updates	3.7.0
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	SQL servers on machines should have vulnerability findings resolved	1.0.0
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	System updates on virtual machine scale sets should be installed	3.0.0
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	System updates should be installed on your machines	4.0.0
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	Vulnerabilities in container security configurations should be remediated	3.0.0
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Posture and Vulnerability Management	PV-6	Rapidly and automatically remediate vulnerabilities	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Endpoint Security	ES-2	Use modern anti-malware software	Endpoint protection health issues should be resolved on your machines	1.0.0
Endpoint Security	ES-2	Use modern anti-malware software	Endpoint protection should be installed on your machines	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Endpoint Security	ES-2	Use modern anti-malware software	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Endpoint Security	ES-2	Use modern anti-malware software	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Endpoint Security	ES-2	Use modern anti-malware software	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
Endpoint Security	ES-3	Ensure anti-malware software and signatures are updated	Endpoint protection health issues should be resolved on your machines	1.0.0
Backup and Recovery	BR-1	Ensure regular automated backups	Azure Backup should be enabled for Virtual Machines	3.0.0
Backup and Recovery	BR-2	Protect backup and recovery data	Azure Backup should be enabled for Virtual Machines	3.0.0
DevOps Security	DS-6	Enforce security of workload throughout DevOps lifecycle	Vulnerabilities in container security configurations should be remediated	3.0.0

NIST SP 800-171 R2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - NIST SP 800-171 R2. For more information about this compliance standard, see NIST SP 800-171 R2.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of	Audit Linux machines that allow remote connections	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		authorized users, and devices (including other systems).	from accounts without passwords	
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Audit Linux machines that have accounts without passwords	3.1.0
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Authentication to Linux machines should require SSH keys	3.2.0
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Disk access resources should use private link	1.0.0
Access Control	3.1.1	Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Access Control	3.1.12	Monitor and control remote access sessions.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	3.1.12	Monitor and control remote access sessions.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	3.1.12	Monitor and control remote access sessions.	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	3.1.12	Monitor and control remote access sessions.	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	3.1.12	Monitor and control remote access sessions.	Disk access resources should use private link	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	3.1.13	Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.	Disk access resources should use private link	1.0.0
Access Control	3.1.14	Route remote access via managed access control points.	Disk access resources should use private link	1.0.0
Access Control	3.1.2	Limit system access to the types of transactions and functions that authorized users are permitted to execute.	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Disk access resources should use private link	1.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	IP Forwarding on your virtual machine should be disabled	3.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Management ports should be closed on your virtual machines	3.0.0
Access Control	3.1.3	Control the flow of CUI in accordance with approved authorizations.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	3.1.4	Separate the duties of individuals to reduce the risk of malevolent activity without collusion.	Audit Windows machines missing any of specified	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			members in the Administrators group	
Access Control	3.1.4	Separate the duties of individuals to reduce the risk of malevolent activity without collusion.	Audit Windows machines that have the specified members in the Administrators group	2.0.0
Risk Assessment	3.11.2	Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	3.11.2	Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.	SQL servers on machines should have vulnerability findings resolved	1.0.0
Risk Assessment	3.11.2	Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.	Vulnerabilities in container security configurations should be remediated	3.0.0
Risk Assessment	3.11.2	Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	3.11.2	Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Risk Assessment	3.11.3	Remediate vulnerabilities in accordance with risk assessments.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	3.11.3	Remediate vulnerabilities in accordance with risk assessments.	SQL servers on machines should have vulnerability findings resolved	1.0.0
Risk Assessment	3.11.3	Remediate vulnerabilities in accordance with risk assessments.	Vulnerabilities in container security configurations	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			should be remediated	
Risk Assessment	3.11.3	Remediate vulnerabilities in accordance with risk assessments.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	3.11.3	Remediate vulnerabilities in accordance with risk assessments.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Disk access resources should use private link	1.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	IP Forwarding on your virtual machine should be disabled	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Management ports should be closed on your virtual machines	3.0.0
System and Communications Protection	3.13.1	Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.10	Establish and manage cryptographic keys for cryptography employed in organizational systems.	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
System and Communications Protection	3.13.10	Establish and manage cryptographic keys for cryptography employed in organizational systems.	OS and data disks should be encrypted with a customer-managed key	3.0.0
System and Communications Protection	3.13.16	Protect the confidentiality of CUI at rest.	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Disk access resources should use private link	1.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	IP Forwarding on your virtual machine should be disabled	3.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Management ports should be closed on your virtual machines	3.0.0
System and Communications Protection	3.13.2	Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		logically separated from internal networks.		
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Disk access resources should use private link	1.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	IP Forwarding on your virtual machine should be disabled	3.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Management ports should be closed on your virtual machines	3.0.0
System and Communications Protection	3.13.5	Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications	3.13.6	Deny network communications traffic by default and allow network	Internet-facing virtual machines should be	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection		communications traffic by exception (i.e., deny all, permit by exception).	protected with network security groups	
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Management ports should be closed on your virtual machines	3.0.0
System and Communications Protection	3.13.6	Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	3.13.8	Implement cryptographic mechanisms to prevent unauthorized disclosure of CUI during transmission unless otherwise protected by alternative physical safeguards.	Windows machines should be configured to use secure communication protocols	4.1.1
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	System updates on virtual machine scale sets should be installed	3.0.0
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	System updates should be installed on your machines	4.0.0
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	Vulnerabilities in security configuration on your	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			machines should be remediated	
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Information Integrity	3.14.1	Identify, report, and correct system flaws in a timely manner.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System and Information Integrity	3.14.2	Provide protection from malicious code at designated locations within organizational systems.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	3.14.2	Provide protection from malicious code at designated locations within organizational systems.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	3.14.2	Provide protection from malicious code at designated locations within organizational systems.	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
System and Information Integrity	3.14.2	Provide protection from malicious code at designated locations within organizational systems.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	3.14.2	Provide protection from malicious code at designated locations within organizational systems.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System and Information Integrity	3.14.3	Monitor system security alerts and advisories and take action in response.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	3.14.4	Update malicious code protection mechanisms when new releases are available.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	3.14.4	Update malicious code protection mechanisms when new releases are available.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	3.14.4	Update malicious code protection mechanisms when new releases are available.	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Information Integrity	3.14.4	Update malicious code protection mechanisms when new releases are available.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	3.14.4	Update malicious code protection mechanisms when new releases are available.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System and Information Integrity	3.14.5	Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	3.14.5	Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
System and Information Integrity	3.14.5	Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed.	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
System and Information Integrity	3.14.5	Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	3.14.5	Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System and Information Integrity	3.14.6	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
System and Information Integrity	3.14.6	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Information Integrity	3.14.6	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Guest Configuration extension should be installed on your machines	1.0.3
System and Information Integrity	3.14.6	Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
System and Information Integrity	3.14.7	Identify unauthorized use of organizational systems.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
System and Information Integrity	3.14.7	Identify unauthorized use of organizational systems.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
System and Information Integrity	3.14.7	Identify unauthorized use of organizational systems.	Guest Configuration extension should be installed on your machines	1.0.3
System and Information Integrity	3.14.7	Identify unauthorized use of organizational systems.	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	Guest Configuration extension should be installed on your machines	1.0.3

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	Virtual machines should have the Log Analytics extension installed	1.0.1
Audit and Accountability	3.3.1	Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	Guest Configuration extension should be installed on your machines	1.0.3
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	Virtual machines should be connected to a specified workspace	1.1.0
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	Virtual machines should have the Log Analytics extension installed	1.0.1
Audit and Accountability	3.3.2	Ensure that the actions of individual system users can be uniquely traced to those users, so they can be held accountable for their actions.	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Configuration Management	3.4.1	Establish and maintain baseline configurations and inventories of organizational systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles.	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Configuration Management	3.4.1	Establish and maintain baseline configurations and inventories of organizational systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles.	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Configuration Management	3.4.2	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Configuration Management	3.4.2	Establish and enforce security configuration settings for information technology products employed in organizational systems.	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Configuration Management	3.4.6	Employ the principle of least functionality by configuring organizational systems to provide only essential capabilities.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Configuration Management	3.4.6	Employ the principle of least functionality by configuring organizational systems to provide only essential capabilities.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	3.4.7	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	3.4.7	Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	3.4.8	Apply deny-by-exception (blacklisting) policy to prevent the use of unauthorized software or deny-all, permit-by-exception (whitelisting) policy to allow the execution of authorized software.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	3.4.8	Apply deny-by-exception (blacklisting) policy to prevent the use of unauthorized software or deny-all, permit-by-exception (whitelisting) policy to allow the execution of authorized software.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	3.4.9	Control and monitor user-installed software.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	3.4.9	Control and monitor user-installed software.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	3.5.10	Store and transmit only cryptographically-protected passwords.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to	Authentication to Linux machines should require SSH keys	3.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		allowing access to organizational systems.		
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification and Authentication	3.5.2	Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	3.5.4	Employ replay-resistant authentication mechanisms for network access to privileged and non-privileged accounts.	Windows machines should meet requirements for 'Security Options - Network Security'	3.0.0
Identification and Authentication	3.5.7	Enforce a minimum password complexity and change of characters when new passwords are created.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	3.5.7	Enforce a minimum password complexity and change of characters when new passwords are created.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	3.5.7	Enforce a minimum password complexity and change of characters when new passwords are created.	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification and Authentication	3.5.7	Enforce a minimum password complexity and change of characters when new passwords are created.	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification and Authentication	3.5.7	Enforce a minimum password complexity and change of characters when new passwords are created.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	3.5.8	Prohibit password reuse for a specified number of generations.	Add system-assigned managed identity to enable Guest Configuration	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			assignments on virtual machines with no identities	
Identification and Authentication	3.5.8	Prohibit password reuse for a specified number of generations.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	3.5.8	Prohibit password reuse for a specified number of generations.	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification and Authentication	3.5.8	Prohibit password reuse for a specified number of generations.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Media Protection	3.8.9	Protect the confidentiality of backup CUI at storage locations.	Azure Backup should be enabled for Virtual Machines	3.0.0

NIST SP 800-53 Rev. 4

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 4. For more information about this compliance standard, see NIST SP 800-53 Rev. 4.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-2 (12)	Account Monitoring / Atypical Usage	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-3	Access Enforcement	Audit Linux machines that have accounts without passwords	3.1.0
Access Control	AC-3	Access Enforcement	Authentication to Linux machines should require SSH keys	3.2.0
Access Control	AC-3	Access Enforcement	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-3	Access Enforcement	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Access Control	AC-4	Information Flow Enforcement	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Access Control	AC-4	Information Flow Enforcement	Disk access resources should use private link	1.0.0
Access Control	AC-4	Information Flow Enforcement	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-4	Information Flow Enforcement	IP Forwarding on your virtual machine should be disabled	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports should be closed on your virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-4 (3)	Dynamic Information Flow Control	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC-4 (3)	Dynamic Information Flow Control	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			assignments on virtual machines with no identities	
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17	Remote Access	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17	Remote Access	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-17	Remote Access	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17	Remote Access	Disk access resources should use private link	1.0.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17 (1)	Automated Monitoring / Control	Disk access resources should use private link	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit And Accountability	AU-6	Audit Review, Analysis, And Reporting	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6	Audit Review, Analysis, And Reporting	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (4)	Central Review And Analysis	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (4)	Central Review And Analysis	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (4)	Central Review And Analysis	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-6 (4)	Central Review And Analysis	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-6 (5)	Integration / Scanning And Monitoring Capabilities	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit And Accountability	AU-12	Audit Generation	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-12	Audit Generation	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-12	Audit Generation	Guest Configuration extension should be installed on your machines	1.0.3

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit And Accountability	AU-12	Audit Generation	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	Guest Configuration extension should be installed on your machines	1.0.3
Audit And Accountability	AU-12 (1)	System-Wide / Time-Correlated Audit Trail	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Configuration Management	CM-6	Configuration Settings	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Configuration Management	CM-6	Configuration Settings	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Configuration Management	CM-7	Least Functionality	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7	Least Functionality	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (5)	Authorized Software / Whitelisting	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (5)	Authorized Software / Whitelisting	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Adaptive application controls for defining safe applications should be	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			enabled on your machines	
Configuration Management	CM-10	Software Usage Restrictions	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-11	User-Installed Software	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-11	User-Installed Software	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Contingency Planning	CP-7	Alternate Processing Site	Audit virtual machines without disaster recovery configured	1.0.0
Contingency Planning	CP-9	Information System Backup	Azure Backup should be enabled for Virtual Machines	3.0.0
Identification And Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification And Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification And Authentication	IA-5	Authenticator Management	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification And Authentication	IA-5	Authenticator Management	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification And Authentication	IA-5	Authenticator Management	Authentication to Linux machines should require SSH keys	3.2.0
Identification And Authentication	IA-5	Authenticator Management	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification And Authentication	IA-5	Authenticator Management	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			assignments on virtual machines with no identities	
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification And Authentication	IA-5 (1)	Password-Based Authentication	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Risk Assessment	RA-5	Vulnerability Scanning	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	SQL servers on machines should have vulnerability findings resolved	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in container security configurations should be remediated	3.0.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	RA-5	Vulnerability Scanning	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System And Communications Protection	SC-3	Security Function Isolation	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Communications Protection	SC-3	Security Function Isolation	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Communications Protection	SC-3	Security Function Isolation	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Communications Protection	SC-5	Denial Of Service Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System And Communications Protection	SC-7	Boundary Protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Disk access resources should use private link	1.0.0
System And Communications Protection	SC-7	Boundary Protection	Internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7	Boundary Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7	Boundary Protection	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System And Communications	SC-7	Boundary Protection	Management ports should be closed on your virtual machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection				
System And Communications Protection	SC-7	Boundary Protection	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Disk access resources should use private link	1.0.0
System And Communications Protection	SC-7 (3)	Access Points	Internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	IP Forwarding on your virtual machine should be disabled	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Management ports should be closed on your virtual machines	3.0.0
System And Communications Protection	SC-7 (3)	Access Points	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System And Communications Protection	SC-8	Transmission Confidentiality And Integrity	Windows machines should be configured to use secure communication protocols	4.1.1
System And Communications Protection	SC-8 (1)	Cryptographic Or Alternate Physical Protection	Windows machines should be configured to use secure communication protocols	4.1.1
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
System And Communications Protection	SC-12	Cryptographic Key Establishment And Management	OS and data disks should be encrypted with a customer-managed key	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Communications Protection	SC-28	Protection Of Information At Rest	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System And Communications Protection	SC-28 (1)	Cryptographic Protection	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System And Information Integrity	SI-2	Flaw Remediation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System And Information Integrity	SI-2	Flaw Remediation	System updates on virtual machine scale sets should be installed	3.0.0
System And Information Integrity	SI-2	Flaw Remediation	System updates should be installed on your machines	4.0.0
System And Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
System And Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Information Integrity	SI-3	Malicious Code Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Information Integrity	SI-3 (1)	Central Management	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System And Information Integrity	SI-3 (1)	Central Management	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System And Information Integrity	SI-3 (1)	Central Management	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System And Information Integrity	SI-4	Information System Monitoring	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
System And Information Integrity	SI-4	Information System Monitoring	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System And Information Integrity	SI-4	Information System Monitoring	Guest Configuration extension should be installed on your machines	1.0.3
System And Information Integrity	SI-4	Information System Monitoring	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
System And Information Integrity	SI-16	Memory Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0

NIST SP 800-53 Rev. 5

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - NIST SP 800-53 Rev. 5. For more information about this compliance standard, see NIST SP 800-53 Rev. 5.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-2 (12)	Account Monitoring for Atypical Usage	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-3	Access Enforcement	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-3	Access Enforcement	Audit Linux machines that have accounts without passwords	3.1.0
Access Control	AC-3	Access Enforcement	Authentication to Linux machines should require SSH keys	3.2.0
Access Control	AC-3	Access Enforcement	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-3	Access Enforcement	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	AC-4	Information Flow Enforcement	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Access Control	AC-4	Information Flow Enforcement	Disk access resources should use private link	1.0.0
Access Control	AC-4	Information Flow Enforcement	Internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-4	Information Flow Enforcement	IP Forwarding on your virtual machine should be disabled	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-4	Information Flow Enforcement	Management ports should be closed on your virtual machines	3.0.0
Access Control	AC-4	Information Flow Enforcement	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Access Control	AC-4 (3)	Dynamic Information Flow Control	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Access Control	AC-4 (3)	Dynamic Information Flow Control	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17	Remote Access	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17	Remote Access	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17	Remote Access	Deploy the Linux Guest Configuration extension to enable Guest	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			Configuration assignments on Linux VMs	
Access Control	AC-17	Remote Access	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17	Remote Access	Disk access resources should use private link	1.0.0
Access Control	AC-17 (1)	Monitoring and Control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Access Control	AC-17 (1)	Monitoring and Control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Access Control	AC-17 (1)	Monitoring and Control	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Access Control	AC-17 (1)	Monitoring and Control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Access Control	AC-17 (1)	Monitoring and Control	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Access Control	AC-17 (1)	Monitoring and Control	Disk access resources should use private link	1.0.0
Audit and Accountability	AU-6	Audit Record Review, Analysis, and Reporting	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	AU-6	Audit Record Review, Analysis, and Reporting	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit and Accountability	AU-6 (4)	Central Review and Analysis	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	AU-6 (4)	Central Review and Analysis	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Audit and Accountability	AU-6 (4)	Central Review and Analysis	Guest Configuration extension should be installed on your machines	1.0.3
Audit and Accountability	AU-6 (4)	Central Review and Analysis	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit and Accountability	AU-6 (5)	Integrated Analysis of Audit Records	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	AU-6 (5)	Integrated Analysis of Audit Records	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit and Accountability	AU-6 (5)	Integrated Analysis of Audit Records	Guest Configuration extension should be installed on your machines	1.0.3
Audit and Accountability	AU-6 (5)	Integrated Analysis of Audit Records	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit and Accountability	AU-12	Audit Record Generation	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	AU-12	Audit Record Generation	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit and Accountability	AU-12	Audit Record Generation	Guest Configuration extension should be installed on your machines	1.0.3
Audit and Accountability	AU-12	Audit Record Generation	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Audit and Accountability	AU-12 (1)	System-wide and Time-correlated Audit Trail	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Audit and Accountability	AU-12 (1)	System-wide and Time-correlated Audit Trail	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Audit and Accountability	AU-12 (1)	System-wide and Time-correlated Audit Trail	Guest Configuration extension should be installed on your machines	1.0.3
Audit and Accountability	AU-12 (1)	System-wide and Time-correlated Audit Trail	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Configuration Management	CM-6	Configuration Settings	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Configuration Management	CM-6	Configuration Settings	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Configuration Management	CM-7	Least Functionality	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7	Least Functionality	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (2)	Prevent Program Execution	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-7 (5)	Authorized Software ??? Allow-by-exception	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-7 (5)	Authorized Software ??? Allow-by-exception	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-10	Software Usage Restrictions	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Configuration Management	CM-11	User-installed Software	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Configuration Management	CM-11	User-installed Software	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Contingency Planning	CP-7	Alternate Processing Site	Audit virtual machines without disaster recovery configured	1.0.0
Contingency Planning	CP-9	System Backup	Azure Backup should be enabled for Virtual Machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification and Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA-5	Authenticator Management	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	IA-5	Authenticator Management	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification and Authentication	IA-5	Authenticator Management	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification and Authentication	IA-5	Authenticator Management	Authentication to Linux machines should require SSH keys	3.2.0
Identification and Authentication	IA-5	Authenticator Management	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification and Authentication	IA-5	Authenticator Management	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identification and Authentication	IA-5 (1)	Password-based Authentication	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Risk Assessment	RA-5	Vulnerability Monitoring and Scanning	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Risk Assessment	RA-5	Vulnerability Monitoring and Scanning	SQL servers on machines should have vulnerability findings resolved	1.0.0
Risk Assessment	RA-5	Vulnerability Monitoring and Scanning	Vulnerabilities in container security configurations should be remediated	3.0.0
Risk Assessment	RA-5	Vulnerability Monitoring and Scanning	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Risk Assessment	RA-5	Vulnerability Monitoring and Scanning	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Communications Protection	SC-3	Security Function Isolation	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Communications Protection	SC-3	Security Function Isolation	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Communications Protection	SC-3	Security Function Isolation	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System and Communications Protection	SC-5	Denial-of-service Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System and Communications Protection	SC-7	Boundary Protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	SC-7	Boundary Protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	SC-7	Boundary Protection	Disk access resources should use private link	1.0.0
System and Communications Protection	SC-7	Boundary Protection	Internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC-7	Boundary Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
System and Communications Protection	SC-7	Boundary Protection	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	SC-7	Boundary Protection	Management ports should be closed on your virtual machines	3.0.0
System and Communications Protection	SC-7	Boundary Protection	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC-7 (3)	Access Points	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
System and Communications Protection	SC-7 (3)	Access Points	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
System and Communications Protection	SC-7 (3)	Access Points	Disk access resources should use private link	1.0.0
System and Communications	SC-7 (3)	Access Points	Internet-facing virtual machines should be protected with network	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protection			security groups	
System and Communications Protection	SC-7 (3)	Access Points	IP Forwarding on your virtual machine should be disabled	3.0.0
System and Communications Protection	SC-7 (3)	Access Points	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
System and Communications Protection	SC-7 (3)	Access Points	Management ports should be closed on your virtual machines	3.0.0
System and Communications Protection	SC-7 (3)	Access Points	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
System and Communications Protection	SC-8	Transmission Confidentiality and Integrity	Windows machines should be configured to use secure communication protocols	4.1.1
System and Communications Protection	SC-8 (1)	Cryptographic Protection	Windows machines should be configured to use secure communication protocols	4.1.1
System and Communications Protection	SC-12	Cryptographic Key Establishment and Management	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
System and Communications Protection	SC-12	Cryptographic Key Establishment and Management	OS and data disks should be encrypted with a customer-managed key	3.0.0
System and Communications Protection	SC-28	Protection of Information at Rest	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System and Communications Protection	SC-28 (1)	Cryptographic Protection	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
System and Information Integrity	SI-2	Flaw Remediation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System and Information Integrity	SI-2	Flaw Remediation	System updates on virtual machine scale sets should be installed	3.0.0
System and Information Integrity	SI-2	Flaw Remediation	System updates should be installed on your machines	4.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
System and Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
System and Information Integrity	SI-2	Flaw Remediation	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
System and Information Integrity	SI-3	Malicious Code Protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
System and Information Integrity	SI-3	Malicious Code Protection	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
System and Information Integrity	SI-3	Malicious Code Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
System and Information Integrity	SI-4	System Monitoring	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
System and Information Integrity	SI-4	System Monitoring	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
System and Information Integrity	SI-4	System Monitoring	Guest Configuration extension should be installed on your machines	1.0.3
System and Information Integrity	SI-4	System Monitoring	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
System and Information Integrity	SI-16	Memory Protection	Windows Defender Exploit Guard should be enabled on your machines	2.0.0

NL BIO Cloud Theme

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for NL BIO Cloud Theme. For more information about this compliance standard, see Baseline Information Security Government Cybersecurity - Digital Government (digitaleoverheid.nl).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	System updates on virtual machine scale sets should be installed	3.0.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	System updates should be installed on your machines	4.0.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
C.04.3 Technical vulnerability management - Timelines	C.04.3	If the probability of abuse and the expected damage are both high, patches are installed no later than within a week.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
C.04.6 Technical vulnerability	C.04.6	Technical weaknesses can be remedied by performing patch	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
management - Timelines		management in a timely manner.		
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	System updates on virtual machine scale sets should be installed	3.0.0
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	System updates should be installed on your machines	4.0.0
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
C.04.6 Technical vulnerability management - Timelines	C.04.6	Technical weaknesses can be remedied by performing patch management in a timely manner.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	System updates on virtual machine scale sets should be installed	3.0.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	System updates should be installed on your machines	4.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
C.04.7 Technical vulnerability management - Evaluated	C.04.7	Evaluations of technical vulnerabilities are recorded and reported.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
C.04.8 Technical vulnerability management - Evaluated	C.04.8	The evaluation reports contain suggestions for improvement and are communicated with managers/owners.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
C.04.8 Technical vulnerability management - Evaluated	C.04.8	The evaluation reports contain suggestions for improvement and are communicated with managers/owners.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
C.04.8 Technical vulnerability management - Evaluated	C.04.8	The evaluation reports contain suggestions for improvement and are communicated with managers/owners.	System updates should be installed on your machines	4.0.0
C.04.8 Technical vulnerability management - Evaluated	C.04.8	The evaluation reports contain suggestions for improvement and are communicated with managers/owners.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
U.03.1 Business Continuity Services - Redundancy	U.03.1	The agreed continuity is guaranteed by sufficiently logical or physically multiple system functions.	Audit virtual machines without disaster recovery configured	1.0.0
U.03.1 Business Continuity Services - Redundancy	U.03.1	The agreed continuity is guaranteed by sufficiently logical or physically multiple system functions.	Azure Backup should be enabled for Virtual Machines	3.0.0
U.03.2 Business Continuity Services - Continuity requirements	U.03.2	The continuity requirements for cloud services agreed with the CSC are ensured by the system architecture.	Audit virtual machines without disaster recovery configured	1.0.0
U.03.2 Business Continuity Services -	U.03.2	The continuity requirements for cloud services agreed with the	Azure Backup should be enabled for Virtual Machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Continuity requirements		CSC are ensured by the system architecture.		
U.04.1 Data and Cloud Service Recovery - Restore function	U.04.1	The data and cloud services are restored within the agreed period and maximum data loss and made available to the CSC.	Audit virtual machines without disaster recovery configured	1.0.0
U.04.2 Data and Cloud Service Recovery - Restore function	U.04.2	The continuous process of recoverable protection of data is monitored.	Audit virtual machines without disaster recovery configured	1.0.0
U.04.3 Data and Cloud Service Recovery - Tested	U.04.3	The functioning of recovery functions is periodically tested and the results are shared with the CSC.	Audit virtual machines without disaster recovery configured	1.0.0
U.05.1 Data protection - Cryptographic measures	U.05.1	Data transport is secured with cryptography where key management is carried out by the CSC itself if possible.	Windows machines should be configured to use secure communication protocols	4.1.1
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	6.0.0-preview
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines scale sets	5.1.0-preview
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	4.0.0-preview
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	3.1.0-preview
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	OS and data disks should be encrypted with a customer-managed key	3.0.0
U.05.2 Data protection - Cryptographic measures	U.05.2	Data stored in the cloud service shall be protected to the latest state of the art.	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Disk access resources should use private link	1.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Internet-facing virtual machines should be protected with network security groups	3.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	IP Forwarding on your virtual machine should be disabled	3.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Management ports should be closed on your virtual machines	3.0.0
U.07.1 Data separation - Isolated	U.07.1	Permanent isolation of data is a multi-tenant architecture. Patches are realized in a controlled manner.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
U.09.3 Malware Protection - Detection,	U.09.3	The malware protection runs on different environments.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
prevention and recovery				
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	IP Forwarding on your virtual machine should be disabled	3.0.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	System updates on virtual machine scale sets should be installed	3.0.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	System updates should be installed on your machines	4.0.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
U.09.3 Malware Protection - Detection, prevention and recovery	U.09.3	The malware protection runs on different environments.	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
U.10.2 Access to IT services and data -	U.10.2	Under the responsibility of the CSP, access is granted to	Audit Linux machines that allow remote connections	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Users		administrators.	from accounts without passwords	
U.10.2 Access to IT services and data - Users	U.10.2	Under the responsibility of the CSP, access is granted to administrators.	Audit Linux machines that have accounts without passwords	3.1.0
U.10.2 Access to IT services and data - Users	U.10.2	Under the responsibility of the CSP, access is granted to administrators.	Audit VMs that do not use managed disks	1.0.0
U.10.2 Access to IT services and data - Users	U.10.2	Under the responsibility of the CSP, access is granted to administrators.	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
U.10.3 Access to IT services and data - Users	U.10.3	Only users with authenticated equipment can access IT services and data.	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
U.10.3 Access to IT services and data - Users	U.10.3	Only users with authenticated equipment can access IT services and data.	Audit Linux machines that have accounts without passwords	3.1.0
U.10.3 Access to IT services and data - Users	U.10.3	Only users with authenticated equipment can access IT services and data.	Audit VMs that do not use managed disks	1.0.0
U.10.3 Access to IT services and data - Users	U.10.3	Only users with authenticated equipment can access IT services and data.	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
U.10.5 Access to IT services and data - Competent	U.10.5	Access to IT services and data is limited by technical measures and has been implemented.	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
U.10.5 Access to IT services and data - Competent	U.10.5	Access to IT services and data is limited by technical measures and has been implemented.	Audit Linux machines that have accounts without passwords	3.1.0
U.10.5 Access to IT services and data - Competent	U.10.5	Access to IT services and data is limited by technical measures and has been implemented.	Audit VMs that do not use managed disks	1.0.0
U.10.5 Access to IT services and data - Competent	U.10.5	Access to IT services and data is limited by technical measures and has been implemented.	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
U.11.1 Cryptoservices -	U.11.1	In the cryptography policy, at least the subjects in accordance	Audit Windows machines that do not store passwords	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Policy		with BIO have been elaborated.	using reversible encryption	
U.11.1 Cryptoservices - Policy	U.11.1	In the cryptography policy, at least the subjects in accordance with BIO have been elaborated.	Windows machines should be configured to use secure communication protocols	4.1.1
U.11.2 Cryptoservices - Cryptographic measures	U.11.2	In case of PKIoverheid certificates use PKIoverheid requirements for key management. In other situations use ISO11770.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
U.11.2 Cryptoservices - Cryptographic measures	U.11.2	In case of PKIoverheid certificates use PKIoverheid requirements for key management. In other situations use ISO11770.	Windows machines should be configured to use secure communication protocols	4.1.1
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	6.0.0-preview
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines scale sets	5.1.0-preview
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	4.0.0-preview
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	3.1.0-preview
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	Managed disks should be double encrypted with both platform-managed and customer-managed keys	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	OS and data disks should be encrypted with a customer-managed key	3.0.0
U.11.3 Cryptoservices - Encrypted	U.11.3	Sensitive data is always encrypted, with private keys managed by the CSC.	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
U.12.1 Interfaces - Network connections	U.12.1	In connection points with external or untrusted zones, measures are taken against attacks.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
U.12.1 Interfaces - Network connections	U.12.1	In connection points with external or untrusted zones, measures are taken against attacks.	IP Forwarding on your virtual machine should be disabled	3.0.0
U.12.2 Interfaces - Network connections	U.12.2	Network components are such that network connections between trusted and untrusted networks are limited.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
U.12.2 Interfaces - Network connections	U.12.2	Network components are such that network connections between trusted and untrusted networks are limited.	IP Forwarding on your virtual machine should be disabled	3.0.0
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	Dependency agent should be enabled for listed virtual machine images	2.0.0
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	Guest Configuration extension should be installed on your machines	1.0.3
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
U.15.1 Logging and monitoring - Events logged	U.15.1	The violation of the policy rules is recorded by the CSP and the CSC.	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
U.15.3 Logging and monitoring - Events logged	U.15.3	CSP maintains a list of all assets that are critical in terms of logging and monitoring and reviews this list.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
U.15.3 Logging and monitoring - Events logged	U.15.3	CSP maintains a list of all assets that are critical in terms of logging and monitoring and reviews this list.	Dependency agent should be enabled for listed virtual machine images	2.0.0
U.15.3 Logging and monitoring - Events logged	U.15.3	CSP maintains a list of all assets that are critical in terms of logging and monitoring and reviews this list.	Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	2.0.0
U.15.3 Logging and monitoring - Events logged	U.15.3	CSP maintains a list of all assets that are critical in terms of logging and monitoring and reviews this list.	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
U.17.1 Multi-tenant architecture - Encrypted	U.17.1	CSC data on transport and at rest is encrypted.	Audit virtual machines without disaster recovery configured	1.0.0
U.17.1 Multi-tenant architecture - Encrypted	U.17.1	CSC data on transport and at rest is encrypted.	Azure Backup should be enabled for Virtual Machines	3.0.0

PCI DSS 3.2.1

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see PCI DSS 3.2.1. For more information about this compliance standard, see PCI DSS 3.2.1.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 1	1.3.2	PCI DSS requirement 1.3.2	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Requirement 1	1.3.4	PCI DSS requirement 1.3.4	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Requirement 10	10.5.4	PCI DSS requirement 10.5.4	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Requirement 11	11.2.1	PCI DSS requirement 11.2.1	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 11	11.2.1	PCI DSS requirement 11.2.1	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 11	11.2.1	PCI DSS requirement 11.2.1	System updates should be installed on your machines	4.0.0
Requirement 11	11.2.1	PCI DSS requirement 11.2.1	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 5	5.1	PCI DSS requirement 5.1	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 5	5.1	PCI DSS requirement 5.1	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 5	5.1	PCI DSS requirement 5.1	System updates should be installed on your machines	4.0.0
Requirement 5	5.1	PCI DSS requirement 5.1	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 6	6.2	PCI DSS requirement 6.2	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 6	6.2	PCI DSS requirement 6.2	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 6	6.2	PCI DSS requirement 6.2	System updates should be installed on your machines	4.0.0
Requirement 6	6.2	PCI DSS requirement 6.2	Vulnerabilities in security configuration on your machines should be remediated	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 6	6.6	PCI DSS requirement 6.6	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 6	6.6	PCI DSS requirement 6.6	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 6	6.6	PCI DSS requirement 6.6	System updates should be installed on your machines	4.0.0
Requirement 6	6.6	PCI DSS requirement 6.6	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 8	8.2.3	PCI DSS requirement 8.2.3	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Requirement 8	8.2.3	PCI DSS requirement 8.2.3	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Requirement 8	8.2.3	PCI DSS requirement 8.2.3	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Requirement 8	8.2.3	PCI DSS requirement 8.2.3	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Requirement 8	8.2.3	PCI DSS requirement 8.2.3	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Requirement 8	8.2.3	PCI DSS requirement 8.2.3	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Requirement 8	8.2.5	PCI DSS requirement 8.2.5	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Requirement 8	8.2.5	PCI DSS requirement 8.2.5	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Requirement 8	8.2.5	PCI DSS requirement 8.2.5	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Requirement 8	8.2.5	PCI DSS requirement 8.2.5	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 8	8.2.5	PCI DSS requirement 8.2.5	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Requirement 8	8.2.5	PCI DSS requirement 8.2.5	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

PCI DSS v4.0

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for PCI DSS v4.0. For more information about this compliance standard, see PCI DSS v4.0.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 01: Install and Maintain Network Security Controls	1.3.2	Network access to and from the cardholder data environment is restricted	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Requirement 01: Install and Maintain Network Security Controls	1.4.2	Network connections between trusted and untrusted networks are controlled	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Requirement 10: Log and Monitor All Access to System Components and Cardholder Data	10.2.2	Audit logs are implemented to support the detection of anomalies and suspicious activity, and the forensic analysis of events	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Requirement 10: Log and Monitor All Access to System Components and Cardholder Data	10.3.3	Audit logs are protected from destruction and unauthorized modifications	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Requirement 11: Test Security of Systems and Networks Regularly	11.3.1	External and internal vulnerabilities are regularly identified, prioritized, and addressed	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 11: Test Security of Systems and Networks Regularly	11.3.1	External and internal vulnerabilities are regularly	Monitor missing Endpoint Protection in Azure Security Center	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		identified, prioritized, and addressed		
Requirement 11: Test Security of Systems and Networks Regularly	11.3.1	External and internal vulnerabilities are regularly identified, prioritized, and addressed	System updates should be installed on your machines	4.0.0
Requirement 11: Test Security of Systems and Networks Regularly	11.3.1	External and internal vulnerabilities are regularly identified, prioritized, and addressed	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.1	Malicious software (malware) is prevented, or detected and addressed	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.1	Malicious software (malware) is prevented, or detected and addressed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.1	Malicious software (malware) is prevented, or detected and addressed	System updates should be installed on your machines	4.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.1	Malicious software (malware) is prevented, or detected and addressed	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.2	Malicious software (malware) is prevented, or detected and addressed	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.2	Malicious software (malware) is prevented, or detected and addressed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.2	Malicious software (malware) is prevented, or detected and addressed	System updates should be installed on your machines	4.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.2	Malicious software (malware) is prevented, or detected and addressed	Vulnerabilities in security configuration on your machines should be remediated	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.3	Malicious software (malware) is prevented, or detected and addressed	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.3	Malicious software (malware) is prevented, or detected and addressed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.3	Malicious software (malware) is prevented, or detected and addressed	System updates should be installed on your machines	4.0.0
Requirement 05: Protect All Systems and Networks from Malicious Software	5.2.3	Malicious software (malware) is prevented, or detected and addressed	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 06: Develop and Maintain Secure Systems and Software	6.3.3	Security vulnerabilities are identified and addressed	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 06: Develop and Maintain Secure Systems and Software	6.3.3	Security vulnerabilities are identified and addressed	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 06: Develop and Maintain Secure Systems and Software	6.3.3	Security vulnerabilities are identified and addressed	System updates should be installed on your machines	4.0.0
Requirement 06: Develop and Maintain Secure Systems and Software	6.3.3	Security vulnerabilities are identified and addressed	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 06: Develop and Maintain Secure Systems and Software	6.4.1	Public-facing web applications are protected against attacks	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Requirement 06: Develop and Maintain Secure Systems and Software	6.4.1	Public-facing web applications are protected against attacks	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Requirement 06: Develop and Maintain	6.4.1	Public-facing web applications are protected against attacks	System updates should be installed on your machines	4.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Secure Systems and Software				
Requirement 06: Develop and Maintain Secure Systems and Software	6.4.1	Public-facing web applications are protected against attacks	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Requirement 08: Identify Users and Authenticate Access to System Components	8.3.6	Strong authentication for users and administrators is established and managed	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Requirement 08: Identify Users and Authenticate Access to System Components	8.3.6	Strong authentication for users and administrators is established and managed	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Requirement 08: Identify Users and Authenticate Access to System Components	8.3.6	Strong authentication for users and administrators is established and managed	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Requirement 08: Identify Users and Authenticate Access to System Components	8.3.6	Strong authentication for users and administrators is established and managed	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Requirement 08: Identify Users and Authenticate Access to System Components	8.3.6	Strong authentication for users and administrators is established and managed	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Requirement 08: Identify Users and Authenticate Access to System Components	8.3.6	Strong authentication for users and administrators is established and managed	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Reserve Bank of India - IT Framework for NBFC

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - Reserve Bank of India - IT Framework for NBFC. For more information about this compliance standard, see Reserve Bank of India - IT Framework for NBFC.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
IT Governance	1	IT Governance-1	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
IT Governance	1	IT Governance-1	SQL servers on machines should have vulnerability findings resolved	1.0.0
IT Governance	1	IT Governance-1	System updates on virtual machine scale sets should be installed	3.0.0
IT Governance	1	IT Governance-1	System updates should be installed on your machines	4.0.0
IT Governance	1	IT Governance-1	Vulnerabilities in container security configurations should be remediated	3.0.0
IT Governance	1	IT Governance-1	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
IT Governance	1	IT Governance-1	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
IT Governance	1.1	IT Governance-1.1	IP Forwarding on your virtual machine should be disabled	3.0.0
IT Governance	1.1	IT Governance-1.1	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
IT Governance	1.1	IT Governance-1.1	Management ports should be closed on your virtual machines	3.0.0
IT Policy	2	IT Policy-2	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
IT Policy	2	IT Policy-2	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Information and Cyber Security	3.1.b	Segregation of Functions-3.1	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
Information and Cyber Security	3.1.b	Segregation of Functions-3.1	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview
Information and Cyber Security	3.1.b	Segregation of Functions-3.1	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Information and Cyber Security	3.1.c	Role based Access Control-3.1	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Information and Cyber Security	3.1.g	Trails-3.1	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Information and Cyber Security	3.1.g	Trails-3.1	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Information and Cyber Security	3.1.g	Trails-3.1	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Information and Cyber Security	3.1.g	Trails-3.1	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Information and Cyber Security	3.1.g	Trails-3.1	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Information and Cyber Security	3.1.g	Trails-3.1	Virtual machines should have the Log Analytics extension installed	1.0.1
Information and Cyber Security	3.1.h	Public Key Infrastructure (PKI)-3.1	Managed disks should use a specific set of disk encryption sets for the customer-managed key encryption	2.0.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	SQL servers on machines should have vulnerability findings resolved	1.0.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	System updates on virtual machine scale sets should be installed	3.0.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	System updates should be installed on your machines	4.0.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	Vulnerabilities in container security configurations should be remediated	3.0.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Information and Cyber Security	3.3	Vulnerability Management-3.3	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
IT Operations	4.2	IT Operations-4.2	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
IT Operations	4.4.a	IT Operations-4.4	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
IT Operations	4.4.b	MIS For Top Management-4.4	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
IS Audit	5	Policy for Information System Audit (IS Audit)-5	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
IS Audit	5	Policy for Information System Audit (IS Audit)-5	Internet-facing virtual machines should be protected with network security groups	3.0.0
IS Audit	5	Policy for Information System Audit (IS Audit)-5	IP Forwarding on your virtual machine should be disabled	3.0.0
IS Audit	5	Policy for Information System Audit (IS Audit)-5	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
IS Audit	5.2	Coverage-5.2	Azure Backup should be enabled for Virtual Machines	3.0.0
Business Continuity Planning	6	Business Continuity Planning (BCP) and Disaster Recovery-6	Audit virtual machines without disaster recovery configured	1.0.0
Business Continuity Planning	6	Business Continuity Planning (BCP) and Disaster Recovery-6	Azure Backup should be enabled for Virtual Machines	3.0.0
Business Continuity Planning	6.2	Recovery strategy / Contingency Plan-6.2	Audit virtual machines without disaster recovery configured	1.0.0
Business Continuity Planning	6.2	Recovery strategy / Contingency Plan-6.2	Azure Backup should be enabled for Virtual Machines	3.0.0
Business Continuity Planning	6.3	Recovery strategy / Contingency Plan-6.3	Azure Backup should be enabled for Virtual Machines	3.0.0
Business Continuity Planning	6.4	Recovery strategy / Contingency Plan-6.4	Audit virtual machines without disaster recovery configured	1.0.0

Reserve Bank of India IT Framework for Banks v2016

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - RBI ITF Banks v2016. For more information about this compliance standard, see RBI ITF Banks v2016 (PDF).

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	6.0.0-preview
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines scale sets	5.1.0-preview
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	4.0.0-preview
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	3.1.0-preview
Network Management And Security		Network Inventory-4.2	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Network Management And Security		Network Inventory-4.2	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
Network Management And Security		Network Device Configuration Management-4.3	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview
Patch/Vulnerability & Change Management		Patch/Vulnerability & Change Management-7.1	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Management And Security		Network Inventory-4.2	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Network Management And Security		Network Device Configuration Management-4.3	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Network Management And Security		Network Device Configuration Management-4.3	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Network Management And Security		Network Inventory-4.2	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Incident Response & Management		Recovery From Cyber - Incidents-19.4	Audit virtual machines without disaster recovery configured	1.0.0
Authentication Framework For Customers		Authentication Framework For Customers-9.1	Authentication to Linux machines should require SSH keys	3.2.0
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.3	Azure Backup should be enabled for Virtual Machines	3.0.0
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	Endpoint protection health issues should be resolved on your machines	1.0.0
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	Endpoint protection should be installed on your machines	1.0.0
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Audit Log Settings		Audit Log Settings-17.1	Guest Configuration extension should be installed on your machines	1.0.3
Secure Configuration		Secure Configuration-5.2	Hotpatch should be enabled for Windows Server Azure Edition VMs	1.0.0
Network Management And Security		Network Device Configuration Management-4.3	Internet-facing virtual machines should be protected with network security groups	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Management And Security		Network Device Configuration Management-4.3	IP Forwarding on your virtual machine should be disabled	3.0.0
Audit Log Settings		Audit Log Settings-17.1	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Network Management And Security		Network Device Configuration Management-4.3	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Network Management And Security		Network Device Configuration Management-4.3	Management ports should be closed on your virtual machines	3.0.0
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Network Management And Security		Network Device Configuration Management-4.3	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Preventing Execution Of Unauthorised Software		Security Update Management-2.3	SQL servers on machines should have vulnerability findings resolved	1.0.0
Preventing Execution Of Unauthorised Software		Security Update Management-2.3	System updates on virtual machine scale sets should be installed	3.0.0
Preventing Execution Of Unauthorised Software		Security Update Management-2.3	System updates should be installed on your machines	4.0.0
Advanced Real-Timethreat Defenceand Management		Advanced Real-Timethreat Defenceand Management-13.1	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
User Access Control / Management		User Access Control / Management-8.4	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Preventing Execution Of Unauthorised Software		Security Update Management-2.3	Vulnerabilities in container security configurations should be remediated	3.0.0
Preventing Execution Of Unauthorised Software		Security Update Management-2.3	Vulnerabilities in security configuration on your machines should be remediated	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Preventing Execution Of Unauthorised Software		Security Update Management-2.3	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Secure Configuration		Secure Configuration-5.1	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
Secure Mail And Messaging Systems		Secure Mail And Messaging Systems-10.1	Windows machines should be configured to use secure communication protocols	4.1.1
Audit Log Settings		Audit Log Settings-17.1	Windows machines should meet requirements of the Azure compute security baseline	2.0.0

RMIT Malaysia

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - RMIT Malaysia. For more information about this compliance standard, see RMIT Malaysia.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Datacenter Operations	10.27	Datacenter Operations - 10.27	Deploy - Configure Log Analytics extension to be enabled on Windows virtual machine scale sets	3.1.0
Datacenter Operations	10.27	Datacenter Operations - 10.27	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Datacenter Operations	10.30	Datacenter Operations - 10.30	Azure Backup should be enabled for Virtual Machines	3.0.0
Network Resilience	10.33	Network Resilience - 10.33	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Network Resilience	10.33	Network Resilience - 10.33	Configure managed disks to disable public network access	2.0.0
Network Resilience	10.33	Network Resilience - 10.33	Internet-facing virtual machines should be protected with network security groups	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Network Resilience	10.33	Network Resilience - 10.33	IP Forwarding on your virtual machine should be disabled	3.0.0
Network Resilience	10.33	Network Resilience - 10.33	Managed disks should disable public network access	2.0.0
Network Resilience	10.33	Network Resilience - 10.33	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Network Resilience	10.35	Network Resilience - 10.35	Deploy - Configure Log Analytics extension to be enabled on Windows virtual machine scale sets	3.1.0
Cloud Services	10.49	Cloud Services - 10.49	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Cloud Services	10.49	Cloud Services - 10.49	Management ports should be closed on your virtual machines	3.0.0
Cloud Services	10.51	Cloud Services - 10.51	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Cloud Services	10.51	Cloud Services - 10.51	Audit virtual machines without disaster recovery configured	1.0.0
Cloud Services	10.53	Cloud Services - 10.53	Managed disks should use a specific set of disk encryption sets for the customer-managed key encryption	2.0.0
Cloud Services	10.53	Cloud Services - 10.53	OS and data disks should be encrypted with a customer-managed key	3.0.0
Access Control	10.54	Access Control - 10.54	Guest Configuration extension should be installed on your machines	1.0.3
Access Control	10.54	Access Control - 10.54	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Access Control	10.54	Access Control - 10.54	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Access Control	10.61	Access Control - 10.61	Guest Configuration extension should be installed on your machines	1.0.3
Access Control	10.61	Access Control - 10.61	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Access Control	10.61	Access Control - 10.61	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Patch and End-of-Life System Management	10.63	Patch and End-of-Life System Management - 10.63	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
Patch and End-of-Life System Management	10.63	Patch and End-of-Life System Management - 10.63	System updates on virtual machine scale sets should be installed	3.0.0
Patch and End-of-Life System Management	10.65	Patch and End-of-Life System Management - 10.65	System updates should be installed on your machines	4.0.0
Patch and End-of-Life System Management	10.65	Patch and End-of-Life System Management - 10.65	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Security of Digital Services	10.66	Security of Digital Services - 10.66	Deploy - Configure Log Analytics extension to be enabled on Windows virtual machines	3.1.0
Security of Digital Services	10.66	Security of Digital Services - 10.66	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Security of Digital Services	10.66	Security of Digital Services - 10.66	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Security of Digital Services	10.66	Security of Digital Services - 10.66	Virtual machines should have the Log Analytics extension installed	1.0.1
Data Loss Prevention (DLP)	11.15	Data Loss Prevention (DLP) - 11.15	Configure managed disks to disable public network access	2.0.0
Data Loss Prevention (DLP)	11.15	Data Loss Prevention (DLP) - 11.15	Managed disks should disable public network access	2.0.0
Data Loss Prevention (DLP)	11.15	Data Loss Prevention (DLP) - 11.15	Managed disks should use a specific set of disk encryption sets for the customer-managed key encryption	2.0.0
Security Operations Centre (SOC)	11.17	Security Operations Centre (SOC) - 11.17	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Security Operations Centre (SOC)	11.17	Security Operations Centre (SOC) - 11.17	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Cyber Risk Management	11.2	Cyber Risk Management - 11.2	Virtual machines and virtual machine scale sets should have encryption at	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			host enabled	
Security Operations Centre (SOC)	11.20	Security Operations Centre (SOC) - 11.20	Virtual machines and virtual machine scale sets should have encryption at host enabled	1.0.0
Cyber Risk Management	11.4	Cyber Risk Management - 11.4	Configure backup on virtual machines without a given tag to an existing recovery services vault in the same location	9.3.0
Cyber Risk Management	11.4	Cyber Risk Management - 11.4	Only approved VM extensions should be installed	1.0.0
Cybersecurity Operations	11.8	Cybersecurity Operations - 11.8	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Control Measures on Cybersecurity	Appendix 5.2	Control Measures on Cybersecurity - Appendix 5.2	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Control Measures on Cybersecurity	Appendix 5.2	Control Measures on Cybersecurity - Appendix 5.2	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	Internet-facing virtual machines should be protected with network security groups	3.0.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	IP Forwarding on your virtual machine should be disabled	3.0.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Control Measures on Cybersecurity	Appendix 5.7	Control Measures on Cybersecurity - Appendix 5.7	Vulnerabilities in container security configurations should be remediated	3.0.0

Spain ENS

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for Spain ENS. For more information about this compliance standard, see CCN-STIC 884.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protective Measures	mp.com.1	Protection of communications	IP Forwarding on your virtual machine should be disabled	3.0.0
Protective Measures	mp.com.1	Protection of communications	Management ports should be closed on your virtual machines	3.0.0
Protective Measures	mp.com.1	Protection of communications	Windows machines should meet requirements for 'Windows Firewall Properties'	3.0.0
Protective Measures	mp.com.2	Protection of communications	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Protective Measures	mp.com.3	Protection of communications	[Deprecated]: Virtual machines should encrypt temp disks, caches, and data flows between Compute and Storage resources	2.1.0-deprecated
Protective Measures	mp.com.3	Protection of communications	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Protective Measures	mp.com.3	Protection of communications	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Protective Measures	mp.com.4	Protection of communications	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Protective Measures	mp.info.3	Information protection	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Protective Measures	mp.info.3	Information protection	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Protective Measures	mp.info.3	Information protection	Deploy the Windows Guest Configuration extension to enable	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			Guest Configuration assignments on Windows VMs	
Protective Measures	mp.info.4	Information protection	Dependency agent should be enabled for listed virtual machine images	2.0.0
Protective Measures	mp.info.6	Information protection	Azure Backup should be enabled for Virtual Machines	3.0.0
Protective Measures	mp.info.6	Information protection	Configure backup on virtual machines with a given tag to a new recovery services vault with a default policy	9.3.0
Protective Measures	mp.info.6	Information protection	Configure backup on virtual machines without a given tag to an existing recovery services vault in the same location	9.3.0
Protective Measures	mp.s.2	Protection of services	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Protective Measures	mp.s.2	Protection of services	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Protective Measures	mp.s.2	Protection of services	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Protective Measures	mp.si.2	Protection of information media	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Protective Measures	mp.si.2	Protection of information media	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Protective Measures	mp.si.2	Protection of information media	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Protective Measures	mp.si.4	Protection of information media	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Protective Measures	mp.si.4	Protection of information media	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Protective Measures	mp.si.4	Protection of information media	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Protective Measures	mp.sw.2	Protection of IT applications	System updates should be installed on your machines	4.0.0
Operational framework	op.acc.1	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Operational framework	op.acc.1	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Operational framework	op.acc.1	Access control	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Operational framework	op.acc.1	Access control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Operational framework	op.acc.2	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Operational framework	op.acc.2	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Operational framework	op.acc.2	Access control	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Operational framework	op.acc.2	Access control	Audit Linux machines that have accounts without passwords	3.1.0
Operational framework	op.acc.2	Access control	Audit VMs that do not use managed disks	1.0.0
Operational framework	op.acc.2	Access control	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Operational framework	op.acc.2	Access control	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Operational framework	op.acc.2	Access control	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Operational framework	op.acc.2	Access control	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Operational framework	op.acc.2	Access control	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Operational framework	op.acc.2	Access control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Operational framework	op.acc.2	Access control	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Operational framework	op.acc.2	Access control	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Operational framework	op.acc.5	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Operational framework	op.acc.5	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Operational framework	op.acc.5	Access control	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Operational framework	op.acc.5	Access control	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Operational framework	op.acc.6	Access control	[Deprecated]: Virtual machines should encrypt temp disks, caches, and data flows between Compute and Storage resources	2.1.0-deprecated

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Operational framework	op.acc.6	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Operational framework	op.acc.6	Access control	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Operational framework	op.acc.6	Access control	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Operational framework	op.acc.6	Access control	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Operational framework	op.acc.6	Access control	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Operational framework	op.cont.3	Continuity of service	Azure Backup should be enabled for Virtual Machines	3.0.0
Operational framework	op.cont.3	Continuity of service	Configure backup on virtual machines with a given tag to an existing recovery services vault in the same location	9.3.0
Operational framework	op.cont.3	Continuity of service	Configure backup on virtual machines without a given tag to a new recovery services vault with a default policy	9.3.0
Operational framework	op.cont.4	Continuity of service	Azure Backup should be enabled for Virtual Machines	3.0.0
Operational framework	op.cont.4	Continuity of service	Configure backup on virtual machines with a given tag to a new recovery services vault with a default policy	9.3.0
Operational framework	op.cont.4	Continuity of service	Configure backup on virtual machines with a given tag to an existing recovery services vault in the same location	9.3.0
Operational framework	op.cont.4	Continuity of service	Configure backup on virtual machines without a given tag to a new recovery services vault with a default policy	9.3.0
Operational framework	op.cont.4	Continuity of service	Configure backup on virtual machines without a given tag to an existing recovery services vault in the same location	9.3.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Operational framework	op.exp.1	Operation	[Preview]: Configure Linux Virtual Machines to be associated with a Data Collection Rule for ChangeTracking and Inventory	1.0.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Linux VMs to install AMA for ChangeTracking and Inventory with user-assigned managed identity	1.5.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Linux VMSS to be associated with a Data Collection Rule for ChangeTracking and Inventory	1.0.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Linux VMSS to install AMA for ChangeTracking and Inventory with user-assigned managed identity	1.4.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Windows Virtual Machines to be associated with a Data Collection Rule for ChangeTracking and Inventory	1.0.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Windows VMs to install AMA for ChangeTracking and Inventory with user-assigned managed identity	1.1.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Windows VMSS to be associated with a Data Collection Rule for ChangeTracking and Inventory	1.0.0-preview
Operational framework	op.exp.1	Operation	[Preview]: Configure Windows VMSS to install AMA for ChangeTracking and Inventory with user-assigned managed identity	1.1.0-preview
Operational framework	op.exp.10	Operation	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Operational framework	op.exp.10	Operation	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Operational framework	op.exp.10	Operation	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Operational framework	op.exp.10	Operation	Deploy the Linux Guest Configuration extension to enable Guest	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			Configuration assignments on Linux VMs	
Operational framework	op.exp.2	Operation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operational framework	op.exp.2	Operation	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Operational framework	op.exp.2	Operation	Configure machines to receive a vulnerability assessment provider	4.0.0
Operational framework	op.exp.2	Operation	SQL servers on machines should have vulnerability findings resolved	1.0.0
Operational framework	op.exp.3	Operation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operational framework	op.exp.3	Operation	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Operational framework	op.exp.3	Operation	Azure Backup should be enabled for Virtual Machines	3.0.0
Operational framework	op.exp.3	Operation	Configure backup on virtual machines with a given tag to a new recovery services vault with a default policy	9.3.0
Operational framework	op.exp.3	Operation	Configure backup on virtual machines with a given tag to an existing recovery services vault in the same location	9.3.0
Operational framework	op.exp.3	Operation	Configure backup on virtual machines without a given tag to an existing recovery services vault in the same location	9.3.0
Operational framework	op.exp.3	Operation	Configure machines to receive a vulnerability assessment provider	4.0.0
Operational framework	op.exp.3	Operation	SQL servers on machines should have vulnerability findings resolved	1.0.0
Operational framework	op.exp.4	Operation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operational framework	op.exp.4	Operation	Configure machines to receive a vulnerability assessment provider	4.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Operational framework	op.exp.4	Operation	SQL servers on machines should have vulnerability findings resolved	1.0.0
Operational framework	op.exp.5	Operation	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operational framework	op.exp.5	Operation	Configure machines to receive a vulnerability assessment provider	4.0.0
Operational framework	op.exp.5	Operation	SQL servers on machines should have vulnerability findings resolved	1.0.0
Operational framework	op.exp.6	Operation	Cloud Services (extended support) role instances should have an endpoint protection solution installed	1.0.0
Operational framework	op.exp.6	Operation	Configure machines to receive a vulnerability assessment provider	4.0.0
Operational framework	op.exp.6	Operation	Configure SQL Virtual Machines to automatically install Microsoft Defender for SQL	1.4.0
Operational framework	op.exp.6	Operation	Configure SQL Virtual Machines to automatically install Microsoft Defender for SQL and DCR with a Log Analytics workspace	1.6.0
Operational framework	op.exp.6	Operation	Configure SQL Virtual Machines to automatically install Microsoft Defender for SQL and DCR with a user-defined LA workspace	1.7.0
Operational framework	op.exp.6	Operation	Configure the Microsoft Defender for SQL Log Analytics workspace	1.3.0
Operational framework	op.exp.6	Operation	Endpoint protection health issues should be resolved on your machines	1.0.0
Operational framework	op.exp.6	Operation	Endpoint protection should be installed on your machines	1.0.0
Operational framework	op.exp.6	Operation	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Operational framework	op.exp.6	Operation	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Operational framework	op.exp.6	Operation	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
Operational framework	op.exp.6	Operation	Windows machines should configure Windows Defender to update protection signatures within one day	1.0.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Operational framework	op.exp.6	Operation	Windows machines should enable Windows Defender Real-time protection	1.0.1
Operational framework	op.exp.8	Operation	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
Operational framework	op.exp.8	Operation	Dependency agent should be enabled for listed virtual machine images	2.0.0
Operational framework	op.exp.8	Operation	Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	2.0.0
Operational framework	op.exp.8	Operation	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Operational framework	op.ext.4	External resources	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Operational framework	op.ext.4	External resources	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Operational framework	op.ext.4	External resources	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Operational framework	op.ext.4	External resources	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Operational framework	op.ext.4	External resources	Audit Linux machines that have accounts without passwords	3.1.0
Operational framework	op.ext.4	External resources	Audit VMs that do not use managed disks	1.0.0
Operational framework	op.ext.4	External resources	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Operational framework	op.ext.4	External resources	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
Operational framework	op.mon.1	System monitoring	Windows Defender Exploit Guard should be enabled on your machines	2.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Operational framework	op.mon.3	System monitoring	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operational framework	op.mon.3	System monitoring	Configure machines to receive a vulnerability assessment provider	4.0.0
Operational framework	op.mon.3	System monitoring	SQL servers on machines should have vulnerability findings resolved	1.0.0
Operational framework	op.nub.1	Cloud services	Cloud Services (extended support) role instances should be configured securely	1.0.0
Operational framework	op.nub.1	Cloud services	Cloud Services (extended support) role instances should have an endpoint protection solution installed	1.0.0
Operational framework	op.nub.1	Cloud services	Cloud Services (extended support) role instances should have system updates installed	1.0.0
Operational framework	op.nub.1	Cloud services	Log Analytics agent should be installed on your Cloud Services (extended support) role instances	2.0.0
Operational framework	op.pl.2	Planning	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Organizational framework	org.4	Organizational framework	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Organizational framework	org.4	Organizational framework	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0

SWIFT CSP-CSCF v2021

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for SWIFT CSP-CSCF v2021. For more information about this compliance standard, see SWIFT CSP CSCF v2021.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
SWIFT Environment Protection	1.1	SWIFT Environment Protection	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
SWIFT Environment Protection	1.1	SWIFT Environment Protection	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
SWIFT Environment Protection	1.1	SWIFT Environment Protection	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
SWIFT Environment Protection	1.1	SWIFT Environment Protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
SWIFT Environment Protection	1.1	SWIFT Environment Protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
SWIFT Environment Protection	1.1	SWIFT Environment Protection	Internet-facing virtual machines should be protected with network security groups	3.0.0
SWIFT Environment Protection	1.1	SWIFT Environment Protection	IP Forwarding on your virtual machine should be disabled	3.0.0
SWIFT Environment Protection	1.2	Operating System Privileged Account Control	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
SWIFT Environment Protection	1.3	Virtualisation Platform Protection	Audit VMs that do not use managed disks	1.0.0
Reduce Attack Surface and Vulnerabilities	2.1	Internal Data Flow Security	Authentication to Linux machines should require SSH keys	3.2.0
Reduce Attack Surface and Vulnerabilities	2.1	Internal Data Flow Security	Windows machines should be configured to use secure communication protocols	4.1.1
Reduce Attack Surface and Vulnerabilities	2.2	Security Updates	Audit Windows VMs with a pending reboot	2.0.0
Reduce Attack Surface and Vulnerabilities	2.2	Security Updates	System updates on virtual machine scale sets should be installed	3.0.0
Reduce Attack Surface and Vulnerabilities	2.2	Security Updates	System updates should be installed on your machines	4.0.0
Reduce Attack Surface and Vulnerabilities	2.3	System Hardening	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Reduce Attack Surface and Vulnerabilities	2.3	System Hardening	Audit Windows machines that contain certificates expiring within the specified number of days	2.0.0
Reduce Attack Surface and Vulnerabilities	2.3	System Hardening	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Reduce Attack Surface and Vulnerabilities	2.3	System Hardening	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Reduce Attack Surface and Vulnerabilities	2.4A	Back-office Data Flow Security	Authentication to Linux machines should require SSH keys	3.2.0
Reduce Attack Surface and Vulnerabilities	2.4A	Back-office Data Flow Security	Windows machines should be configured to use secure communication protocols	4.1.1
Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Audit virtual machines without disaster recovery configured	1.0.0
Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Audit VMs that do not use managed disks	1.0.0
Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Azure Backup should be enabled for Virtual Machines	3.0.0
Reduce Attack Surface and Vulnerabilities	2.6	Operator Session Confidentiality and Integrity	Windows machines should be configured to use secure communication protocols	4.1.1
Reduce Attack Surface and Vulnerabilities	2.7	Vulnerability Scanning	Vulnerabilities in container security configurations should be remediated	3.0.0
Reduce Attack Surface and Vulnerabilities	2.7	Vulnerability Scanning	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Reduce Attack Surface and Vulnerabilities	2.7	Vulnerability Scanning	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Physically Secure the Environment	3.1	Physical Security	Audit VMs that do not use managed disks	1.0.0
Prevent Compromise of Credentials	4.1	Password Policy	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Prevent Compromise of Credentials	4.1	Password Policy	Audit Linux machines that have accounts without passwords	3.1.0
Prevent Compromise of Credentials	4.1	Password Policy	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Prevent Compromise of Credentials	4.1	Password Policy	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Prevent Compromise of Credentials	4.1	Password Policy	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Prevent Compromise of Credentials	4.1	Password Policy	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Prevent Compromise of Credentials	4.1	Password Policy	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Manage Identities and Segregate Privileges	5.2	Token Management	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Manage Identities and Segregate Privileges	5.4	Physical and Logical Password Storage	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
Detect Anomalous Activity to Systems or Transaction Records	6.1	Malware Protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Detect Anomalous Activity to Systems or Transaction Records	6.1	Malware Protection	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
Detect Anomalous Activity to Systems or Transaction Records	6.1	Malware Protection	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0
Detect Anomalous Activity to Systems or Transaction Records	6.1	Malware Protection	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Audit virtual machines without disaster recovery configured	1.0.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Azure Backup should be enabled for Virtual Machines	3.0.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
Detect Anomalous Activity to Systems or Transaction Records	6.4	Logging and Monitoring	Virtual machines should have the Log Analytics extension installed	1.0.1
Detect Anomalous Activity to Systems or Transaction Records	6.5A	Intrusion Detection	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Detect Anomalous Activity to Systems or Transaction Records	6.5A	Intrusion Detection	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
Detect Anomalous Activity to Systems or Transaction Records	6.5A	Intrusion Detection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0

SWIFT CSP-CSCF v2022

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for SWIFT CSP-CSCF v2022. For more information about this compliance standard, see SWIFT CSP CSCF v2022.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	Allowlist rules in your adaptive application control policy should be updated	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	Internet-facing virtual machines should be protected with network security groups	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.1	Ensure the protection of the user's local SWIFT infrastructure from potentially compromised elements of the general IT environment and external environment.	IP Forwarding on your virtual machine should be disabled	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.2	Restrict and control the allocation and usage of administrator-level operating system accounts.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.3	Secure the virtualisation platform and virtual machines (VMs) that host SWIFT-related components to the same level as physical systems.	Audit VMs that do not use managed disks	1.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.4	Control/Protect Internet access from operator PCs and systems within the secure zone.	Internet-facing virtual machines should be protected with network security groups	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.4	Control/Protect Internet access from operator PCs and systems within the secure zone.	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	Internet-facing virtual machines should be protected with network security groups	3.0.0
1. Restrict Internet Access & Protect Critical Systems from General IT Environment	1.5A	Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.	IP Forwarding on your virtual machine should be disabled	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.1	Ensure the confidentiality, integrity, and authenticity of application data flows between local SWIFT-related components.	Authentication to Linux machines should require SSH keys	3.2.0
2. Reduce Attack Surface and Vulnerabilities	2.1	Ensure the confidentiality, integrity, and authenticity of application data flows between local SWIFT-related components.	Windows machines should be configured to use secure communication protocols	4.1.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
2. Reduce Attack Surface and Vulnerabilities	2.2	Minimise the occurrence of known technical vulnerabilities on operator PCs and within the local SWIFT infrastructure by ensuring vendor support, applying mandatory software updates, and applying timely security updates aligned to the assessed risk.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
2. Reduce Attack Surface and Vulnerabilities	2.2	Minimise the occurrence of known technical vulnerabilities on operator PCs and within the local SWIFT infrastructure by ensuring vendor support, applying mandatory software updates, and applying timely security updates aligned to the assessed risk.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
2. Reduce Attack Surface and Vulnerabilities	2.2	Minimise the occurrence of known technical vulnerabilities on operator PCs and within the local SWIFT infrastructure by ensuring vendor support, applying mandatory software updates, and applying timely security updates aligned to the assessed risk.	Audit Windows VMs with a pending reboot	2.0.0
2. Reduce Attack Surface and Vulnerabilities	2.2	Minimise the occurrence of known technical vulnerabilities on operator PCs and within the local SWIFT infrastructure by ensuring vendor support, applying mandatory software updates, and applying timely security updates aligned to the assessed risk.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
2. Reduce Attack Surface and Vulnerabilities	2.2	Minimise the occurrence of known technical vulnerabilities on operator PCs and within the local SWIFT infrastructure by ensuring vendor support, applying mandatory software updates, and applying timely security updates aligned to the assessed risk.	System updates on virtual machine scale sets should be installed	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.2	Minimise the occurrence of known technical vulnerabilities on operator PCs and within the local SWIFT infrastructure by ensuring vendor support, applying mandatory software updates, and	System updates should be installed on your machines	4.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
		applying timely security updates aligned to the assessed risk.		
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Audit Windows machines that contain certificates expiring within the specified number of days	2.0.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
2. Reduce Attack Surface and Vulnerabilities	2.3	Reduce the cyber-attack surface of SWIFT-related components by performing system hardening.	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.4A	Back-office Data Flow Security	Authentication to Linux machines should require SSH keys	3.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
2. Reduce Attack Surface and Vulnerabilities	2.4A	Back-office Data Flow Security	Windows machines should be configured to use secure communication protocols	4.1.1
2. Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Audit virtual machines without disaster recovery configured	1.0.0
2. Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Audit VMs that do not use managed disks	1.0.0
2. Reduce Attack Surface and Vulnerabilities	2.5A	External Transmission Data Protection	Azure Backup should be enabled for Virtual Machines	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.6	Protect the confidentiality and integrity of interactive operator sessions that connect to the local or remote (operated by a service provider) SWIFT infrastructure or service provider SWIFT-related applications	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
2. Reduce Attack Surface and Vulnerabilities	2.6	Protect the confidentiality and integrity of interactive operator sessions that connect to the local or remote (operated by a service provider) SWIFT infrastructure or service provider SWIFT-related applications	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
2. Reduce Attack Surface and Vulnerabilities	2.6	Protect the confidentiality and integrity of interactive operator sessions that connect to the local or remote (operated by a service provider) SWIFT infrastructure or service provider SWIFT-related applications	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
2. Reduce Attack Surface and Vulnerabilities	2.6	Protect the confidentiality and integrity of interactive operator sessions that connect to the local or remote (operated by a service provider) SWIFT infrastructure or service provider SWIFT-related applications	Windows machines should be configured to use secure communication protocols	4.1.1

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
2. Reduce Attack Surface and Vulnerabilities	2.6	Protect the confidentiality and integrity of interactive operator sessions that connect to the local or remote (operated by a service provider) SWIFT infrastructure or service provider SWIFT-related applications	Windows machines should meet requirements for 'Security Options - Interactive Logon'	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.7	Identify known vulnerabilities within the local SWIFT environment by implementing a regular vulnerability scanning process and act upon results.	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.7	Identify known vulnerabilities within the local SWIFT environment by implementing a regular vulnerability scanning process and act upon results.	Vulnerabilities in container security configurations should be remediated	3.0.0
2. Reduce Attack Surface and Vulnerabilities	2.7	Identify known vulnerabilities within the local SWIFT environment by implementing a regular vulnerability scanning process and act upon results.	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
2. Reduce Attack Surface and Vulnerabilities	2.7	Identify known vulnerabilities within the local SWIFT environment by implementing a regular vulnerability scanning process and act upon results.	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
3. Physically Secure the Environment	3.1	Prevent unauthorised physical access to sensitive equipment, workplace environments, hosting sites, and storage.	Audit VMs that do not use managed disks	1.0.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
4. Prevent Compromise of	4.1	Ensure passwords are sufficiently resistant against common	Audit Linux machines that allow remote connections	3.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Credentials		password attacks by implementing and enforcing an effective password policy.	from accounts without passwords	
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Audit Linux machines that have accounts without passwords	3.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
4. Prevent Compromise of Credentials	4.1	Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
5. Manage Identities and Segregate Privileges	5.1	Enforce the security principles of need-to-know access, least privilege, and separation of duties for operator accounts.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
5. Manage Identities and Segregate Privileges	5.1	Enforce the security principles of need-to-know access, least privilege, and separation of duties for operator accounts.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
5. Manage Identities and Segregate Privileges	5.1	Enforce the security principles of need-to-know access, least privilege, and separation of duties for operator accounts.	Audit Windows machines that contain certificates expiring within the specified number of days	2.0.0
5. Manage Identities and Segregate Privileges	5.1	Enforce the security principles of need-to-know access, least privilege, and separation of duties for operator accounts.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
5. Manage Identities and Segregate Privileges	5.2	Ensure the proper management, tracking, and use of connected and disconnected hardware authentication or personal tokens (when tokens are used).	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
5. Manage Identities and Segregate Privileges	5.4	Protect physically and logically the repository of recorded passwords.	Audit Windows machines that do not store passwords using reversible encryption	2.0.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.1	Ensure that local SWIFT infrastructure is protected against malware and act upon results.	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.1	Ensure that local SWIFT infrastructure is protected against malware and act upon results.	Microsoft Antimalware for Azure should be configured to automatically update protection signatures	1.0.0
6. Detect Anomalous Activity to Systems or	6.1	Ensure that local SWIFT infrastructure is protected against malware and act upon results.	Microsoft IaaSAntimalware extension should be deployed on Windows servers	1.1.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Transaction Records				
6. Detect Anomalous Activity to Systems or Transaction Records	6.1	Ensure that local SWIFT infrastructure is protected against malware and act upon results.	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	2.0.1-preview
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Audit virtual machines without disaster recovery configured	1.0.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Azure Backup should be enabled for Virtual Machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	2.0.1
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	The Log Analytics extension should be installed on Virtual Machine Scale Sets	1.0.1
6. Detect Anomalous Activity to Systems or Transaction Records	6.4	Record security events and detect anomalous actions and operations within the local SWIFT environment.	Virtual machines should have the Log Analytics extension installed	1.0.1
6. Detect Anomalous Activity to Systems or Transaction Records	6.5A	Detect and contain anomalous network activity into and within the local or remote SWIFT environment.	[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	1.0.2-preview
6. Detect Anomalous Activity to Systems or Transaction Records	6.5A	Detect and contain anomalous network activity into and within the local or remote SWIFT environment.	[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	1.0.2-preview
6. Detect Anomalous Activity to Systems or Transaction Records	6.5A	Detect and contain anomalous network activity into and within the local or remote SWIFT environment.	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0

System and Organization Controls (SOC) 2

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance details for System and Organization Controls (SOC) 2. For more information about this compliance standard, see System and Organization Controls (SOC) 2.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Additional Criteria For Availability	A1.2	Environmental protections, software, data back-up processes, and recovery infrastructure	Azure Backup should be enabled for Virtual Machines	3.0.0
Risk Assessment	CC3.2	COSO Principle 7	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Authentication to Linux machines should require SSH keys	3.2.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Internet-facing virtual machines should be protected with network security groups	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Management ports should be closed on your virtual machines	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Logical and Physical Access Controls	CC6.1	Logical access security software, infrastructure, and architectures	Windows machines should be configured to use secure communication protocols	4.1.1
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	All network ports should be restricted on network security	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			groups associated to your virtual machine	
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Authentication to Linux machines should require SSH keys	3.2.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Internet-facing virtual machines should be protected with network security groups	3.0.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	IP Forwarding on your virtual machine should be disabled	3.0.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Management ports should be closed on your virtual machines	3.0.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Logical and Physical Access Controls	CC6.6	Security measures against threats outside system boundaries	Windows machines should be configured to use secure communication protocols	4.1.1
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	Internet-facing virtual machines should be protected with network security groups	3.0.0
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	Management ports should be closed on your virtual machines	3.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	Non-internet-facing virtual machines should be protected with network security groups	3.0.0
Logical and Physical Access Controls	CC6.7	Restrict the movement of information to authorized users	Windows machines should be configured to use secure communication protocols	4.1.1
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	6.0.0-preview
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines scale sets	5.1.0-preview
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	4.0.0-preview
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	3.1.0-preview
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Allowlist rules in your adaptive application control policy should be updated	3.0.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Audit VMs that do not use managed disks	1.0.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Endpoint protection health issues should be resolved on your machines	1.0.0
Logical and Physical Access	CC6.8	Prevent or detect against unauthorized or malicious	Endpoint protection should be installed on your machines	1.0.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Controls		software		
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Guest Configuration extension should be installed on your machines	1.0.3
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Only approved VM extensions should be installed	1.0.0
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Logical and Physical Access Controls	CC6.8	Prevent or detect against unauthorized or malicious software	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
System Operations	CC7.1	Detection and monitoring of new vulnerabilities	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
System Operations	CC7.1	Detection and monitoring of new vulnerabilities	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
System Operations	CC7.1	Detection and monitoring of new vulnerabilities	Allowlist rules in your adaptive application control policy should be updated	3.0.0
System Operations	CC7.2	Monitor system components for anomalous behavior	Windows Defender Exploit Guard should be enabled on your machines	2.0.0
Change Management	CC8.1	Changes to infrastructure, data, and software	[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	6.0.0-preview
Change Management	CC8.1	Changes to infrastructure, data, and software	[Preview]: Guest Attestation extension should be installed on	5.1.0-preview

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
			supported Linux virtual machines scale sets	
Change Management	CC8.1	Changes to infrastructure, data, and software	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	4.0.0-preview
Change Management	CC8.1	Changes to infrastructure, data, and software	[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	3.1.0-preview
Change Management	CC8.1	Changes to infrastructure, data, and software	[Preview]: Secure Boot should be enabled on supported Windows virtual machines	4.0.0-preview
Change Management	CC8.1	Changes to infrastructure, data, and software	[Preview]: vTPM should be enabled on supported virtual machines	2.0.0-preview
Change Management	CC8.1	Changes to infrastructure, data, and software	Audit VMs that do not use managed disks	1.0.0
Change Management	CC8.1	Changes to infrastructure, data, and software	Guest Configuration extension should be installed on your machines	1.0.3
Change Management	CC8.1	Changes to infrastructure, data, and software	Linux machines should meet requirements for the Azure compute security baseline	2.2.0
Change Management	CC8.1	Changes to infrastructure, data, and software	Only approved VM extensions should be installed	1.0.0
Change Management	CC8.1	Changes to infrastructure, data, and software	Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	1.0.1
Change Management	CC8.1	Changes to infrastructure, data, and software	Windows machines should meet requirements of the Azure compute security baseline	2.0.0
Additional Criteria For Processing Integrity	PI1.5	Store inputs and outputs completely, accurately, and timely	Azure Backup should be enabled for Virtual Machines	3.0.0

UK OFFICIAL and UK NHS

To review how the available Azure Policy built-ins for all Azure services map to this compliance standard, see Azure Policy Regulatory Compliance - UK OFFICIAL and UK NHS. For more information

about this compliance standard, see UK OFFICIAL.

Expand table

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Data in transit protection	1	Data in transit protection	Windows machines should be configured to use secure communication protocols	4.1.1
Identity and authentication	10	Identity and authentication	Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	4.1.0
Identity and authentication	10	Identity and authentication	Add system-assigned managed identity to enable Guest Configuration assignments on VMs with a user-assigned identity	4.1.0
Identity and authentication	10	Identity and authentication	Audit Linux machines that allow remote connections from accounts without passwords	3.1.0
Identity and authentication	10	Identity and authentication	Audit Linux machines that do not have the passwd file permissions set to 0644	3.1.0
Identity and authentication	10	Identity and authentication	Audit Linux machines that have accounts without passwords	3.1.0
Identity and authentication	10	Identity and authentication	Audit VMs that do not use managed disks	1.0.0
Identity and authentication	10	Identity and authentication	Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	2.1.0
Identity and authentication	10	Identity and authentication	Audit Windows machines that do not have the maximum password age set to specified number of days	2.1.0
Identity and authentication	10	Identity and authentication	Audit Windows machines that do not have the minimum password age set to specified number of days	2.1.0
Identity and authentication	10	Identity and authentication	Audit Windows machines that do not have the password complexity setting enabled	2.0.0
Identity and authentication	10	Identity and authentication	Audit Windows machines that do not restrict the minimum password length to specified number of characters	2.1.0
Identity and authentication	10	Identity and authentication	Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	3.1.0
Identity and authentication	10	Identity and authentication	Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	1.2.0

Domain	Control ID	Control title	Policy (Azure portal)	Policy version (GitHub)
Identity and authentication	10	Identity and authentication	Virtual machines should be migrated to new Azure Resource Manager resources	1.0.0
External interface protection	11	External interface protection	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
External interface protection	11	External interface protection	Adaptive network hardening recommendations should be applied on internet facing virtual machines	3.0.0
External interface protection	11	External interface protection	All network ports should be restricted on network security groups associated to your virtual machine	3.0.0
External interface protection	11	External interface protection	Endpoint protection solution should be installed on virtual machine scale sets	3.0.0
External interface protection	11	External interface protection	Management ports of virtual machines should be protected with just-in-time network access control	3.0.0
Operational security	5.2	Vulnerability management	A vulnerability assessment solution should be enabled on your virtual machines	3.0.0
Operational security	5.2	Vulnerability management	Monitor missing Endpoint Protection in Azure Security Center	3.0.0
Operational security	5.2	Vulnerability management	System updates on virtual machine scale sets should be installed	3.0.0
Operational security	5.2	Vulnerability management	System updates should be installed on your machines	4.0.0
Operational security	5.2	Vulnerability management	Vulnerabilities in security configuration on your machines should be remediated	3.1.0
Operational security	5.2	Vulnerability management	Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	3.0.0
Operational security	5.3	Protective Monitoring	Adaptive application controls for defining safe applications should be enabled on your machines	3.0.0
Operational security	5.3	Protective Monitoring	Audit virtual machines without disaster recovery configured	1.0.0

Next steps

- Learn more about Azure Policy Regulatory Compliance.
- See the built-ins on the Azure Policy GitHub repo.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Policy built-in definitions for Azure Virtual Machines

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This page is an index of Azure Policy built-in policy definitions for Azure Virtual Machines. For additional Azure Policy built-ins for other services, see Azure Policy built-in definitions.

The name of each built-in policy definition links to the policy definition in the Azure portal. Use the link in the **Version** column to view the source on the Azure Policy GitHub repo.

Microsoft.Compute

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: A managed identity should be enabled on your machines	Resources managed by Automanage should have a managed identity.	Audit, Disabled	1.0.0-preview
[Preview]: Add user-assigned managed identity to enable Guest Configuration assignments on virtual machines	This policy adds a user-assigned managed identity to virtual machines hosted in Azure that are supported by Guest Configuration. A user-assigned managed identity is a prerequisite for all Guest Configuration assignments and must be added to machines before using any Guest Configuration policy definitions. For more information on Guest Configuration, visit https://aka.ms/gcpol.	AuditIfNotExists, DeployIfNotExists, Disabled	2.1.0-preview
[Preview]: Assign Built-In User-Assigned Managed Identity to Virtual Machine Scale Sets	Create and assign a built-in user-assigned managed identity or assign a pre-created user-assigned managed identity at scale to virtual machine scale sets. For more detailed documentation, visit aka.ms/managedidentitypolicy.	AuditIfNotExists, DeployIfNotExists, Disabled	1.1.0-preview
[Preview]: Assign Built-In User-Assigned Managed Identity to Virtual Machines	Create and assign a built-in user-assigned managed identity or assign a pre-created user-assigned managed identity at scale to virtual machines. For more detailed documentation, visit aka.ms/managedidentitypolicy.	AuditIfNotExists, DeployIfNotExists, Disabled	1.1.0-preview
[Preview]: Audit SSH Posture Control on Linux machines	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if SSH Server is not securely configured on the Linux machines.	AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Automanage Configuration Profile Assignment should be Conformant	Resources managed by Automanage should have a status of Conformant or ConformantCorrected.	AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Azure Backup should be enabled for Managed Disks	Ensure protection of your Managed Disks by enabling Azure Backup. Azure Backup is a secure and cost effective data protection solution for Azure.	AuditIfNotExists, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Azure Security agent should be installed on your Linux virtual machine scale sets	Install the Azure Security agent on your Linux virtual machine scale sets in order to monitor your machines for security configurations and vulnerabilities. Results of the assessments can seen and managed in Azure Security Center.	AuditIfNotExists, Disabled	2.0.0-preview
[Preview]: Azure Security agent should be installed on your Linux virtual machines	Install the Azure Security agent on your Linux virtual machines in order to monitor your machines for security configurations and vulnerabilities. Results of the assessments can seen and managed in Azure Security Center.	AuditIfNotExists, Disabled	2.0.0-preview
[Preview]: Azure Security agent should be installed on your Windows virtual machine scale sets	Install the Azure Security agent on your Windows virtual machine scale sets in order to monitor your machines for security configurations and vulnerabilities. Results of the assessments can seen and managed in Azure Security Center.	AuditIfNotExists, Disabled	2.1.0-preview
[Preview]: Azure Security agent should be installed on your Windows virtual machines	Install the Azure Security agent on your Windows virtual machines in order to monitor your machines for security configurations and vulnerabilities. Results of the assessments can seen and managed in Azure Security Center.	AuditIfNotExists, Disabled	2.1.0-preview
[Preview]: Boot Diagnostics should be enabled on virtual machines	Azure virtual machines should have boot diagniostics enabled.	Audit, Disabled	1.0.0-preview
[Preview]: ChangeTracking extension should be installed on your Linux virtual machine	Install ChangeTracking Extension on Linux virtual machines to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitoring Agent.	AuditIfNotExists, Disabled	2.0.0-preview
[Preview]: ChangeTracking extension should be installed on your Linux virtual machine scale sets	Install ChangeTracking Extension on Linux virtual machine scale sets to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitoring Agent.	AuditIfNotExists, Disabled	2.0.0-preview
[Preview]: ChangeTracking extension should be installed on your Windows virtual machine	Install ChangeTracking Extension on Windows virtual machines to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitoring Agent.	AuditIfNotExists, Disabled	2.0.0-preview
[Preview]: ChangeTracking extension should be installed on your Windows virtual machine scale sets	Install ChangeTracking Extension on Windows virtual machine scale sets to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitoring Agent.	AuditIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure Azure Defender for	Configure Windows machines to automatically install the Azure Defender for SQL agent where the Azure Monitor Agent is installed. Security Center collects	DeployIfNotExists, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
SQL agent on virtual machine	events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Creates a resource group and Log Analytics workspace in the same region as the machine. Target virtual machines must be in a supported location.		
[Preview]: Configure backup for Azure Disks (Managed Disks) with a given tag to an existing backup vault in the same region	Enforce backup for all Azure Disks (Managed Disks) that contain a given tag to a central backup vault. Learn more at https://aka.ms/AB-DiskBackupAzPolicies	DeployIfNotExists, AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure backup for Azure Disks (Managed Disks) without a given tag to an existing backup vault in the same region	Enforce backup for all Azure Disks (Managed Disks) that do not contain a given tag to a central backup vault. Learn more at https://aka.ms/AB-DiskBackupAzPolicies	DeployIfNotExists, AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure ChangeTracking Extension for Linux virtual machine scale sets	Configure Linux virtual machine scale sets to automatically install the ChangeTracking Extension to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitor Agent.	DeployIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure ChangeTracking Extension for Linux virtual machines	Configure Linux virtual machines to automatically install the ChangeTracking Extension to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitor Agent.	DeployIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure ChangeTracking Extension for Windows virtual machine scale sets	Configure Windows virtual machine scale sets to automatically install the ChangeTracking Extension to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitor Agent.	DeployIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure ChangeTracking Extension for Windows virtual machines	Configure Windows virtual machines to automatically install the ChangeTracking Extension to enable File Integrity Monitoring(FIM) in Azure Security Center. FIM examines operating system files, Windows registries, application software, Linux system files, and more, for changes that might indicate an attack. The extension can be installed in virtual machines and locations supported by Azure Monitor Agent.	DeployIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure Linux Virtual Machines to be associated with a Data Collection Rule for ChangeTracking and Inventory	Deploy Association to link Linux virtual machines to the specified Data Collection Rule to enable ChangeTracking and Inventory. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure Linux VMs to install AMA for	Automate the deployment of Azure Monitor Agent extension on your Linux virtual machines for enabling ChangeTracking and Inventory. This policy will install the extension and configure it to use the specified user-assigned	DeployIfNotExists, Disabled	1.5.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
ChangeTracking and Inventory with user-assigned managed identity	managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.		
[Preview]: Configure Linux VMSS to be associated with a Data Collection Rule for ChangeTracking and Inventory	Deploy Association to link Linux virtual machine scale sets to the specified Data Collection Rule to enable ChangeTracking and Inventory. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure Linux VMSS to install AMA for ChangeTracking and Inventory with user-assigned managed identity	Automate the deployment of Azure Monitor Agent extension on your Linux virtual machine scale sets for enabling ChangeTracking and Inventory. This policy will install the extension and configure it to use the specified user-assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	1.4.0-preview
[Preview]: Configure SSH Posture Control on Linux machines	This policy creates a Guest Configuration assignment to set SSH Posture Control on Linux machines.	DeployIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure supported Linux virtual machine scale sets to automatically install the Azure Security agent	Configure supported Linux virtual machine scale sets to automatically install the Azure Security agent. Security Center collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Target virtual machines must be in a supported location.	DeployIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure supported Linux virtual machine scale sets to automatically install the Guest Attestation extension	Configure supported Linux virtual machines scale sets to automatically install the Guest Attestation extension to allow Azure Security Center to proactively attest and monitor the boot integrity. Boot integrity is attested via Remote Attestation.	DeployIfNotExists, Disabled	6.1.0-preview
[Preview]: Configure supported Linux virtual machines to automatically enable Secure Boot	Configure supported Linux virtual machines to automatically enable Secure Boot to mitigate against malicious and unauthorized changes to the boot chain. Once enabled, only trusted bootloaders, kernel and kernel drivers will be allowed to run.	DeployIfNotExists, Disabled	5.0.0-preview
[Preview]: Configure supported Linux virtual machines to automatically install the Azure Security agent	Configure supported Linux virtual machines to automatically install the Azure Security agent. Security Center collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Target virtual machines must be in a supported location.	DeployIfNotExists, Disabled	7.0.0-preview
[Preview]: Configure supported Linux virtual machines to automatically install the Guest Attestation extension	Configure supported Linux virtual machines to automatically install the Guest Attestation extension to allow Azure Security Center to proactively attest and monitor the boot integrity. Boot integrity is attested via Remote Attestation.	DeployIfNotExists, Disabled	7.1.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Configure supported virtual machines to automatically enable vTPM	Configure supported virtual machines to automatically enable vTPM to facilitate Measured Boot and other OS security features that require a TPM. Once enabled, vTPM can be used to attest boot integrity.	DeployIfNotExists, Disabled	2.0.0-preview
[Preview]: Configure supported Windows machines to automatically install the Azure Security agent	Configure supported Windows machines to automatically install the Azure Security agent. Security Center collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Target virtual machines must be in a supported location.	DeployIfNotExists, Disabled	5.1.0-preview
[Preview]: Configure supported Windows virtual machine scale sets to automatically install the Azure Security agent	Configure supported Windows virtual machine scale sets to automatically install the Azure Security agent. Security Center collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Target Windows virtual machine scale sets must be in a supported location.	DeployIfNotExists, Disabled	2.1.0-preview
[Preview]: Configure supported Windows virtual machine scale sets to automatically install the Guest Attestation extension	Configure supported Windows virtual machines scale sets to automatically install the Guest Attestation extension to allow Azure Security Center to proactively attest and monitor the boot integrity. Boot integrity is attested via Remote Attestation.	DeployIfNotExists, Disabled	4.1.0-preview
[Preview]: Configure supported Windows virtual machines to automatically enable Secure Boot	Configure supported Windows virtual machines to automatically enable Secure Boot to mitigate against malicious and unauthorized changes to the boot chain. Once enabled, only trusted bootloaders, kernel and kernel drivers will be allowed to run.	DeployIfNotExists, Disabled	3.0.0-preview
[Preview]: Configure supported Windows virtual machines to automatically install the Guest Attestation extension	Configure supported Windows virtual machines to automatically install the Guest Attestation extension to allow Azure Security Center to proactively attest and monitor the boot integrity. Boot integrity is attested via Remote Attestation.	DeployIfNotExists, Disabled	5.1.0-preview
[Preview]: Configure system-assigned managed identity to enable Azure Monitor assignments on VMs	Configure system-assigned managed identity to virtual machines hosted in Azure that are supported by Azure Monitor and do not have a system-assigned managed identity. A system-assigned managed identity is a prerequisite for all Azure Monitor assignments and must be added to machines before using any Azure Monitor extension. Target virtual machines must be in a supported location.	Modify, Disabled	6.0.0-preview
[Preview]: Configure VMs created with Shared Image Gallery images to install the Guest Attestation extension	Configure virtual machines created with Shared Image Gallery images to automatically install the Guest Attestation extension to allow Azure Security Center to proactively attest and monitor the boot integrity. Boot integrity is attested via Remote Attestation.	DeployIfNotExists, Disabled	2.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Configure VMSS created with Shared Image Gallery images to install the Guest Attestation extension	Configure VMSS created with Shared Image Gallery images to automatically install the Guest Attestation extension to allow Azure Security Center to proactively attest and monitor the boot integrity. Boot integrity is attested via Remote Attestation.	DeployIfNotExists, Disabled	2.1.0-preview
[Preview]: Configure Windows Server to disable local users.	Creates a Guest Configuration assignment to configure disabling local users on Windows Server. This ensures that Windows Servers can only be accessed by AAD (Azure Active Directory) account or a list of explicitly allowed users by this policy, improving overall security posture.	DeployIfNotExists, Disabled	1.2.0-preview
[Preview]: Configure Windows Virtual Machines to be associated with a Data Collection Rule for ChangeTracking and Inventory	Deploy Association to link Windows virtual machines to specified Data Collection Rule to enable ChangeTracking and Inventory. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure Windows VMs to install AMA for ChangeTracking and Inventory with user-assigned managed identity	Automate the deployment of Azure Monitor Agent extension on your Windows virtual machines for enabling ChangeTracking and Inventory. This policy will install the extension and configure it to use the specified user-assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	1.1.0-preview
[Preview]: Configure Windows VMSS to be associated with a Data Collection Rule for ChangeTracking and Inventory	Deploy Association to link Windows virtual machine scale sets to specified Data Collection Rule to enable ChangeTracking and Inventory. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	1.0.0-preview
[Preview]: Configure Windows VMSS to install AMA for ChangeTracking and Inventory with user-assigned managed identity	Automate the deployment of Azure Monitor Agent extension on your Windows virtual machine scale sets for enabling ChangeTracking and Inventory. This policy will install the extension and configure it to use the specified user-assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	1.1.0-preview
[Preview]: Deploy Microsoft Defender for Endpoint agent on Linux virtual machines	Deploys Microsoft Defender for Endpoint agent on applicable Linux VM images.	DeployIfNotExists, AuditIfNotExists, Disabled	3.0.0-preview
[Preview]: Deploy Microsoft Defender for Endpoint agent on Windows virtual machines	Deploys Microsoft Defender for Endpoint on applicable Windows VM images.	DeployIfNotExists, AuditIfNotExists, Disabled	2.0.1-preview
[Preview]: Enable system-assigned identity to SQL VM	Enable system-assigned identity at scale to SQL virtual machines. You need to assign this policy at subscription level. Assign at resource group level will not work as expected.	DeployIfNotExists, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines	Install Guest Attestation extension on supported Linux virtual machines to allow Azure Security Center to proactively attest and monitor the boot integrity. Once installed, boot integrity will be attested via Remote Attestation. This assessment applies to Trusted Launch and Confidential Linux virtual machines.	AuditIfNotExists, Disabled	6.0.0-preview
[Preview]: Guest Attestation extension should be installed on supported Linux virtual machines scale sets	Install Guest Attestation extension on supported Linux virtual machines scale sets to allow Azure Security Center to proactively attest and monitor the boot integrity. Once installed, boot integrity will be attested via Remote Attestation. This assessment applies to Trusted Launch and Confidential Linux virtual machine scale sets.	AuditIfNotExists, Disabled	5.1.0-preview
[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines	Install Guest Attestation extension on supported virtual machines to allow Azure Security Center to proactively attest and monitor the boot integrity. Once installed, boot integrity will be attested via Remote Attestation. This assessment applies to Trusted Launch and Confidential Windows virtual machines.	AuditIfNotExists, Disabled	4.0.0-preview
[Preview]: Guest Attestation extension should be installed on supported Windows virtual machines scale sets	Install Guest Attestation extension on supported virtual machines scale sets to allow Azure Security Center to proactively attest and monitor the boot integrity. Once installed, boot integrity will be attested via Remote Attestation. This assessment applies to Trusted Launch and Confidential Windows virtual machine scale sets.	AuditIfNotExists, Disabled	3.1.0-preview
[Preview]: Linux machines should meet requirements for the Azure security baseline for Docker hosts	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. The machine is not configured correctly for one of the recommendations in the Azure security baseline for Docker hosts.	AuditIfNotExists, Disabled	1.2.0-preview
[Preview]: Linux machines should meet STIG compliance requirement for Azure compute	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the machine is not configured correctly for one of the recommendations in STIG compliance requirement for Azure compute. DISA (Defense Information Systems Agency) provides technical guides STIG (Security Technical Implementation Guide) to secure compute OS as required by Department of Defense (DoD). For more details, https://public.cyber.mil/stigs/.	AuditIfNotExists, Disabled	1.2.0-preview
[Preview]: Linux machines with OMI installed should have version 1.6.8-1 or later	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Due to a security fix included in version 1.6.8-1 of the OMI package for Linux, all machines should be updated to the latest release. Upgrade apps/packages that use OMI to resolve the issue. For more information, see https://aka.ms/omiguidance.	AuditIfNotExists, Disabled	1.2.0-preview
[Preview]: Linux virtual machines should use only signed and trusted boot components	All OS boot components (boot loader, kernel, kernel drivers) must be signed by trusted publishers. Defender for Cloud has identified untrusted OS boot components on one or more of your Linux machines. To protect your machines from potentially malicious components, add them to your allow list or remove the identified components.	AuditIfNotExists, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Linux virtual machines should use Secure Boot	To protect against the installation of malware-based rootkits and boot kits, enable Secure Boot on supported Linux virtual machines. Secure Boot ensures that only signed operating systems and drivers will be allowed to run. This assessment only applies to Linux virtual machines that have the Azure Monitor Agent installed.	AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Log Analytics Extension should be enabled for listed virtual machine images	Reports virtual machines as non-compliant if the virtual machine image is not in the list defined and the extension is not installed.	AuditIfNotExists, Disabled	2.0.1-preview
[Preview]: Machines should have ports closed that might expose attack vectors	Azure's Terms Of Use prohibit the use of Azure services in ways that could damage, disable, overburden, or impair any Microsoft server, or the network. The exposed ports identified by this recommendation need to be closed for your continued security. For each identified port, the recommendation also provides an explanation of the potential threat.	AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Managed Disks should be Zone Resilient	Managed Disks can be configured to be either Zone Aligned, Zone Redundant, or neither. Managed Disks with exactly one zone assignment are Zone Aligned. Managed Disks with a sku name that ends in ZRS are Zone Redundant. This policy assists in identifying and enforcing these resilience configurations for Managed Disks.	Audit, Deny, Disabled	1.0.0-preview
[Preview]: Network traffic data collection agent should be installed on Linux virtual machines	Security Center uses the Microsoft Dependency agent to collect network traffic data from your Azure virtual machines to enable advanced network protection features such as traffic visualization on the network map, network hardening recommendations and specific network threats.	AuditIfNotExists, Disabled	1.0.2-preview
[Preview]: Network traffic data collection agent should be installed on Windows virtual machines	Security Center uses the Microsoft Dependency agent to collect network traffic data from your Azure virtual machines to enable advanced network protection features such as traffic visualization on the network map, network hardening recommendations and specific network threats.	AuditIfNotExists, Disabled	1.0.2-preview
[Preview]: Secure Boot should be enabled on supported Windows virtual machines	Enable Secure Boot on supported Windows virtual machines to mitigate against malicious and unauthorized changes to the boot chain. Once enabled, only trusted bootloaders, kernel and kernel drivers will be allowed to run. This assessment applies to Trusted Launch and Confidential Windows virtual machines.	Audit, Disabled	4.0.0-preview
[Preview]: Set prerequisite for Scheduling recurring updates on Azure virtual machines.	This policy will set the prerequisite needed to schedule recurring updates on Azure Update Manager by configuring patch orchestration to 'Customer Managed Schedules'. This change will automatically set the patch mode to 'AutomaticByPlatform' and enables 'BypassPlatformSafetyChecksOnUserSchedule' to 'True' on Azure VMs. The prerequisite is not applicable for Arc-enabled servers. Learn more - https://learn.microsoft.com/en-us/azure/update-manager/dynamic-scope-overview?tabs=avms#prerequisites	DeployIfNotExists, Disabled	1.1.0-preview
[Preview]: System updates should be installed on your machines (powered by Update Center)	Your machines are missing system, security, and critical updates. Software updates often include critical patches to security holes. Such holes are frequently exploited in malware attacks so it's vital to keep your software updated. To install all outstanding patches and secure your machines, follow the remediation steps.	AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Virtual Machine Scale Sets	Virtual Machine Scale Sets can be configured to be either Zone Aligned, Zone Redundant, or neither. Virtual Machine Scale Sets that have exactly one entry in	Audit, Deny, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
should be Zone Resilient	their zones array are considered Zone Aligned. In contrast, Virtual Machine Scale Sets with 3 or more entries in their zones array and a capacity of at least 3 are recognized as Zone Redundant. This policy helps identify and enforce these resilience configurations.		
[Preview]: Virtual machines guest attestation status should be healthy	Guest attestation is performed by sending a trusted log (TCGLog) to an attestation server. The server uses these logs to determine whether boot components are trustworthy. This assessment is intended to detect compromises of the boot chain which might be the result of a bootkit or rootkit infection. This assessment only applies to Trusted Launch enabled virtual machines that have Guest Attestation extension installed.	AuditIfNotExists, Disabled	1.0.0-preview
[Preview]: Virtual Machines should be Zone Aligned	Virtual Machines can be configured to be Zone Aligned or not. They are considered Zone Aligned if they have only one entry in their zones array. This policy ensures that they are configured to operate within a single availability zone.	Audit, Deny, Disabled	1.0.0-preview
[Preview]: vTPM should be enabled on supported virtual machines	Enable virtual TPM device on supported virtual machines to facilitate Measured Boot and other OS security features that require a TPM. Once enabled, vTPM can be used to attest boot integrity. This assessment only applies to trusted launch enabled virtual machines.	Audit, Disabled	2.0.0-preview
[Preview]: Windows machines should meet STIG compliance requirements for Azure compute	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the machine is not configured correctly for one of the recommendations in STIG compliance requirements for Azure compute. DISA (Defense Information Systems Agency) provides technical guides STIG (Security Technical Implementation Guide) to secure compute OS as required by Department of Defense (DoD). For more details, https://public.cyber.mil/stigs/.	AuditIfNotExists, Disabled	1.0.0-preview
A vulnerability assessment solution should be enabled on your virtual machines	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.	AuditIfNotExists, Disabled	3.0.0
Adaptive application controls for defining safe applications should be enabled on your machines	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This helps harden your machines against malware. To simplify the process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.	AuditIfNotExists, Disabled	3.0.0
Adaptive network hardening recommendations should be applied on internet facing virtual machines	Azure Security Center analyzes the traffic patterns of Internet facing virtual machines and provides Network Security Group rule recommendations that reduce the potential attack surface	AuditIfNotExists, Disabled	3.0.0
Add system-assigned managed identity to enable Guest Configuration assignments on virtual machines with no identities	This policy adds a system-assigned managed identity to virtual machines hosted in Azure that are supported by Guest Configuration but do not have any managed identities. A system-assigned managed identity is a prerequisite for all Guest Configuration assignments and must be added to machines before using any Guest Configuration policy definitions. For more information on Guest Configuration, visit https://aka.ms/gcpol.	modify	4.1.0
Add system-assigned managed	This policy adds a system-assigned managed identity to virtual machines hosted in Azure that are supported by Guest Configuration and have at least	modify	4.1.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
identity to enable Guest Configuration assignments on VMs with a user-assigned identity	one user-assigned identity but do not have a system-assigned managed identity. A system-assigned managed identity is a prerequisite for all Guest Configuration assignments and must be added to machines before using any Guest Configuration policy definitions. For more information on Guest Configuration, visit https://aka.ms/gcpol.		
All network ports should be restricted on network security groups associated to your virtual machine	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0
Allowed virtual machine size SKUs	This policy enables you to specify a set of virtual machine size SKUs that your organization can deploy.	Deny	1.0.1
Allowlist rules in your adaptive application control policy should be updated	Monitor for changes in behavior on groups of machines configured for auditing by Azure Security Center's adaptive application controls. Security Center uses machine learning to analyze the running processes on your machines and suggest a list of known-safe applications. These are presented as recommended apps to allow in adaptive application control policies.	AuditIfNotExists, Disabled	3.0.0
Audit Linux machines that allow remote connections from accounts without passwords	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Linux machines that allow remote connections from accounts without passwords	AuditIfNotExists, Disabled	3.1.0
Audit Linux machines that do not have the passwd file permissions set to 0644	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Linux machines that do not have the passwd file permissions set to 0644	AuditIfNotExists, Disabled	3.1.0
Audit Linux machines that don't have the specified applications installed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the Chef InSpec resource indicates that one or more of the packages provided by the parameter are not installed.	AuditIfNotExists, Disabled	4.2.0
Audit Linux machines that have accounts without passwords	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Linux machines that have accounts without passwords	AuditIfNotExists, Disabled	3.1.0
Audit Linux machines that have the specified applications installed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the Chef InSpec resource indicates that one or more of the packages provided by the parameter are installed.	AuditIfNotExists, Disabled	4.2.0
Audit virtual machines without disaster recovery configured	Audit virtual machines which do not have disaster recovery configured. To learn more about disaster recovery, visit https://aka.ms/asr-doc.	auditIfNotExists	1.0.0
Audit VMs that do not use managed disks	This policy audits VMs that do not use managed disks	audit	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Audit Windows machines missing any of specified members in the Administrators group	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the local Administrators group does not contain one or more members that are listed in the policy parameter.	auditIfNotExists	2.0.0
Audit Windows machines network connectivity	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if a network connection status to an IP and TCP port does not match the policy parameter.	auditIfNotExists	2.0.0
Audit Windows machines on which the DSC configuration is not compliant	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the Windows PowerShell command Get-DSCConfigurationStatus returns that the DSC configuration for the machine is not compliant.	auditIfNotExists	3.0.0
Audit Windows machines on which the Log Analytics agent is not connected as expected	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the agent is not installed, or if it is installed but the COM object AgentConfigManager.MgmtSvcCfg returns that it is registered to a workspace other than the ID specified in the policy parameter.	auditIfNotExists	2.0.0
Audit Windows machines on which the specified services are not installed and 'Running'	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if result of the Windows PowerShell command Get-Service do not include the service name with matching status as specified by the policy parameter.	auditIfNotExists	3.0.0
Audit Windows machines on which Windows Serial Console is not enabled	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the machine does not have the Serial Console software installed or if the EMS port number or baud rate are not configured with the same values as the policy parameters.	auditIfNotExists	3.0.0
Audit Windows machines that allow re-use of the passwords after the specified number of unique passwords	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows machines that allow re-use of the passwords after the specified number of unique passwords. Default value for unique passwords is 24	AuditIfNotExists, Disabled	2.1.0
Audit Windows machines that are not joined to the specified domain	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the value of the Domain property in WMI class win32_computersystem does not match the value in the policy parameter.	auditIfNotExists	2.0.0
Audit Windows machines that are not set to the specified time zone	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the value of the property StandardName in WMI class Win32_TimeZone does not match the selected time zone for the policy parameter.	auditIfNotExists	3.0.0
Audit Windows machines that contain certificates expiring within the specified number of days	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if certificates in the specified store have an expiration date out of range for the number of days given as parameter. The policy also provides the option to only check for specific certificates or exclude specific certificates, and whether to report on expired certificates.	auditIfNotExists	2.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Audit Windows machines that do not contain the specified certificates in Trusted Root	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the machine Trusted Root certificate store (Cert:\LocalMachine\Root) does not contain one or more of the certificates listed by the policy parameter.	auditIfNotExists	3.0.0
Audit Windows machines that do not have the maximum password age set to specified number of days	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows machines that do not have the maximum password age set to specified number of days. Default value for maximum password age is 70 days	AuditIfNotExists, Disabled	2.1.0
Audit Windows machines that do not have the minimum password age set to specified number of days	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows machines that do not have the minimum password age set to specified number of days. Default value for minimum password age is 1 day	AuditIfNotExists, Disabled	2.1.0
Audit Windows machines that do not have the password complexity setting enabled	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows machines that do not have the password complexity setting enabled	AuditIfNotExists, Disabled	2.0.0
Audit Windows machines that do not have the specified Windows PowerShell execution policy	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the Windows PowerShell command Get-ExecutionPolicy returns a value other than what was selected in the policy parameter.	AuditIfNotExists, Disabled	3.0.0
Audit Windows machines that do not have the specified Windows PowerShell modules installed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if a module isn't available in a location specified by the environment variable PSModulePath.	AuditIfNotExists, Disabled	3.0.0
Audit Windows machines that do not restrict the minimum password length to specified number of characters	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows machines that do not restrict the minimum password length to specified number of characters. Default value for minimum password length is 14 characters	AuditIfNotExists, Disabled	2.1.0
Audit Windows machines that do not store passwords using reversible encryption	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows machines that do not store passwords using reversible encryption	AuditIfNotExists, Disabled	2.0.0
Audit Windows machines that don't have the specified applications installed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the application name is not found in any of the following registry paths: HKLM:SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,	auditIfNotExists	2.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
	HKLM:SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Uninstall, HKCU:Software\Microsoft\Windows\CurrentVersion\Uninstall.		
Audit Windows machines that have extra accounts in the Administrators group	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the local Administrators group contains members that are not listed in the policy parameter.	auditIfNotExists	2.0.0
Audit Windows machines that have not restarted within the specified number of days	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the WMI property LastBootUpTime in class Win32_Operatingsystem is outside the range of days provided by the policy parameter.	auditIfNotExists	2.0.0
Audit Windows machines that have the specified applications installed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the application name is found in any of the following registry paths: HKLM:SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall, HKLM:SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Uninstall, HKCU:Software\Microsoft\Windows\CurrentVersion\Uninstall.	auditIfNotExists	2.0.0
Audit Windows machines that have the specified members in the Administrators group	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the local Administrators group contains one or more of the members listed in the policy parameter.	auditIfNotExists	2.0.0
Audit Windows VMs with a pending reboot	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the machine is pending reboot for any of the following reasons: component based servicing, Windows Update, pending file rename, pending computer rename, configuration manager pending reboot. Each detection has a unique registry path.	auditIfNotExists	2.0.0
Authentication to Linux machines should require SSH keys	Although SSH itself provides an encrypted connection, using passwords with SSH still leaves the VM vulnerable to brute-force attacks. The most secure option for authenticating to an Azure Linux virtual machine over SSH is with a public-private key pair, also known as SSH keys. Learn more: https://docs.microsoft.com/azure/virtual-machines/linux/create-ssh-keys-detailed.	AuditIfNotExists, Disabled	3.2.0
Azure Backup should be enabled for Virtual Machines	Ensure protection of your Azure Virtual Machines by enabling Azure Backup. Azure Backup is a secure and cost effective data protection solution for Azure.	AuditIfNotExists, Disabled	3.0.0
Cloud Services (extended support) role instances should be configured securely	Protect your Cloud Service (extended support) role instances from attacks by ensuring they are not expolosed to any OS vulnerabilities.	AuditIfNotExists, Disabled	1.0.0
Cloud Services (extended support) role instances should have an endpoint protection solution installed	Protect your Cloud Services (extended support) role instances from threats and vulnerabilities by ensuring an endpoint protection solution is installed on them.	AuditIfNotExists, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Cloud Services (extended support) role instances should have system updates installed	Secure your Cloud Services (extended support) role instances by ensuring the latest security and critical updates are installed on them.	AuditIfNotExists, Disabled	1.0.0
Configure Azure Defender for Servers to be disabled for all resources (resource level)	Azure Defender for Servers provides real-time threat protection for server workloads and generates hardening recommendations as well as alerts about suspicious activities. This policy will disable the Defender for Servers plan for all resources (VMs, VMSSs and ARC Machines) in the selected scope (subscription or resource group).	DeployIfNotExists, Disabled	1.0.0
Configure Azure Defender for Servers to be disabled for resources (resource level) with the selected tag	Azure Defender for Servers provides real-time threat protection for server workloads and generates hardening recommendations as well as alerts about suspicious activities. This policy will disable the Defender for Servers plan for all resources (VMs, VMSSs and ARC Machines) that have the selected tag name and tag value(s).	DeployIfNotExists, Disabled	1.0.0
Configure Azure Defender for Servers to be enabled ('P1' subplan) for all resources (resource level) with the selected tag	Azure Defender for Servers provides real-time threat protection for server workloads and generates hardening recommendations as well as alerts about suspicious activities. This policy will enable the Defender for Servers plan (with 'P1' subplan) for all resources (VMs and ARC Machines) that have the selected tag name and tag value(s).	DeployIfNotExists, Disabled	1.0.0
Configure Azure Defender for Servers to be enabled (with 'P1' subplan) for all resources (resource level)	Azure Defender for Servers provides real-time threat protection for server workloads and generates hardening recommendations as well as alerts about suspicious activities. This policy will enable the Defender for Servers plan (with 'P1' subplan) for all resources (VMs and ARC Machines) in the selected scope (subscription or resource group).	DeployIfNotExists, Disabled	1.0.0
Configure backup on virtual machines with a given tag to a new recovery services vault with a default policy	Enforce backup for all virtual machines by deploying a recovery services vault in the same location and resource group as the virtual machine. Doing this is useful when different application teams in your organization are allocated separate resource groups and need to manage their own backups and restores. You can optionally include virtual machines containing a specified tag to control the scope of assignment. See https://aka.ms/AzureVMAppCentricBackupIncludeTag.	auditIfNotExists, AuditIfNotExists, deployIfNotExists, DeployIfNotExists, disabled, Disabled	9.3.0
Configure backup on virtual machines with a given tag to an existing recovery services vault in the same location	Enforce backup for all virtual machines by backing them up to an existing central recovery services vault in the same location and subscription as the virtual machine. Doing this is useful when there is a central team in your organization managing backups for all resources in a subscription. You can optionally include virtual machines containing a specified tag to control the scope of assignment. See https://aka.ms/AzureVMCentralBackupIncludeTag.	auditIfNotExists, AuditIfNotExists, deployIfNotExists, DeployIfNotExists, disabled, Disabled	9.3.0
Configure backup on virtual machines without a given tag to a new recovery services vault with a default policy	Enforce backup for all virtual machines by deploying a recovery services vault in the same location and resource group as the virtual machine. Doing this is useful when different application teams in your organization are allocated separate resource groups and need to manage their own backups and restores. You can optionally exclude virtual machines containing a specified tag to control the scope of assignment. See https://aka.ms/AzureVMAppCentricBackupExcludeTag.	auditIfNotExists, AuditIfNotExists, deployIfNotExists, DeployIfNotExists, disabled, Disabled	9.3.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Configure backup on virtual machines without a given tag to an existing recovery services vault in the same location	Enforce backup for all virtual machines by backing them up to an existing central recovery services vault in the same location and subscription as the virtual machine. Doing this is useful when there is a central team in your organization managing backups for all resources in a subscription. You can optionally exclude virtual machines containing a specified tag to control the scope of assignment. See https://aka.ms/AzureVMCentralBackupExcludeTag.	auditIfNotExists, AuditIfNotExists, deployIfNotExists, DeployIfNotExists, disabled, Disabled	9.3.0
Configure disaster recovery on virtual machines by enabling replication via Azure Site Recovery	Virtual machines without disaster recovery configurations are vulnerable to outages and other disruptions. If the virtual machine does not already have disaster recovery configured, this would initiate the same by enabling replication using preset configurations to facilitate business continuity. You can optionally include/exclude virtual machines containing a specified tag to control the scope of assignment. To learn more about disaster recovery, visit https://aka.ms/asr-doc.	DeployIfNotExists, Disabled	2.1.0
Configure disk access resources with private endpoints	Private endpoints connect your virtual networks to Azure services without a public IP address at the source or destination. By mapping private endpoints to disk access resources, you can reduce data leakage risks. Learn more about private links at: https://aka.ms/disksprivatelinksdoc.	DeployIfNotExists, Disabled	1.0.0
Configure Linux Machines to be associated with a Data Collection Rule or a Data Collection Endpoint	Deploy Association to link Linux virtual machines, virtual machine scale sets, and Arc machines to the specified Data Collection Rule or the specified Data Collection Endpoint. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	6.5.0
Configure Linux Server to disable local users.	Creates a Guest Configuration assignment to configure disabling local users on Linux Server. This ensures that Linux Servers can only be accessed by AAD (Azure Active Directory) account or a list of explicitly allowed users by this policy, improving overall security posture.	DeployIfNotExists, Disabled	1.3.0-preview
Configure Linux Virtual Machine Scale Sets to be associated with a Data Collection Rule or a Data Collection Endpoint	Deploy Association to link Linux virtual machine scale sets to the specified Data Collection Rule or the specified Data Collection Endpoint. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	4.4.0
Configure Linux virtual machine scale sets to run Azure Monitor Agent with system-assigned managed identity-based authentication	Automate the deployment of Azure Monitor Agent extension on your Linux virtual machine scale sets for collecting telemetry data from the guest OS. This policy will install the extension if the OS and region are supported and system-assigned managed identity is enabled, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	3.6.0
Configure Linux virtual machine scale sets to run Azure Monitor Agent with user-assigned managed identity-based authentication	Automate the deployment of Azure Monitor Agent extension on your Linux virtual machine scale sets for collecting telemetry data from the guest OS. This policy will install the extension and configure it to use the specified user-assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	3.8.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Configure Linux Virtual Machines to be associated with a Data Collection Rule or a Data Collection Endpoint	Deploy Association to link Linux virtual machines to the specified Data Collection Rule or the specified Data Collection Endpoint. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	4.4.0
Configure Linux virtual machines to run Azure Monitor Agent with system-assigned managed identity-based authentication	Automate the deployment of Azure Monitor Agent extension on your Linux virtual machines for collecting telemetry data from the guest OS. This policy will install the extension if the OS and region are supported and system-assigned managed identity is enabled, and skip install otherwise. Learn more: https://aka.ms/AMAOverview .	DeployIfNotExists, Disabled	3.6.0
Configure Linux virtual machines to run Azure Monitor Agent with user-assigned managed identity-based authentication	Automate the deployment of Azure Monitor Agent extension on your Linux virtual machines for collecting telemetry data from the guest OS. This policy will install the extension and configure it to use the specified user-assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview .	DeployIfNotExists, Disabled	3.8.0
Configure machines to receive a vulnerability assessment provider	Azure Defender includes vulnerability scanning for your machines at no extra cost. You don't need a Qualys license or even a Qualys account - everything's handled seamlessly inside Security Center. When you enable this policy, Azure Defender automatically deploys the Qualys vulnerability assessment provider to all supported machines that don't already have it installed.	DeployIfNotExists, Disabled	4.0.0
Configure managed disks to disable public network access	Disable public network access for your managed disk resource so that it's not accessible over the public internet. This can reduce data leakage risks. Learn more at: https://aka.ms/disksprivatelinksdoc .	Modify, Disabled	2.0.0
Configure periodic checking for missing system updates on azure virtual machines	Configure auto-assessment (every 24 hours) for OS updates on native Azure virtual machines. You can control the scope of assignment according to machine subscription, resource group, location or tag. Learn more about this for Windows: https://aka.ms/computevm-windowspatchassessmentmode, for Linux: https://aka.ms/computevm-linuxpatchassessmentmode .	modify	4.8.0
Configure secure communication protocols(TLS 1.1 or TLS 1.2) on Windows machines	Creates a Guest Configuration assignment to configure specified secure protocol version(TLS 1.1 or TLS 1.2) on Windows machine.	DeployIfNotExists, Disabled	1.0.1
Configure SQL Virtual Machines to automatically install Azure Monitor Agent	Automate the deployment of Azure Monitor Agent extension on your Windows SQL Virtual Machines. Learn more: https://aka.ms/AMAOverview .	DeployIfNotExists, Disabled	1.4.0
Configure SQL Virtual Machines to automatically install Microsoft Defender for SQL	Configure Windows SQL Virtual Machines to automatically install the Microsoft Defender for SQL extension. Microsoft Defender for SQL collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations).	DeployIfNotExists, Disabled	1.4.0
Configure SQL Virtual Machines to automatically install	Microsoft Defender for SQL collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Create	DeployIfNotExists, Disabled	1.6.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Microsoft Defender for SQL and DCR with a Log Analytics workspace	a resource group, a Data Collection Rule and Log Analytics workspace in the same region as the machine.		
Configure SQL Virtual Machines to automatically install Microsoft Defender for SQL and DCR with a user-defined LA workspace	Microsoft Defender for SQL collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Create a resource group and a Data Collection Rule in the same region as the user-defined Log Analytics workspace.	DeployIfNotExists, Disabled	1.7.0
Configure the Microsoft Defender for SQL Log Analytics workspace	Microsoft Defender for SQL collects events from the agent and uses them to provide security alerts and tailored hardening tasks (recommendations). Create a resource group and Log Analytics workspace in the same region as the machine.	DeployIfNotExists, Disabled	1.3.0
Configure time zone on Windows machines.	This policy creates a Guest Configuration assignment to set specified time zone on Windows virtual machines.	deployIfNotExists	2.1.0
Configure virtual machines to be onboarded to Azure Automanage	Azure Automanage enrolls, configures, and monitors virtual machines with best practice as defined in the Microsoft Cloud Adoption Framework for Azure. Use this policy to apply Automanage to your selected scope.	AuditIfNotExists, DeployIfNotExists, Disabled	2.4.0
Configure virtual machines to be onboarded to Azure Automanage with Custom Configuration Profile	Azure Automanage enrolls, configures, and monitors virtual machines with best practice as defined in the Microsoft Cloud Adoption Framework for Azure. Use this policy to apply Automanage with your own customized Configuration Profile to your selected scope.	AuditIfNotExists, DeployIfNotExists, Disabled	1.4.0
Configure Windows Machines to be associated with a Data Collection Rule or a Data Collection Endpoint	Deploy Association to link Windows virtual machines, virtual machine scale sets, and Arc machines to the specified Data Collection Rule or the specified Data Collection Endpoint. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	4.5.0
Configure Windows Virtual Machine Scale Sets to be associated with a Data Collection Rule or a Data Collection Endpoint	Deploy Association to link Windows virtual machine scale sets to the specified Data Collection Rule or the specified Data Collection Endpoint. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	3.3.0
Configure Windows virtual machine scale sets to run Azure Monitor Agent using system-assigned managed identity	Automate the deployment of Azure Monitor Agent extension on your Windows virtual machine scale sets for collecting telemetry data from the guest OS. This policy will install the extension if the OS and region are supported and system-assigned managed identity is enabled, and skip install otherwise. Learn more: https://aka.ms/AMAOverview .	DeployIfNotExists, Disabled	3.4.0
Configure Windows virtual machine scale sets to run	Automate the deployment of Azure Monitor Agent extension on your Windows virtual machine scale sets for collecting telemetry data from the guest OS. This policy will install the extension and configure it to use the specified user-	DeployIfNotExists, Disabled	1.6.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Azure Monitor Agent with user-assigned managed identity-based authentication	assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.		
Configure Windows Virtual Machines to be associated with a Data Collection Rule or a Data Collection Endpoint	Deploy Association to link Windows virtual machines to the specified Data Collection Rule or the specified Data Collection Endpoint. The list of locations and OS images are updated over time as support is increased.	DeployIfNotExists, Disabled	3.3.0
Configure Windows virtual machines to run Azure Monitor Agent using system-assigned managed identity	Automate the deployment of Azure Monitor Agent extension on your Windows virtual machines for collecting telemetry data from the guest OS. This policy will install the extension if the OS and region are supported and system-assigned managed identity is enabled, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	4.4.0
Configure Windows virtual machines to run Azure Monitor Agent with user-assigned managed identity-based authentication	Automate the deployment of Azure Monitor Agent extension on your Windows virtual machines for collecting telemetry data from the guest OS. This policy will install the extension and configure it to use the specified user-assigned managed identity if the OS and region are supported, and skip install otherwise. Learn more: https://aka.ms/AMAOverview.	DeployIfNotExists, Disabled	1.6.0
Create and assign a built-in user-assigned managed identity	Create and assign a built-in user-assigned managed identity at scale to SQL virtual machines.	AuditIfNotExists, DeployIfNotExists, Disabled	1.6.0
Dependency agent should be enabled for listed virtual machine images	Reports virtual machines as non-compliant if the virtual machine image is not in the list defined and the agent is not installed. The list of OS images is updated over time as support is updated.	AuditIfNotExists, Disabled	2.0.0
Dependency agent should be enabled in virtual machine scale sets for listed virtual machine images	Reports virtual machine scale sets as non-compliant if the virtual machine image is not in the list defined and the agent is not installed. The list of OS images is updated over time as support is updated.	AuditIfNotExists, Disabled	2.0.0
Deploy - Configure Dependency agent to be enabled on Windows virtual machine scale sets	Deploy Dependency agent for Windows virtual machine scale sets if the virtual machine image is in the list defined and the agent is not installed. If your scale set upgradePolicy is set to Manual, you need to apply the extension to all the virtual machines in the set by updating them.	DeployIfNotExists, Disabled	3.2.0
Deploy - Configure Dependency agent to be enabled on Windows virtual machines	Deploy Dependency agent for Windows virtual machines if the virtual machine image is in the list defined and the agent is not installed.	DeployIfNotExists, Disabled	3.2.0
Deploy - Configure Log Analytics extension to be enabled on	Deploy Log Analytics extension for Windows virtual machine scale sets if the virtual machine image is in the list defined and the extension is not installed. If your scale set upgradePolicy is set to Manual, you need to apply the extension to all the virtual machine in the set by updating them. Deprecation notice: The	DeployIfNotExists, Disabled	3.1.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows virtual machine scale sets	Log Analytics agent is on a deprecation path and won't be supported after August 31, 2024. You must migrate to the replacement 'Azure Monitor agent' prior to that date.		
Deploy - Configure Log Analytics extension to be enabled on Windows virtual machines	Deploy Log Analytics extension for Windows virtual machines if the virtual machine image is in the list defined and the extension is not installed. Deprecation notice: The Log Analytics agent is on a deprecation path and won't be supported after August 31, 2024. You must migrate to the replacement 'Azure Monitor agent' prior to that date.	DeployIfNotExists, Disabled	3.1.0
Deploy default Microsoft IaaSAntimalware extension for Windows Server	This policy deploys a Microsoft IaaSAntimalware extension with a default configuration when a VM is not configured with the antimalware extension.	deployIfNotExists	1.1.0
Deploy Dependency agent for Linux virtual machine scale sets	Deploy Dependency agent for Linux virtual machine scale sets if the VM Image (OS) is in the list defined and the agent is not installed. Note: if your scale set upgradePolicy is set to Manual, you need to apply the extension to the all virtual machines in the set by calling upgrade on them. In CLI this would be az vmss update-instances.	deployIfNotExists	5.1.0
Deploy Dependency agent for Linux virtual machine scale sets with Azure Monitoring Agent settings	Deploy Dependency agent for Linux virtual machine scale sets with Azure Monitoring Agent settings if the VM Image (OS) is in the list defined and the agent is not installed. Note: if your scale set upgradePolicy is set to Manual, you need to apply the extension to the all virtual machines in the set by calling upgrade on them. In CLI this would be az vmss update-instances.	DeployIfNotExists, Disabled	3.2.0
Deploy Dependency agent for Linux virtual machines	Deploy Dependency agent for Linux virtual machines if the VM Image (OS) is in the list defined and the agent is not installed.	deployIfNotExists	5.1.0
Deploy Dependency agent for Linux virtual machines with Azure Monitoring Agent settings	Deploy Dependency agent for Linux virtual machines with Azure Monitoring Agent settings if the VM Image (OS) is in the list defined and the agent is not installed.	DeployIfNotExists, Disabled	3.2.0
Deploy Dependency agent to be enabled on Windows virtual machine scale sets with Azure Monitoring Agent settings	Deploy Dependency agent for Windows virtual machine scale sets with Azure Monitoring Agent settings if the virtual machine image is in the list defined and the agent is not installed. If your scale set upgradePolicy is set to Manual, you need to apply the extension to all the virtual machines in the set by updating them.	DeployIfNotExists, Disabled	1.3.0
Deploy Dependency agent to be enabled on Windows virtual machines with Azure Monitoring Agent settings	Deploy Dependency agent for Windows virtual machines with Azure Monitoring Agent settings if the virtual machine image is in the list defined and the agent is not installed.	DeployIfNotExists, Disabled	1.3.0
Deploy Log Analytics extension for Linux virtual machine scale sets.	Deploy Log Analytics extension for Linux virtual machine scale sets if the VM Image (OS) is in the list defined and the extension is not installed. Note: if your scale set upgradePolicy is set to Manual, you need to apply the extension to the all VMs in the set by calling upgrade on them. In CLI this would be az vmss update-instances. Deprecation notice: The Log Analytics agent will not be	deployIfNotExists	3.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
See deprecation notice below	supported after August 31, 2024. You must migrate to the replacement 'Azure Monitor agent' prior to that date		
Deploy Log Analytics extension for Linux VMs. See deprecation notice below	Deploy Log Analytics extension for Linux VMs if the VM Image (OS) is in the list defined and the extension is not installed. Deprecation notice: The Log Analytics agent is on a deprecation path and won't be supported after August 31, 2024. You must migrate to the replacement 'Azure Monitor agent' prior to that date	deployIfNotExists	3.0.0
Deploy the Linux Guest Configuration extension to enable Guest Configuration assignments on Linux VMs	This policy deploys the Linux Guest Configuration extension to Linux virtual machines hosted in Azure that are supported by Guest Configuration. The Linux Guest Configuration extension is a prerequisite for all Linux Guest Configuration assignments and must be deployed to machines before using any Linux Guest Configuration policy definition. For more information on Guest Configuration, visit https://aka.ms/gcpol.	deployIfNotExists	3.1.0
Deploy the Windows Guest Configuration extension to enable Guest Configuration assignments on Windows VMs	This policy deploys the Windows Guest Configuration extension to Windows virtual machines hosted in Azure that are supported by Guest Configuration. The Windows Guest Configuration extension is a prerequisite for all Windows Guest Configuration assignments and must be deployed to machines before using any Windows Guest Configuration policy definition. For more information on Guest Configuration, visit https://aka.ms/gcpol.	deployIfNotExists	1.2.0
Disk access resources should use private link	Azure Private Link lets you connect your virtual network to Azure services without a public IP address at the source or destination. The Private Link platform handles the connectivity between the consumer and services over the Azure backbone network. By mapping private endpoints to diskAccesses, data leakage risks are reduced. Learn more about private links at: https://aka.ms/disksprivatelinksdoc.	AuditIfNotExists, Disabled	1.0.0
Disks and OS image should support TrustedLaunch	TrustedLaunch improves security of a Virtual Machine which requires OS Disk & OS Image to support it (Gen 2). To learn more about TrustedLaunch, visit https://aka.ms/trustedlaunch	Audit, Disabled	1.0.0
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0
Endpoint protection should be installed on your machines	To protect your machines from threats and vulnerabilities, install a supported endpoint protection solution.	AuditIfNotExists, Disabled	1.0.0
Endpoint protection solution should be installed on virtual machine scale sets	Audit the existence and health of an endpoint protection solution on your virtual machines scale sets, to protect them from threats and vulnerabilities.	AuditIfNotExists, Disabled	3.0.0
Guest Configuration extension should be installed on your machines	To ensure secure configurations of in-guest settings of your machine, install the Guest Configuration extension. In-guest settings that the extension monitors include the configuration of the operating system, application configuration or presence, and environment settings. Once installed, in-guest policies will be available such as 'Windows Exploit guard should be enabled'. Learn more at https://aka.ms/gcpol.	AuditIfNotExists, Disabled	1.0.3

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Hotpatch should be enabled for Windows Server Azure Edition VMs	Minimize reboots and install updates quickly with hotpatch. Learn more at https://docs.microsoft.com/azure/automanage/automanage-hotpatch	Audit, Deny, Disabled	1.0.0
Internet-facing virtual machines should be protected with network security groups	Protect your virtual machines from potential threats by restricting access to them with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc	AuditIfNotExists, Disabled	3.0.0
IP Forwarding on your virtual machine should be disabled	Enabling IP forwarding on a virtual machine's NIC allows the machine to receive traffic addressed to other destinations. IP forwarding is rarely required (e.g., when using the VM as a network virtual appliance), and therefore, this should be reviewed by the network security team.	AuditIfNotExists, Disabled	3.0.0
Linux machines should meet requirements for the Azure compute security baseline	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if the machine is not configured correctly for one of the recommendations in the Azure compute security baseline.	AuditIfNotExists, Disabled	2.2.0
Linux machines should only have local accounts that are allowed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Managing user accounts using Azure Active Directory is a best practice for management of identities. Reducing local machine accounts helps prevent the proliferation of identities managed outside a central system. Machines are non-compliant if local user accounts exist that are enabled and not listed in the policy parameter.	AuditIfNotExists, Disabled	2.2.0
Linux virtual machine scale sets should have Azure Monitor Agent installed	Linux virtual machine scale sets should be monitored and secured through the deployed Azure Monitor Agent. The Azure Monitor Agent collects telemetry data from the guest OS. This policy will audit virtual machine scale sets with supported OS images in supported regions. Learn more: https://aka.ms/AMAOverview.	AuditIfNotExists, Disabled	3.3.0
Linux virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	Although a virtual machine's OS and data disks are encrypted-at-rest by default using platform managed keys; resource disks (temp disks), data caches, and data flowing between Compute and Storage resources are not encrypted. Use Azure Disk Encryption or EncryptionAtHost to remediate. Visit https://aka.ms/diskencryptioncomparison to compare encryption offerings. This policy requires two prerequisites to be deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	1.2.1
Linux virtual machines should have Azure Monitor Agent installed	Linux virtual machines should be monitored and secured through the deployed Azure Monitor Agent. The Azure Monitor Agent collects telemetry data from the guest OS. This policy will audit virtual machines with supported OS images in supported regions. Learn more: https://aka.ms/AMAOverview.	AuditIfNotExists, Disabled	3.3.0
Local authentication methods should be disabled on Linux machines	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Linux servers don't have local authentication methods disabled. This is to validate that Linux Servers can only be accessed by AAD (Azure Active Directory) account or a list of explicitly allowed users by this policy, improving overall security posture.	AuditIfNotExists, Disabled	1.2.0-preview
Local authentication methods should be disabled on Windows Servers	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol. Machines are non-compliant if Windows servers don't have local authentication methods disabled. This is to validate that Windows Servers can only be accessed by AAD (Azure Active Directory) account or a list of explicitly allowed users by this policy, improving overall security posture.	AuditIfNotExists, Disabled	1.0.0-preview

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Log Analytics agent should be installed on your Cloud Services (extended support) role instances	Security Center collects data from your Cloud Services (extended support) role instances to monitor for security vulnerabilities and threats.	AuditIfNotExists, Disabled	2.0.0
Log Analytics extension should be enabled in virtual machine scale sets for listed virtual machine images	Reports virtual machine scale sets as non-compliant if the virtual machine image is not in the list defined and the extension is not installed.	AuditIfNotExists, Disabled	2.0.1
Machines should be configured to periodically check for missing system updates	To ensure periodic assessments for missing system updates are triggered automatically every 24 hours, the AssessmentMode property should be set to 'AutomaticByPlatform'. Learn more about AssessmentMode property for Windows: https://aka.ms/computevm-windowspatchassessmentmode, for Linux: https://aka.ms/computevm-linuxpatchassessmentmode.	Audit, Deny, Disabled	3.7.0
Machines should have secret findings resolved	Audits virtual machines to detect whether they contain secret findings from the secret scanning solutions on your virtual machines.	AuditIfNotExists, Disabled	1.0.2
Managed disks should be double encrypted with both platform-managed and customer-managed keys	High security sensitive customers who are concerned of the risk associated with any particular encryption algorithm, implementation, or key being compromised can opt for additional layer of encryption using a different encryption algorithm/mode at the infrastructure layer using platform managed encryption keys. The disk encryption sets are required to use double encryption. Learn more at https://aka.ms/disks-doubleEncryption.	Audit, Deny, Disabled	1.0.0
Managed disks should disable public network access	Disabling public network access improves security by ensuring that a managed disk isn't exposed on the public internet. Creating private endpoints can limit exposure of managed disks. Learn more at: https://aka.ms/disksprivatelinksdoc.	Audit, Disabled	2.0.0
Managed disks should use a specific set of disk encryption sets for the customer-managed key encryption	Requiring a specific set of disk encryption sets to be used with managed disks give you control over the keys used for encryption at rest. You are able to select the allowed encrypted sets and all others are rejected when attached to a disk. Learn more at https://aka.ms/disks-cmk.	Audit, Deny, Disabled	2.0.0
Management ports of virtual machines should be protected with just-in-time network access control	Possible network Just In Time (JIT) access will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	3.0.0
Management ports should be closed on your virtual machines	Open remote management ports are exposing your VM to a high level of risk from Internet-based attacks. These attacks attempt to brute force credentials to gain admin access to the machine.	AuditIfNotExists, Disabled	3.0.0
Microsoft Antimalware for Azure should be configured to automatically	This policy audits any Windows virtual machine not configured with automatic update of Microsoft Antimalware protection signatures.	AuditIfNotExists, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
update protection signatures			
Microsoft IaaSAntimalware extension should be deployed on Windows servers	This policy audits any Windows server VM without Microsoft IaaSAntimalware extension deployed.	AuditIfNotExists, Disabled	1.1.0
Monitor missing Endpoint Protection in Azure Security Center	Servers without an installed Endpoint Protection agent will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	3.0.0
Non-internet-facing virtual machines should be protected with network security groups	Protect your non-internet-facing virtual machines from potential threats by restricting access with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc	AuditIfNotExists, Disabled	3.0.0
Only approved VM extensions should be installed	This policy governs the virtual machine extensions that are not approved.	Audit, Deny, Disabled	1.0.0
OS and data disks should be encrypted with a customer-managed key	Use customer-managed keys to manage the encryption at rest of the contents of your managed disks. By default, the data is encrypted at rest with platform-managed keys, but customer-managed keys are commonly required to meet regulatory compliance standards. Customer-managed keys enable the data to be encrypted with an Azure Key Vault key created and owned by you. You have full control and responsibility for the key lifecycle, including rotation and management. Learn more at https://aka.ms/disks-cmk .	Audit, Deny, Disabled	3.0.0
Private endpoints for Guest Configuration assignments should be enabled	Private endpoint connections enforce secure communication by enabling private connectivity to Guest Configuration for virtual machines. Virtual machines will be non-compliant unless they have the tag, 'EnablePrivateNetworkGC'. This tag enforces secure communication through private connectivity to Guest Configuration for Virtual Machines. Private connectivity limits access to traffic coming only from known networks and prevents access from all other IP addresses, including within Azure.	Audit, Deny, Disabled	1.1.0
Protect your data with authentication requirements when exporting or uploading to a disk or snapshot.	When export/upload URL is used, the system checks if the user has an identity in Azure Active Directory and has necessary permissions to export/upload the data. Please refer to aka.ms/DisksAzureADAuth.	Modify, Disabled	1.0.0
Require automatic OS image patching on Virtual Machine Scale Sets	This policy enforces enabling automatic OS image patching on Virtual Machine Scale Sets to always keep Virtual Machines secure by safely applying latest security patches every month.	deny	1.0.0
Schedule recurring updates using Azure Update Manager	You can use Azure Update Manager in Azure to save recurring deployment schedules to install operating system updates for your Windows Server and Linux machines in Azure, in on-premises environments, and in other cloud environments connected using Azure Arc-enabled servers. This policy will also change the patch mode for the Azure Virtual Machine to 'AutomaticByPlatform'. See more: https://aka.ms/umc-scheduled-patching	DeployIfNotExists, Disabled	3.12.0
SQL servers on machines should	SQL vulnerability assessment scans your database for security vulnerabilities, and exposes any deviations from best practices such as misconfigurations,	AuditIfNotExists, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
have vulnerability findings resolved	excessive permissions, and unprotected sensitive data. Resolving the vulnerabilities found can greatly improve your database security posture.		
System updates on virtual machine scale sets should be installed	Audit whether there are any missing system security updates and critical updates that should be installed to ensure that your Windows and Linux virtual machine scale sets are secure.	AuditIfNotExists, Disabled	3.0.0
System updates should be installed on your machines	Missing security system updates on your servers will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	4.0.0
The legacy Log Analytics extension should not be installed on Linux virtual machine scale sets	Automatically prevent installation of the legacy Log Analytics Agent as the final step of migrating from legacy agents to Azure Monitor Agent. After you have uninstalled existing legacy extensions, this policy will deny all future installations of the legacy agent extension on Linux virtual machine scale sets. Learn more: https://aka.ms/migratetoAMA	Deny, Audit, Disabled	1.0.0
The legacy Log Analytics extension should not be installed on Linux virtual machines	Automatically prevent installation of the legacy Log Analytics Agent as the final step of migrating from legacy agents to Azure Monitor Agent. After you have uninstalled existing legacy extensions, this policy will deny all future installations of the legacy agent extension on Linux virtual machines. Learn more: https://aka.ms/migratetoAMA	Deny, Audit, Disabled	1.0.0
The legacy Log Analytics extension should not be installed on virtual machine scale sets	Automatically prevent installation of the legacy Log Analytics Agent as the final step of migrating from legacy agents to Azure Monitor Agent. After you have uninstalled existing legacy extensions, this policy will deny all future installations of the legacy agent extension on Windows virtual machine scale sets. Learn more: https://aka.ms/migratetoAMA	Deny, Audit, Disabled	1.0.0
The legacy Log Analytics extension should not be installed on virtual machines	Automatically prevent installation of the legacy Log Analytics Agent as the final step of migrating from legacy agents to Azure Monitor Agent. After you have uninstalled existing legacy extensions, this policy will deny all future installations of the legacy agent extension on Windows virtual machines. Learn more: https://aka.ms/migratetoAMA	Deny, Audit, Disabled	1.0.0
The Log Analytics extension should be installed on Virtual Machine Scale Sets	This policy audits any Windows/Linux Virtual Machine Scale Sets if the Log Analytics extension is not installed.	AuditIfNotExists, Disabled	1.0.1
Virtual Machine should have TrustedLaunch enabled	Enable TrustedLaunch on Virtual Machine for enhanced security, use VM SKU (Gen 2) that supports TrustedLaunch. To learn more about TrustedLaunch, visit https://learn.microsoft.com/en-us/azure/virtual-machines/trusted-launch	Audit, Disabled	1.0.0
Virtual machines and virtual machine scale sets should have encryption at host enabled	Use encryption at host to get end-to-end encryption for your virtual machine and virtual machine scale set data. Encryption at host enables encryption at rest for your temporary disk and OS/data disk caches. Temporary and ephemeral OS disks are encrypted with platform-managed keys when encryption at host is enabled. OS/data disk caches are encrypted at rest with either customer-managed or platform-managed key, depending on the encryption type selected on the disk. Learn more at https://aka.ms/vm-hbe.	Audit, Deny, Disabled	1.0.0
Virtual machines should be connected to a specified workspace	Reports virtual machines as non-compliant if they aren't logging to the Log Analytics workspace specified in the policy/initiative assignment.	AuditIfNotExists, Disabled	1.1.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should be migrated to new Azure Resource Manager resources	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance, access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management	Audit, Deny, Disabled	1.0.0
Virtual machines should have the Log Analytics extension installed	This policy audits any Windows/Linux virtual machines if the Log Analytics extension is not installed.	AuditIfNotExists, Disabled	1.0.1
Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	The Guest Configuration extension requires a system assigned managed identity. Azure virtual machines in the scope of this policy will be non-compliant when they have the Guest Configuration extension installed but do not have a system assigned managed identity. Learn more at https://aka.ms/gcpol	AuditIfNotExists, Disabled	1.0.1
Vulnerabilities in container security configurations should be remediated	Audit vulnerabilities in security configuration on machines with Docker installed and display as recommendations in Azure Security Center.	AuditIfNotExists, Disabled	3.0.0
Vulnerabilities in security configuration on your machines should be remediated	Servers which do not satisfy the configured baseline will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	3.1.0
Vulnerabilities in security configuration on your virtual machine scale sets should be remediated	Audit the OS vulnerabilities on your virtual machine scale sets to protect them from attacks.	AuditIfNotExists, Disabled	3.0.0
Windows Defender Exploit Guard should be enabled on your machines	Windows Defender Exploit Guard uses the Azure Policy Guest Configuration agent. Exploit Guard has four components that are designed to lock down devices against a wide variety of attack vectors and block behaviors commonly used in malware attacks while enabling enterprises to balance their security risk and productivity requirements (Windows only).	AuditIfNotExists, Disabled	2.0.0
Windows machines should be configured to use secure communication protocols	To protect the privacy of information communicated over the Internet, your machines should use the latest version of the industry-standard cryptographic protocol, Transport Layer Security (TLS). TLS secures communications over a network by encrypting a connection between machines.	AuditIfNotExists, Disabled	4.1.1
Windows machines should configure Windows Defender to update protection signatures within one day	To provide adequate protection against newly released malware, Windows Defender protection signatures need to be updated regularly to account for newly released malware. This policy is not applied to Arc connected servers and it requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For more information on Guest Configuration, visit https://aka.ms/gcpol .	AuditIfNotExists, Disabled	1.0.1

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows machines should enable Windows Defender Real-time protection	Windows machines should enable the Real-time protection in the Windows Defender to provide adequate protection against newly released malware. This policy is not applicable to arc connected servers and it requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For more information on Guest Configuration, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	1.0.1
Windows machines should meet requirements for 'Administrative Templates - Control Panel'	Windows machines should have the specified Group Policy settings in the category 'Administrative Templates - Control Panel' for input personalization and prevention of enabling lock screens. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Administrative Templates - MSS (Legacy)'	Windows machines should have the specified Group Policy settings in the category 'Administrative Templates - MSS (Legacy)' for automatic logon, screen saver, network behavior, safe DLL, and event log. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Administrative Templates - Network'	Windows machines should have the specified Group Policy settings in the category 'Administrative Templates - Network' for guest logons, simultaneous connections, network bridge, ICS, and multicast name resolution. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Administrative Templates - System'	Windows machines should have the specified Group Policy settings in the category 'Administrative Templates - System' for settings that control the administrative experience and Remote Assistance. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Accounts'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Accounts' for limiting local account use of blank passwords and guest account status. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Audit'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Audit' for forcing audit policy subcategory and shutting down if unable to log security audits. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Devices'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Devices' for undocking without logging on, installing print drivers, and formatting/ejecting media. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Interactive Logon'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Interactive Logon' for displaying last user name and requiring ctrl-alt-del. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for	Windows machines should have the specified Group Policy settings in the category 'Security Options - Microsoft Network Client' for Microsoft network client/server and SMB v1. This policy requires that the Guest Configuration	AuditIfNotExists, Disabled	3.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
'Security Options - Microsoft Network Client'	prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.		
Windows machines should meet requirements for 'Security Options - Microsoft Network Server'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Microsoft Network Server' for disabling SMB v1 server. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Network Access'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Network Access' for including access for anonymous users, local accounts, and remote access to the registry. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Network Security'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Network Security' for including Local System behavior, PKU2U, LAN Manager, LDAP client, and NTLM SSP. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Recovery console'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Recovery console' for allowing floppy copy and access to all drives and folders. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - Shutdown'	Windows machines should have the specified Group Policy settings in the category 'Security Options - Shutdown' for allowing shutdown without logon and clearing the virtual memory pagefile. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - System objects'	Windows machines should have the specified Group Policy settings in the category 'Security Options - System objects' for case insensitivity for non-Windows subsystems and permissions of internal system objects. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - System settings'	Windows machines should have the specified Group Policy settings in the category 'Security Options - System settings' for certificate rules on executables for SRP and optional subsystems. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Options - User Account Control'	Windows machines should have the specified Group Policy settings in the category 'Security Options - User Account Control' for mode for admins, behavior of elevation prompt, and virtualizing file and registry write failures. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Security Settings - Account Policies'	Windows machines should have the specified Group Policy settings in the category 'Security Settings - Account Policies' for password history, age, length, complexity, and storing passwords using reversible encryption. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows machines should meet requirements for 'System Audit Policies - Account Logon'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Account Logon' for auditing credential validation and other account logon events. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - Account Management'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Account Management' for auditing application, security, and user group management, and other management events. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - Detailed Tracking'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Detailed Tracking' for auditing DPAPI, process creation/termination, RPC events, and PNP activity. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - Logon-Logoff'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Logon-Logoff' for auditing IPSec, network policy, claims, account lockout, group membership, and logon/logoff events. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - Object Access'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Object Access' for auditing file, registry, SAM, storage, filtering, kernel, and other system types. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - Policy Change'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Policy Change' for auditing changes to system audit policies. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - Privilege Use'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - Privilege Use' for auditing nonsensitive and other privilege use. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'System Audit Policies - System'	Windows machines should have the specified Group Policy settings in the category 'System Audit Policies - System' for auditing IPsec driver, system integrity, system extension, state change, and other system events. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'User Rights Assignment'	Windows machines should have the specified Group Policy settings in the category 'User Rights Assignment' for allowing log on locally, RDP, access from the network, and many other user activities. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.	AuditIfNotExists, Disabled	3.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows machines should meet requirements for 'Windows Components'	Windows machines should have the specified Group Policy settings in the category 'Windows Components' for basic authentication, unencrypted traffic, Microsoft accounts, telemetry, Cortana, and other Windows behaviors. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol .	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements for 'Windows Firewall Properties'	Windows machines should have the specified Group Policy settings in the category 'Windows Firewall Properties' for firewall state, connections, rule management, and notifications. This policy requires that the Guest Configuration prerequisites have been deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol .	AuditIfNotExists, Disabled	3.0.0
Windows machines should meet requirements of the Azure compute security baseline	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol . Machines are non-compliant if the machine is not configured correctly for one of the recommendations in the Azure compute security baseline.	AuditIfNotExists, Disabled	2.0.0
Windows machines should only have local accounts that are allowed	Requires that prerequisites are deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol . This definition is not supported on Windows Server 2012 or 2012 R2. Managing user accounts using Azure Active Directory is a best practice for management of identities. Reducing local machine accounts helps prevent the proliferation of identities managed outside a central system. Machines are non-compliant if local user accounts exist that are enabled and not listed in the policy parameter.	AuditIfNotExists, Disabled	2.0.0
Windows virtual machine scale sets should have Azure Monitor Agent installed	Windows virtual machine scale sets should be monitored and secured through the deployed Azure Monitor Agent. The Azure Monitor Agent collects telemetry data from the guest OS. Virtual machine scale sets with supported OS and in supported regions are monitored for Azure Monitor Agent deployment. Learn more: https://aka.ms/AMAOverview .	AuditIfNotExists, Disabled	3.2.0
Windows virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	Although a virtual machine's OS and data disks are encrypted-at-rest by default using platform managed keys; resource disks (temp disks), data caches, and data flowing between Compute and Storage resources are not encrypted. Use Azure Disk Encryption or EncryptionAtHost to remediate. Visit https://aka.ms/diskencryptioncomparison to compare encryption offerings. This policy requires two prerequisites to be deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol .	AuditIfNotExists, Disabled	1.1.1
Windows virtual machines should have Azure Monitor Agent installed	Windows virtual machines should be monitored and secured through the deployed Azure Monitor Agent. The Azure Monitor Agent collects telemetry data from the guest OS. Windows virtual machines with supported OS and in supported regions are monitored for Azure Monitor Agent deployment. Learn more: https://aka.ms/AMAOverview .	AuditIfNotExists, Disabled	3.2.0

Microsoft.VirtualMachineImages

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
VM Image Builder templates should use private link	Azure Private Link lets you connect your virtual network to Azure services without a public IP address at the source or destination. The Private Link platform handles the connectivity between the consumer and services over the Azure backbone network. By mapping private endpoints to your VM Image Builder building resources, data leakage risks are reduced. Learn more about private links at:	Audit, Disabled, Deny	1.1.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
	https://docs.microsoft.com/azure/virtual-machines/linux/image-builder-networking#deploy-using-an-existing-vnet.		

Microsoft.ClassicCompute

Expand table

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
A vulnerability assessment solution should be enabled on your virtual machines	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.	AuditIfNotExists, Disabled	3.0.0
Adaptive application controls for defining safe applications should be enabled on your machines	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This helps harden your machines against malware. To simplify the process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.	AuditIfNotExists, Disabled	3.0.0
All network ports should be restricted on network security groups associated to your virtual machine	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0
Allowlist rules in your adaptive application control policy should be updated	Monitor for changes in behavior on groups of machines configured for auditing by Azure Security Center's adaptive application controls. Security Center uses machine learning to analyze the running processes on your machines and suggest a list of known-safe applications. These are presented as recommended apps to allow in adaptive application control policies.	AuditIfNotExists, Disabled	3.0.0
Audit virtual machines without disaster recovery configured	Audit virtual machines which do not have disaster recovery configured. To learn more about disaster recovery, visit https://aka.ms/asr-doc.	auditIfNotExists	1.0.0
Endpoint protection health issues should be resolved on your machines	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection.	AuditIfNotExists, Disabled	1.0.0
Endpoint protection should be installed on your machines	To protect your machines from threats and vulnerabilities, install a supported endpoint protection solution.	AuditIfNotExists, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Internet-facing virtual machines should be protected with network security groups	Protect your virtual machines from potential threats by restricting access to them with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc	AuditIfNotExists, Disabled	3.0.0
IP Forwarding on your virtual machine should be disabled	Enabling IP forwarding on a virtual machine's NIC allows the machine to receive traffic addressed to other destinations. IP forwarding is rarely required (e.g., when using the VM as a network virtual appliance), and therefore, this should be reviewed by the network security team.	AuditIfNotExists, Disabled	3.0.0
Machines should have secret findings resolved	Audits virtual machines to detect whether they contain secret findings from the secret scanning solutions on your virtual machines.	AuditIfNotExists, Disabled	1.0.2
Management ports should be closed on your virtual machines	Open remote management ports are exposing your VM to a high level of risk from Internet-based attacks. These attacks attempt to brute force credentials to gain admin access to the machine.	AuditIfNotExists, Disabled	3.0.0
Monitor missing Endpoint Protection in Azure Security Center	Servers without an installed Endpoint Protection agent will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	3.0.0
Non-internet-facing virtual machines should be protected with network security groups	Protect your non-internet-facing virtual machines from potential threats by restricting access with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc	AuditIfNotExists, Disabled	3.0.0
System updates should be installed on your machines	Missing security system updates on your servers will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	4.0.0
Virtual machines should be migrated to new Azure Resource Manager resources	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance, access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management	Audit, Deny, Disabled	1.0.0
Vulnerabilities in container security configurations should be remediated	Audit vulnerabilities in security configuration on machines with Docker installed and display as recommendations in Azure Security Center.	AuditIfNotExists, Disabled	3.0.0
Vulnerabilities in security configuration on your machines should be remediated	Servers which do not satisfy the configured baseline will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	3.1.0

Next steps

- See the built-ins on the Azure Policy GitHub repo.
- Review the Azure Policy definition structure.

- Review Understanding policy effects.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify non-compliant resources using Azure portal

Article • 03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using Azure portal. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

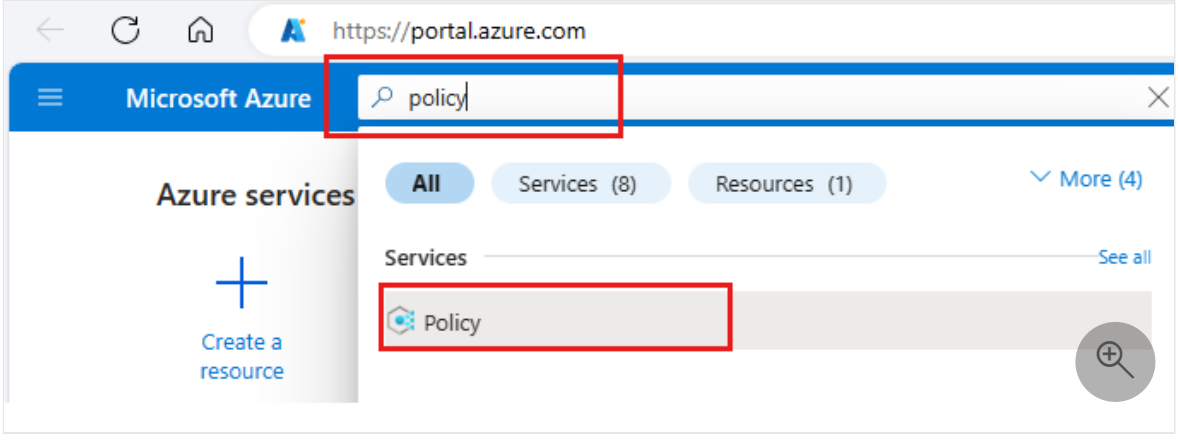
- If you don't have an Azure account, create a free account before you begin.
- A resource group with at least one virtual machine that doesn't use managed disks.

Create a policy assignment

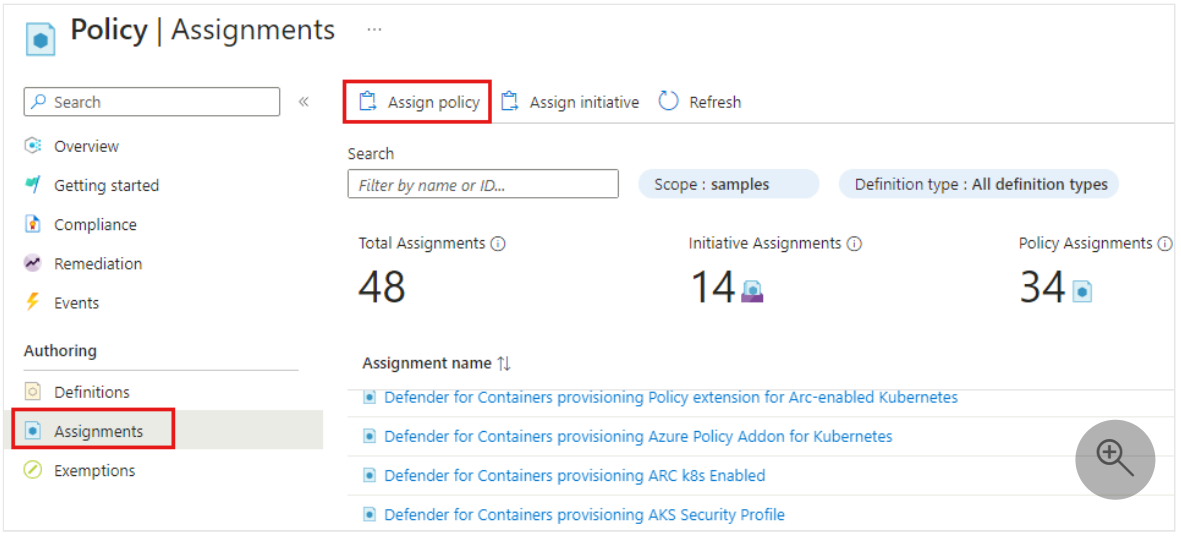
In this quickstart, you create a policy assignment with a built-in policy definition, Audit VMs that do not use managed disks.

1. Sign in to the Azure portal.

2. Search for *policy* and select it from the list.



3. Select **Assignments** on the **Policy** pane.



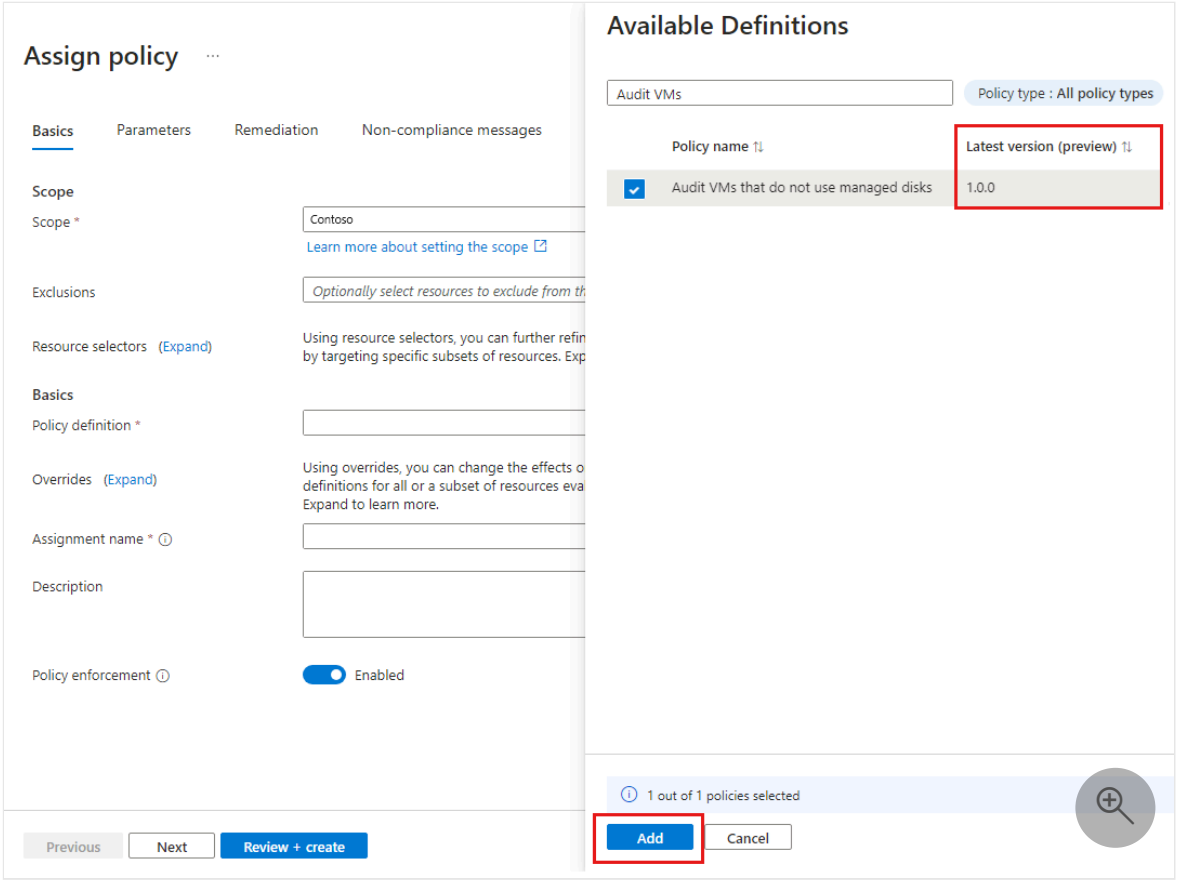
4. Select **Assign Policy** from the **Policy Assignments** pane.

5. On the **Assign Policy** pane **Basics** tab configure the following options:

Expand table

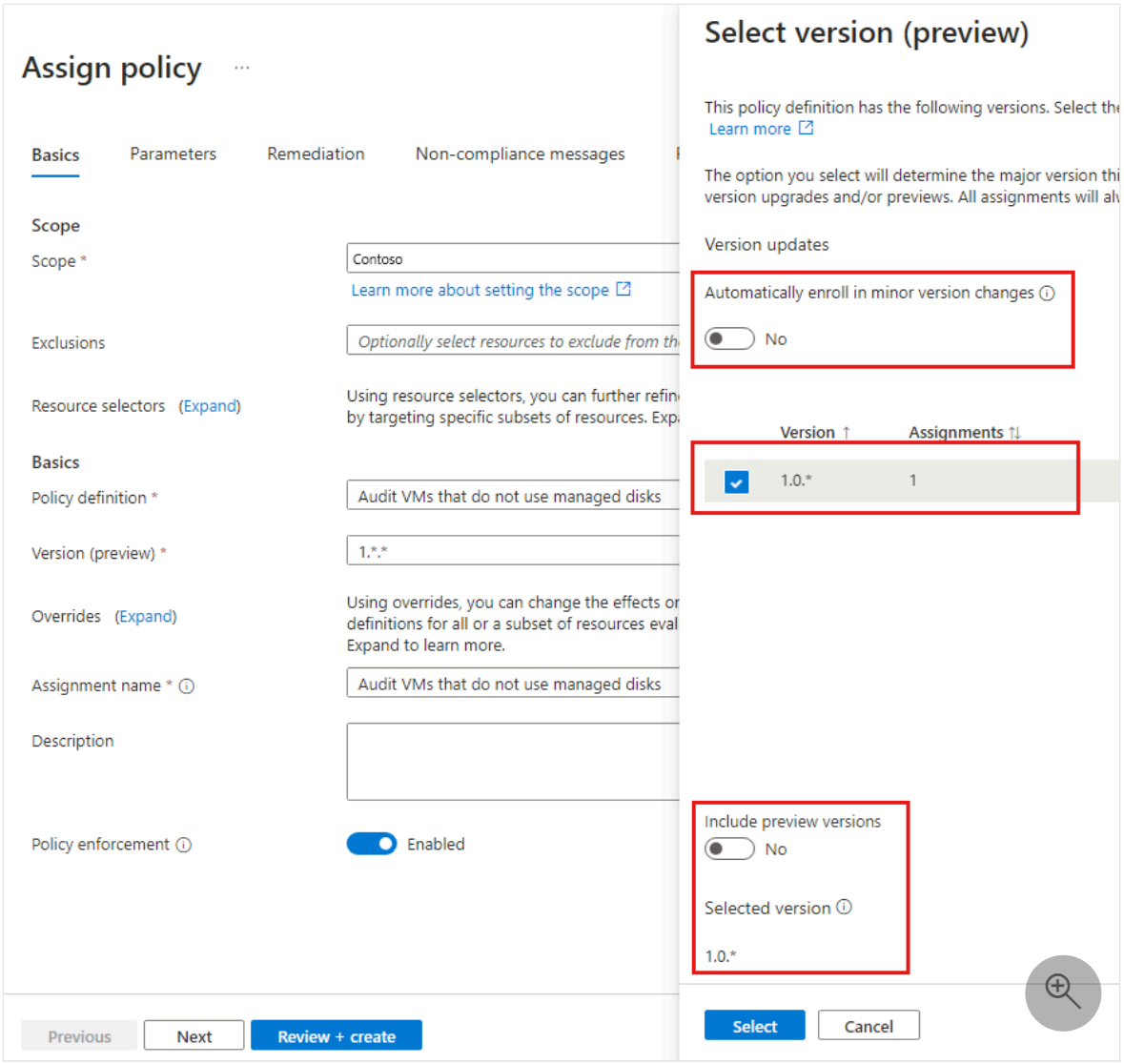
Field	Action
Scope	Use the ellipsis (`...`) and then select a subscription and a resource group. Then choose **Select** to apply the scope.
Exclusions	Optional and isn't used in this example.
Resource selectors	Skip resource selectors for this example. Resource selectors let you refine the resources affected by the policy assignment.
Policy definition	Select the ellipsis (`...`) to open the list of available definitions.
Available Definitions	Search the policy definitions list for *Audit VMs that do not use managed disks* definition, select the policy, and select **Add**. There's a column that shows the latest version of the definition.
Version (preview)	Accept the version in format `1.*.*` to ingest major, minor, and patch versions. Select the ellipsis (`...`) to view available versions and the options to enroll in minor version updates or preview versions. You must select a version to change the options. For more information, go to definition version within assignment.
Assignment name	By default uses the name of the selected policy. You can change it but for this example, use the default name.
Description	Optional to provide details about this policy assignment.

Field	Action
Policy enforcement	Defaults to *Enabled*. For more information, go to enforcement mode.



6. After a Policy definition is selected, you can change the **Version (preview)** options.

   For example, if you select the options shown in the image, the **Version (preview)** is changed to `1.0.*`.



7. Select **Next** to view each tab for **Parameters** and **Remediation**. No changes are needed for this example.

Expand table

Tab name	Options
Parameters	If the policy definition you selected on the **Basics** tab has parameters, you configure them on the **Parameters** tab. This example doesn't use parameters.
Remediation	You can create a managed identity. For this example, **Create a Managed Identity** is unchecked. This box *must* be checked when a policy or initiative includes a policy with either the deployIfNotExists or modify effect. For more information, go to managed identities and how remediation access control works.

8. Select **Next** and on the **Non-compliance messages** tab create a **Non-compliance message** like *Virtual machines should use managed disks*.

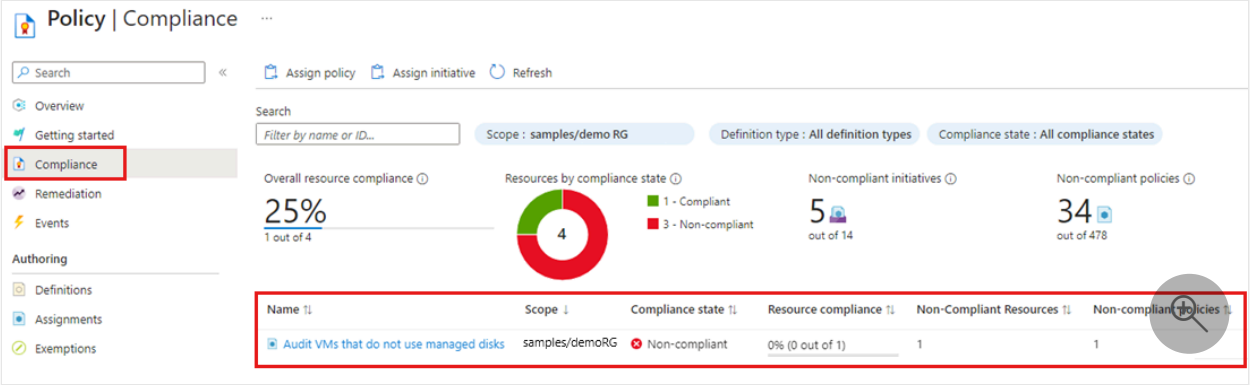
This custom message is displayed when a resource is denied or for non-compliant resources during regular evaluation.

9. Select **Next** and on the **Review + create** tab, review the policy assignment details.

10. Select **Create** to create the policy assignment.

Identify non-compliant resources

On the **Policy** pane, select **Compliance** and locate the *Audit VMs that do not use managed disks* policy assignment. The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.



The policy assignment shows resources that aren't compliant with a **Compliance state** of **Non-compliant**. To get more details, select the policy assignment name to view the **Resource Compliance**.

When a condition is evaluated against your existing resources and found true, then those resources are marked as non-compliant with the policy. The following table shows how different policy effects work with the condition evaluation for the resulting compliance state. Although you don't see the evaluation logic in the Azure portal, the compliance state results are shown. The compliance state result is either compliant or non-compliant.

Expand table

Resource State	Effect	Policy Evaluation	Compliance State
New or Updated	Audit, Modify, AuditIfNotExist	True	Non-Compliant
New or Updated	Audit, Modify, AuditIfNotExist	False	Compliant

Resource State	Effect	Policy Evaluation	Compliance State
Exists	Deny, Audit, Append, Modify, DeployIfNotExist, AuditIfNotExist	True	Non-Compliant
Exists	Deny, Audit, Append, Modify, DeployIfNotExist, AuditIfNotExist	False	Compliant

The `DeployIfNotExist` and `AuditIfNotExist` effects require the `IF` statement to be `TRUE` and the existence condition to be `FALSE` to be non-compliant. When `TRUE`, the `IF` condition triggers evaluation of the existence condition for the related resources.

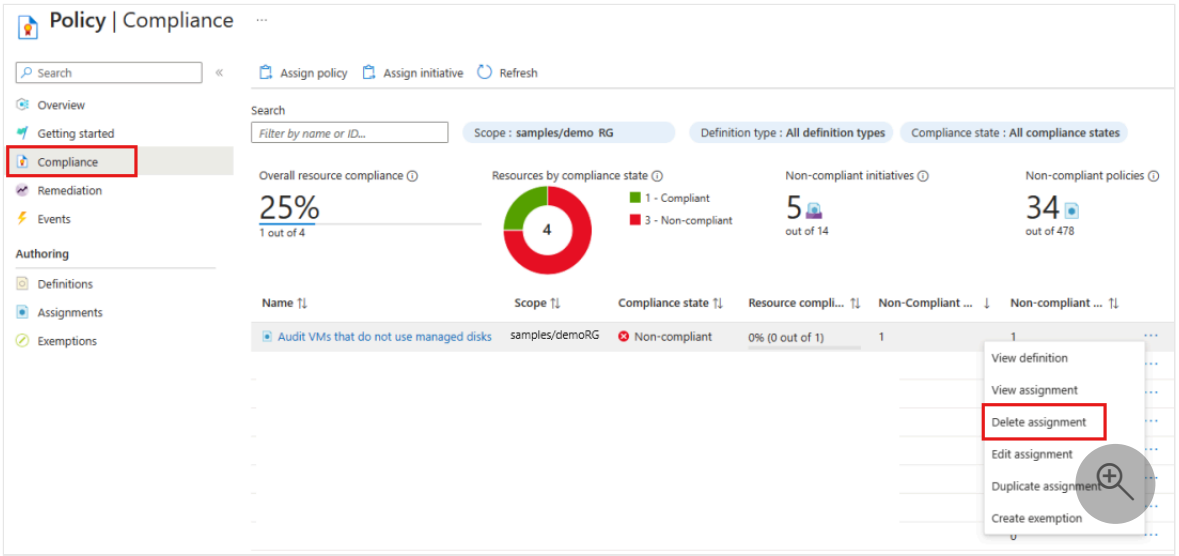
Clean up resources

You can delete a policy assignment from **Compliance** or from **Assignments**.

To remove the policy assignment created in this article, follow these steps:

1. On the **Policy** pane, select **Compliance** and locate the *Audit VMs that do not use managed disks* policy assignment.

2. Select the policy assignment's ellipsis and select **Delete assignment**.



Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify non-compliant resources using Azure CLI

Article • 03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using Azure CLI. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

Azure CLI is used to create and manage Azure resources from the command line or in scripts. This guide uses Azure CLI to create a policy assignment and to identify non-compliant resources in your Azure environment.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Azure CLI.
- Visual Studio Code.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Connect to Azure

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

Azure CLI

```
az login
```

```
# Run these commands if you have multiple subscriptions
az account list --output table
az account set --subscription <subscriptionID>
```

Register resource provider

When a resource provider is registered, it's available to use in your Azure subscription.

To verify if `Microsoft.PolicyInsights` is registered, run `Get-AzResourceProvider`. The resource provider contains several resource types. If the result is `NotRegistered` run `Register-AzResourceProvider`:

Azure CLI

```
az provider show \
  --namespace Microsoft.PolicyInsights \
  --query "{Provider:namespace,State:registrationState}" \
  --output table

az provider register --namespace Microsoft.PolicyInsights
```

The Azure CLI commands use a backslash (`\`) for line continuation to improve readability. For more information, go to az provider.

Create policy assignment

Use the following commands to create a new policy assignment for your resource group. This example uses an existing resource group that contains a virtual machine *without* managed disks. The resource group is the scope for the policy assignment. This example uses the built-in policy definition Audit VMs that do not use managed disks.

Run the following commands and replace `<resourceGroupName>` with your resource group name:

Azure PowerShell

```
rgid=$(az group show --resource-group <resourceGroupName> --query id --
output tsv)

definition=$(az policy definition list \
  --query "[?displayName=='Audit VMs that do not use managed disks']".name \
  --output tsv)
```

The `rgid` variable stores the resource group ID. The `definition` variable stores the policy definition's name, which is a GUID.

Run the following command to create the policy assignment:

Azure CLI

```
az policy assignment create \
  --name 'audit-vm-managed-disks' \
  --display-name 'Audit VM managed disks' \
  --scope $rgid \
  --policy $definition \
  --description 'Azure CLI policy assignment to resource group'
```

- `name` creates the policy assignment name used in the assignment's `ResourceId`.
- `display-name` is the name for the policy assignment and is visible in Azure portal.
- `scope` uses the `$rgid` variable to assign the policy to the resource group.
- `policy` assigns the policy definition stored in the `$definition` variable.
- `description` can be used to add context about the policy assignment.

The results of the policy assignment resemble the following example:

Output

```
"description": "Azure CLI policy assignment to resource group",
"displayName": "Audit VM managed disks",
"enforcementMode": "Default",
"id":
"/subscriptions/{subscriptionID}/resourceGroups/{resourceGroupName}/provider
s/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks",
"identity": null,
"location": null,
"metadata": {
  "createdBy": "11111111-1111-1111-1111-111111111111",
  "createdOn": "2024-02-23T18:42:27.4780803Z",
  "updatedBy": null,
  "updatedOn": null
},
"name": "audit-vm-managed-disks",
```

If you want to redisplay the policy assignment information, run the following command:

Azure CLI

```
az policy assignment show --name "audit-vm-managed-disks" --scope $rgid
```

For more information, go to az policy assignment.

Identify non-compliant resources

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

Use the following command to identify resources that aren't compliant with the policy assignment you created:

Azure CLI

```
policyid=$(az policy assignment show \
  --name "audit-vm-managed-disks" \
  --scope $rgid \
  --query id \
  --output tsv)

az policy state list --resource $policyid --filter "(isCompliant eq false)"
```

The `policyid` variable uses an expression to get the policy assignment's ID. The `filter` parameter limits the output to non-compliant resources.

The `az policy state list` output is verbose, but for this article the `complianceState` shows `NonCompliant`:

Output

```
"complianceState": "NonCompliant",
"components": null,
"effectiveParameters": "",
"isCompliant": false,
```

For more information, go to az policy state.

Clean up resources

To remove the policy assignment, run the following command:

Azure CLI

```
az policy assignment delete --name "audit-vm-managed-disks" --scope $rgid
```

To sign out of your Azure CLI session:

Azure CLI

```
az logout
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify noncompliant resources using Terraform

Article • 04/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. This quickstart steps you through the process of creating a policy assignment to identify virtual machines that are not using managed disks.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Terraform enables the definition, preview, and deployment of cloud infrastructure. Using Terraform, you create configuration files using HCL syntax. The HCL syntax allows you to specify the cloud provider - such as Azure - and the elements that make up your cloud infrastructure. After you create your configuration files, you create an *execution plan* that allows you to preview your infrastructure changes before they're deployed. Once you verify the changes, you apply the execution plan to deploy the infrastructure.

In this article, you learn how to:

- ✓ Retrieve the current Azure client configuration.
- ✓ Create a Azure resource group with the generated random name.
- ✓ Create Subscription Policy Assignment to identify virtual machines that aren't using managed disks

Prerequisites

- Create an Azure account with an active subscription. You can create an account for free.
- Install and configure Terraform

Implement the Terraform code

The sample code for this article is located in the Azure Terraform GitHub repo. You can view the log file containing the test results from current and previous versions of

Terraform. See more articles and sample code showing how to use Terraform to manage Azure resources

1. Create a directory in which to test and run the sample Terraform code, and make it the current directory.

2. Create a file named `providers.tf` and insert the following code.

```Terraform
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~>4.0"
    }
    random = {
      source  = "hashicorp/random"
      version = "~>3.0"
    }
  }
}

provider "azurerm" {
  features {}
}
```

3. Create a file named `main.tf` and insert the following code.

```Terraform
# Create a random pet name to ensure unique resource group name
resource "random_pet" "rg_name" {
  prefix = var.resource_group_name_prefix
}

# Create a resource group
resource "azurerm_resource_group" "example" {
  location = var.resource_group_location
  name     = random_pet.rg_name.id
}

# Get the current subscription
data "azurerm_subscription" "current" {}

# Create a subscription policy assignment
resource "azurerm_subscription_policy_assignment" "auditvms" {
  name                 = "audit-vm-manageddisks"
  subscription_id      = coalesce(var.scope,
"/subscriptions/${data.azurerm_subscription.current.subscription_id}")
  policy_definition_id =
```

```
"/providers/Microsoft.Authorization/policyDefinitions/06a78e20-9358-41c9-923c-fb736d382a4d"
  description          = "Shows all virtual machines not using managed disks"
  display_name         = "Audit VMs without managed disks assignment"
}
```

4. Create a file named `variables.tf` and insert the following code.

Terraform

```
variable "resource_group_location" {
  type        = string
  default     = "eastus"
  description = "Location of the resource group."
}

variable "resource_group_name_prefix" {
  type        = string
  default     = "rg"
  description = "Prefix of the resource group name that's combined with a random ID so name is unique in your Azure subscription."
}

variable "scope" {
  type        = string
  default     = ""
  description = "Scope of the policy assignment."
}
```

5. Create a file named `outputs.tf` and insert the following code.

Terraform

```
output "resource_group_name" {
  value = azurerm_resource_group.example.name
}

output "assignment_id" {
  value = azurerm_subscription_policy_assignment.auditvms.id
}

output "subscription_id" {
  value = data.azurerm_subscription.current.subscription_id
}
```

Specify scope

A scope determines what resources or grouping of resources the policy assignment gets enforced on. It could range from a management group to an individual resource. To use any of the following scopes, update the `scope` variable in the `variables.tf` file. If you leave the `scope` variable value blank, the "subscription" scope is used.

- Subscription: `/subscriptions/<subscription_id>`
- Resource group: `/subscriptions/<subscription_id>/resourceGroups/<resource_group_name>`
- Resource: `/subscriptions/<subscription_id>/resourceGroups/<resource_group_name>/providers/<resource_provider_namespace>/[{parentResourcePath}/]`

Initialize Terraform

> **ⓘ Important**
>
> If you are using the 4.x azurerm provider, you must explicitly specify the Azure subscription ID to authenticate to Azure before running the Terraform commands.
>
> One way to specify the Azure subscription ID without putting it in the `providers` block is to specify the subscription ID in an environment variable named `ARM_SUBSCRIPTION_ID`.
>
> For more information, see the Azure provider reference documentation.

Run terraform init to initialize the Terraform deployment. This command downloads the Azure provider required to manage your Azure resources.

```
Console
terraform init -upgrade
```

Key points:

- The `-upgrade` parameter upgrades the necessary provider plugins to the newest version that complies with the configuration's version constraints.

Create a Terraform execution plan

Run terraform plan to create an execution plan.

```
Console

terraform plan -out main.tfplan
```

Key points:

- The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
- The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

Apply a Terraform execution plan

Run terraform apply to apply the execution plan to your cloud infrastructure.

```
Console

terraform apply main.tfplan
```

Key points:

- The example `terraform apply` command assumes you previously ran `terraform plan -out main.tfplan`.
- If you specified a different filename for the `-out` parameter, use that same filename in the call to `terraform apply`.
- If you didn't use the `-out` parameter, call `terraform apply` without any parameters.

Verify the results

1. Get the `_assignment\_id_` returned by `terraform apply`.

2. run the following command to view the resources that are not compliant under your new policy assignment.

   ```
   Console
   ```

```
armclient post
"/subscriptions/<subscription_id>/providers/Microsoft.PolicyInsights/po
licyStates/latest/queryResults?api-version=2019-10-
01&$filter=IsCompliant eq false and PolicyAssignmentId eq
'<policyAssignmentID>'&$apply=groupby((ResourceId))" > <json file to
direct the output with the resource IDs into>
```

3. The results are comparable to what you see listed under **Noncompliant resources** in the Azure portal view.

Clean up resources

When you no longer need the resources created via Terraform, do the following steps:

1. Run terraform plan and specify the `destroy` flag.

   Console

   ```
   terraform plan -destroy -out main.destroy.tfplan
   ```

   Key points:

   - The `terraform plan` command creates an execution plan, but doesn't execute it. Instead, it determines what actions are necessary to create the configuration specified in your configuration files. This pattern allows you to verify whether the execution plan matches your expectations before making any changes to actual resources.
   - The optional `-out` parameter allows you to specify an output file for the plan. Using the `-out` parameter ensures that the plan you reviewed is exactly what is applied.

2. Run terraform apply to apply the execution plan.

   Console

   ```
   terraform apply main.destroy.tfplan
   ```

Troubleshoot Terraform on Azure

Troubleshoot common problems when using Terraform on Azure.

Next steps

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify non-compliant resources using Azure PowerShell

Article • 03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using Azure PowerShell. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

The Azure PowerShell modules can be used to manage Azure resources from the command line or in scripts. This article explains how to use Azure PowerShell to create a policy assignment.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Azure PowerShell.
- Visual Studio Code.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Connect to Azure

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

Azure PowerShell

```
Connect-AzAccount
```

```
# Run these commands if you have multiple subscriptions
Get-AzSubScription
Set-AzContext -Subscription <subscriptionID>
```

Register resource provider

When a resource provider is registered, it's available to use in your Azure subscription.

To verify if `Microsoft.PolicyInsights` is registered, run `Get-AzResourceProvider`. The resource provider contains several resource types. If the result is `NotRegistered` run `Register-AzResourceProvider`:

Azure PowerShell

```
 Get-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights' |
   Select-Object -Property ResourceTypes, RegistrationState

Register-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights'
```

For more information, go to Get-AzResourceProvider and Register-AzResourceProvider.

Create policy assignment

Use the following commands to create a new policy assignment for your resource group. This example uses an existing resource group that contains a virtual machine *without* managed disks. The resource group is the scope for the policy assignment. This example uses the built-in policy definition Audit VMs that do not use managed disks.

Run the following commands and replace `<resourceGroupName>` with your resource group name:

Azure PowerShell

```
$rg = Get-AzResourceGroup -Name '<resourceGroupName>'

$definition = Get-AzPolicyDefinition |
  Where-Object { $_.DisplayName -eq 'Audit VMs that do not use managed
disks' }
```

The `$rg` variable stores properties for the resource group and the `$definition` variable stores the policy definition's properties. The properties are used in subsequent commands.

Run the following command to create the policy assignment:

Azure PowerShell

```
$policyparms = @{
Name = 'audit-vm-managed-disks'
DisplayName = 'Audit VM managed disks'
Scope = $rg.ResourceId
PolicyDefinition = $definition
Description = 'Az PowerShell policy assignment to resource group'
}

New-AzPolicyAssignment @policyparms
```

The `$policyparms` variable uses splatting to create parameter values and improve readability. The `New-AzPolicyAssignment` command uses the parameter values defined in the `$policyparms` variable.

- `Name` creates the policy assignment name used in the assignment's `ResourceId`.
- `DisplayName` is the name for the policy assignment and is visible in Azure portal.
- `Scope` uses the `$rg.ResourceId` property to assign the policy to the resource group.
- `PolicyDefinition` assigns the policy definition stored in the `$definition` variable.
- `Description` can be used to add context about the policy assignment.

The results of the policy assignment resemble the following example:

Output

```
Name               : audit-vm-managed-disks
ResourceId         :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers
/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks
ResourceName       : audit-vm-managed-disks
ResourceGroupName  : {resourceGroupName}
ResourceType       : Microsoft.Authorization/policyAssignments
SubscriptionId     : {subscriptionId}
PolicyAssignmentId :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers
/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks
Properties         :
Microsoft.Azure.Commands.ResourceManager.Cmdlets.Implementation.Policy.PsPol
icyAssignmentProperties
```

For more information, go to New-AzPolicyAssignment.

If you want to redisplay the policy assignment information, run the following command:

Azure PowerShell

```
Get-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

Identify non-compliant resources

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

Use the following command to identify resources that aren't compliant with the policy assignment you created:

Azure PowerShell

```
$complianceparms = @{
ResourceGroupName = $rg.ResourceGroupName
PolicyAssignmentName = 'audit-vm-managed-disks'
Filter = 'IsCompliant eq false'
}

Get-AzPolicyState @complianceparms
```

The `$complianceparms` variable uses splatting to create parameter values used in the `Get-AzPolicyState` command.

- `ResourceGroupName` gets the resource group name from the `$rg.ResourceGroupName` property.
- `PolicyAssignmentName` specifies the name used when the policy assignment was created.
- `Filter` uses an expression to find resources that aren't compliant with the policy assignment.

Your results resemble the following example and `ComplianceState` shows `NonCompliant`:

Output

```
Timestamp                : 2/14/2024 18:25:37
ResourceId               :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers
/microsoft.compute/virtualmachines/{vmId}
PolicyAssignmentId       :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers
/microsoft.authorization/policyassignments/audit-vm-managed-disks
PolicyDefinitionId       :
/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-
```

```
923c-fb736d382a4d
IsCompliant               : False
SubscriptionId            : {subscriptionId}
ResourceType              : Microsoft.Compute/virtualMachines
ResourceLocation          : {location}
ResourceGroup             : {resourceGroupName}
ResourceTags              : tbd
PolicyAssignmentName      : audit-vm-managed-disks
PolicyAssignmentOwner     : tbd
PolicyAssignmentScope     :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}
PolicyDefinitionName      : 06a78e20-9358-41c9-923c-fb736d382a4d
PolicyDefinitionAction    : audit
PolicyDefinitionCategory  : tbd
ManagementGroupIds        : {managementGroupId}
ComplianceState           : NonCompliant
AdditionalProperties      : {[complianceReasonCode, ]}
```

For more information, go to Get-AzPolicyState.

Clean up resources

To remove the policy assignment, run the following command:

Azure PowerShell

```
Remove-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope
$rg.ResourceId
```

To sign out of your Azure PowerShell session:

Azure PowerShell

```
Disconnect-AzAccount
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify non-compliant resources using Azure PowerShell

03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using Azure PowerShell. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

The Azure PowerShell modules can be used to manage Azure resources from the command line or in scripts. This article explains how to use Azure PowerShell to create a policy assignment.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Azure PowerShell.
- Visual Studio Code.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Connect to Azure

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

Azure PowerShell

```
Connect-AzAccount

# Run these commands if you have multiple subscriptions
```

```
Get-AzSubScription
Set-AzContext -Subscription <subscriptionID>
```

Register resource provider

When a resource provider is registered, it's available to use in your Azure subscription.

To verify if `Microsoft.PolicyInsights` is registered, run `Get-AzResourceProvider`. The resource provider contains several resource types. If the result is `NotRegistered` run `Register-AzResourceProvider`:

Azure PowerShell

```
Get-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights' |
  Select-Object -Property ResourceTypes, RegistrationState

Register-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights'
```

For more information, go to Get-AzResourceProvider and Register-AzResourceProvider.

Create policy assignment

Use the following commands to create a new policy assignment for your resource group. This example uses an existing resource group that contains a virtual machine *without* managed disks. The resource group is the scope for the policy assignment. This example uses the built-in policy definition Audit VMs that do not use managed disks.

Run the following commands and replace `<resourceGroupName>` with your resource group name:

Azure PowerShell

```
$rg = Get-AzResourceGroup -Name '<resourceGroupName>'

$definition = Get-AzPolicyDefinition |
  Where-Object { $_.DisplayName -eq 'Audit VMs that do not use managed disks' }
```

The `$rg` variable stores properties for the resource group and the `$definition` variable stores the policy definition's properties. The properties are used in subsequent commands.

Run the following command to create the policy assignment:

Azure PowerShell

```
$policyparms = @{
Name = 'audit-vm-managed-disks'
DisplayName = 'Audit VM managed disks'
Scope = $rg.ResourceId
PolicyDefinition = $definition
Description = 'Az PowerShell policy assignment to resource group'
}

New-AzPolicyAssignment @policyparms
```

The `$policyparms` variable uses splatting to create parameter values and improve readability. The `New-AzPolicyAssignment` command uses the parameter values defined in the `$policyparms` variable.

- `Name` creates the policy assignment name used in the assignment's `ResourceId`.
- `DisplayName` is the name for the policy assignment and is visible in Azure portal.
- `Scope` uses the `$rg.ResourceId` property to assign the policy to the resource group.
- `PolicyDefinition` assigns the policy definition stored in the `$definition` variable.
- `Description` can be used to add context about the policy assignment.

The results of the policy assignment resemble the following example:

Output

```
Name               : audit-vm-managed-disks
ResourceId         :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Micro
soft.Authorization/policyAssignments/audit-vm-managed-disks
ResourceName       : audit-vm-managed-disks
ResourceGroupName  : {resourceGroupName}
ResourceType       : Microsoft.Authorization/policyAssignments
SubscriptionId     : {subscriptionId}
PolicyAssignmentId :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Micro
soft.Authorization/policyAssignments/audit-vm-managed-disks
Properties         :
Microsoft.Azure.Commands.ResourceManager.Cmdlets.Implementation.Policy.PsPolicyAss
ignmentProperties
```

For more information, go to New-AzPolicyAssignment.

If you want to redisplay the policy assignment information, run the following command:

Azure PowerShell

```
Get-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

Identify non-compliant resources

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

Use the following command to identify resources that aren't compliant with the policy assignment you created:

Azure PowerShell

```
$complianceparms = @{
ResourceGroupName = $rg.ResourceGroupName
PolicyAssignmentName = 'audit-vm-managed-disks'
Filter = 'IsCompliant eq false'
}

Get-AzPolicyState @complianceparms
```

The `$complianceparms` variable uses splatting to create parameter values used in the `Get-AzPolicyState` command.

- `ResourceGroupName` gets the resource group name from the `$rg.ResourceGroupName` property.
- `PolicyAssignmentName` specifies the name used when the policy assignment was created.
- `Filter` uses an expression to find resources that aren't compliant with the policy assignment.

Your results resemble the following example and `ComplianceState` shows `NonCompliant`:

Output

```
Timestamp                : 2/14/2024 18:25:37
ResourceId               :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/micro
soft.compute/virtualmachines/{vmId}
PolicyAssignmentId       :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/micro
soft.authorization/policyassignments/audit-vm-managed-disks
PolicyDefinitionId       :
/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-923c-
fb736d382a4d
IsCompliant              : False
SubscriptionId           : {subscriptionId}
ResourceType             : Microsoft.Compute/virtualMachines
ResourceLocation         : {location}
ResourceGroup            : {resourceGroupName}
ResourceTags             : tbd
PolicyAssignmentName     : audit-vm-managed-disks
```

```
PolicyAssignmentOwner    : tbd
PolicyAssignmentScope    :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}
PolicyDefinitionName     : 06a78e20-9358-41c9-923c-fb736d382a4d
PolicyDefinitionAction   : audit
PolicyDefinitionCategory : tbd
ManagementGroupIds       : {managementGroupId}
ComplianceState          : NonCompliant
AdditionalProperties     : {[complianceReasonCode, ]}
```

For more information, go to Get-AzPolicyState.

Clean up resources

To remove the policy assignment, run the following command:

Azure PowerShell

```
Remove-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

To sign out of your Azure PowerShell session:

Azure PowerShell

```
Disconnect-AzAccount
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Quickstart: Create a policy assignment to identify non-compliant resources using Azure PowerShell

03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using Azure PowerShell. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

The Azure PowerShell modules can be used to manage Azure resources from the command line or in scripts. This article explains how to use Azure PowerShell to create a policy assignment.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Azure PowerShell.
- Visual Studio Code.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Connect to Azure

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

Azure PowerShell

```
Connect-AzAccount

# Run these commands if you have multiple subscriptions
```

```
Get-AzSubScription
Set-AzContext -Subscription <subscriptionID>
```

Register resource provider

When a resource provider is registered, it's available to use in your Azure subscription.

To verify if `Microsoft.PolicyInsights` is registered, run `Get-AzResourceProvider`. The resource provider contains several resource types. If the result is `NotRegistered` run `Register-AzResourceProvider`:

Azure PowerShell

```
Get-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights' |
  Select-Object -Property ResourceTypes, RegistrationState

Register-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights'
```

For more information, go to Get-AzResourceProvider and Register-AzResourceProvider.

Create policy assignment

Use the following commands to create a new policy assignment for your resource group. This example uses an existing resource group that contains a virtual machine *without* managed disks. The resource group is the scope for the policy assignment. This example uses the built-in policy definition Audit VMs that do not use managed disks.

Run the following commands and replace `<resourceGroupName>` with your resource group name:

Azure PowerShell

```
$rg = Get-AzResourceGroup -Name '<resourceGroupName>'

$definition = Get-AzPolicyDefinition |
  Where-Object { $_.DisplayName -eq 'Audit VMs that do not use managed disks' }
```

The `$rg` variable stores properties for the resource group and the `$definition` variable stores the policy definition's properties. The properties are used in subsequent commands.

Run the following command to create the policy assignment:

Azure PowerShell

```
$policyparms = @{
Name = 'audit-vm-managed-disks'
DisplayName = 'Audit VM managed disks'
Scope = $rg.ResourceId
PolicyDefinition = $definition
Description = 'Az PowerShell policy assignment to resource group'
}

New-AzPolicyAssignment @policyparms
```

The `$policyparms` variable uses splatting to create parameter values and improve readability. The `New-AzPolicyAssignment` command uses the parameter values defined in the `$policyparms` variable.

- `Name` creates the policy assignment name used in the assignment's `ResourceId`.
- `DisplayName` is the name for the policy assignment and is visible in Azure portal.
- `Scope` uses the `$rg.ResourceId` property to assign the policy to the resource group.
- `PolicyDefinition` assigns the policy definition stored in the `$definition` variable.
- `Description` can be used to add context about the policy assignment.

The results of the policy assignment resemble the following example:

Output

```
Name               : audit-vm-managed-disks
ResourceId         :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Micro
soft.Authorization/policyAssignments/audit-vm-managed-disks
ResourceName       : audit-vm-managed-disks
ResourceGroupName  : {resourceGroupName}
ResourceType       : Microsoft.Authorization/policyAssignments
SubscriptionId     : {subscriptionId}
PolicyAssignmentId :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Micro
soft.Authorization/policyAssignments/audit-vm-managed-disks
Properties         :
Microsoft.Azure.Commands.ResourceManager.Cmdlets.Implementation.Policy.PsPolicyAss
ignmentProperties
```

For more information, go to New-AzPolicyAssignment.

If you want to redisplay the policy assignment information, run the following command:

Azure PowerShell

```
Get-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

Identify non-compliant resources

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

Use the following command to identify resources that aren't compliant with the policy assignment you created:

Azure PowerShell

```
$complianceparms = @{
ResourceGroupName = $rg.ResourceGroupName
PolicyAssignmentName = 'audit-vm-managed-disks'
Filter = 'IsCompliant eq false'
}

Get-AzPolicyState @complianceparms
```

The `$complianceparms` variable uses splatting to create parameter values used in the `Get-AzPolicyState` command.

- `ResourceGroupName` gets the resource group name from the `$rg.ResourceGroupName` property.
- `PolicyAssignmentName` specifies the name used when the policy assignment was created.
- `Filter` uses an expression to find resources that aren't compliant with the policy assignment.

Your results resemble the following example and `ComplianceState` shows `NonCompliant`:

Output

```
Timestamp                : 2/14/2024 18:25:37
ResourceId               :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/micro
soft.compute/virtualmachines/{vmId}
PolicyAssignmentId       :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/micro
soft.authorization/policyassignments/audit-vm-managed-disks
PolicyDefinitionId       :
/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-923c-
fb736d382a4d
IsCompliant              : False
SubscriptionId           : {subscriptionId}
ResourceType             : Microsoft.Compute/virtualMachines
ResourceLocation         : {location}
ResourceGroup            : {resourceGroupName}
ResourceTags             : tbd
PolicyAssignmentName     : audit-vm-managed-disks
```

```
PolicyAssignmentOwner    : tbd
PolicyAssignmentScope    :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}
PolicyDefinitionName     : 06a78e20-9358-41c9-923c-fb736d382a4d
PolicyDefinitionAction   : audit
PolicyDefinitionCategory : tbd
ManagementGroupIds       : {managementGroupId}
ComplianceState          : NonCompliant
AdditionalProperties     : {[complianceReasonCode, ]}
```

For more information, go to Get-AzPolicyState.

Clean up resources

To remove the policy assignment, run the following command:

Azure PowerShell

```
Remove-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

To sign out of your Azure PowerShell session:

Azure PowerShell

```
Disconnect-AzAccount
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Quickstart: Create a policy assignment to identify non-compliant resources using Azure PowerShell

03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using Azure PowerShell. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

The Azure PowerShell modules can be used to manage Azure resources from the command line or in scripts. This article explains how to use Azure PowerShell to create a policy assignment.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Azure PowerShell.
- Visual Studio Code.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Connect to Azure

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

Azure PowerShell

```
Connect-AzAccount

# Run these commands if you have multiple subscriptions
```

```
Get-AzSubScription
Set-AzContext -Subscription <subscriptionID>
```

Register resource provider

When a resource provider is registered, it's available to use in your Azure subscription.

To verify if `Microsoft.PolicyInsights` is registered, run `Get-AzResourceProvider`. The resource provider contains several resource types. If the result is `NotRegistered` run `Register-AzResourceProvider`:

Azure PowerShell

```
Get-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights' |
  Select-Object -Property ResourceTypes, RegistrationState

Register-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights'
```

For more information, go to Get-AzResourceProvider and Register-AzResourceProvider.

Create policy assignment

Use the following commands to create a new policy assignment for your resource group. This example uses an existing resource group that contains a virtual machine *without* managed disks. The resource group is the scope for the policy assignment. This example uses the built-in policy definition Audit VMs that do not use managed disks.

Run the following commands and replace `<resourceGroupName>` with your resource group name:

Azure PowerShell

```
$rg = Get-AzResourceGroup -Name '<resourceGroupName>'

$definition = Get-AzPolicyDefinition |
  Where-Object { $_.DisplayName -eq 'Audit VMs that do not use managed disks' }
```

The `$rg` variable stores properties for the resource group and the `$definition` variable stores the policy definition's properties. The properties are used in subsequent commands.

Run the following command to create the policy assignment:

Azure PowerShell

```
$policyparms = @{
Name = 'audit-vm-managed-disks'
DisplayName = 'Audit VM managed disks'
Scope = $rg.ResourceId
PolicyDefinition = $definition
Description = 'Az PowerShell policy assignment to resource group'
}

New-AzPolicyAssignment @policyparms
```

The `$policyparms` variable uses splatting to create parameter values and improve readability. The `New-AzPolicyAssignment` command uses the parameter values defined in the `$policyparms` variable.

- `Name` creates the policy assignment name used in the assignment's `ResourceId`.
- `DisplayName` is the name for the policy assignment and is visible in Azure portal.
- `Scope` uses the `$rg.ResourceId` property to assign the policy to the resource group.
- `PolicyDefinition` assigns the policy definition stored in the `$definition` variable.
- `Description` can be used to add context about the policy assignment.

The results of the policy assignment resemble the following example:

Output

```
Name               : audit-vm-managed-disks
ResourceId         :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Micro
soft.Authorization/policyAssignments/audit-vm-managed-disks
ResourceName       : audit-vm-managed-disks
ResourceGroupName  : {resourceGroupName}
ResourceType       : Microsoft.Authorization/policyAssignments
SubscriptionId     : {subscriptionId}
PolicyAssignmentId :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Micro
soft.Authorization/policyAssignments/audit-vm-managed-disks
Properties         :
Microsoft.Azure.Commands.ResourceManager.Cmdlets.Implementation.Policy.PsPolicyAss
ignmentProperties
```

For more information, go to New-AzPolicyAssignment.

If you want to redisplay the policy assignment information, run the following command:

Azure PowerShell

```
Get-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

Identify non-compliant resources

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

Use the following command to identify resources that aren't compliant with the policy assignment you created:

Azure PowerShell

```
$complianceparms = @{
ResourceGroupName = $rg.ResourceGroupName
PolicyAssignmentName = 'audit-vm-managed-disks'
Filter = 'IsCompliant eq false'
}

Get-AzPolicyState @complianceparms
```

The `$complianceparms` variable uses splatting to create parameter values used in the `Get-AzPolicyState` command.

- `ResourceGroupName` gets the resource group name from the `$rg.ResourceGroupName` property.
- `PolicyAssignmentName` specifies the name used when the policy assignment was created.
- `Filter` uses an expression to find resources that aren't compliant with the policy assignment.

Your results resemble the following example and `ComplianceState` shows `NonCompliant`:

Output

```
Timestamp                : 2/14/2024 18:25:37
ResourceId               :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/micro
soft.compute/virtualmachines/{vmId}
PolicyAssignmentId       :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/micro
soft.authorization/policyassignments/audit-vm-managed-disks
PolicyDefinitionId       :
/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-923c-
fb736d382a4d
IsCompliant              : False
SubscriptionId           : {subscriptionId}
ResourceType             : Microsoft.Compute/virtualMachines
ResourceLocation         : {location}
ResourceGroup            : {resourceGroupName}
ResourceTags             : tbd
PolicyAssignmentName     : audit-vm-managed-disks
```

```
PolicyAssignmentOwner    : tbd
PolicyAssignmentScope    :
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}
PolicyDefinitionName     : 06a78e20-9358-41c9-923c-fb736d382a4d
PolicyDefinitionAction   : audit
PolicyDefinitionCategory : tbd
ManagementGroupIds       : {managementGroupId}
ComplianceState          : NonCompliant
AdditionalProperties     : {[complianceReasonCode, ]}
```

For more information, go to Get-AzPolicyState.

Clean up resources

To remove the policy assignment, run the following command:

Azure PowerShell

```
Remove-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope $rg.ResourceId
```

To sign out of your Azure PowerShell session:

Azure PowerShell

```
Disconnect-AzAccount
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Quickstart: Create a policy assignment to identify non-compliant resources with REST API

Article • 03/04/2025

The first step in understanding compliance in Azure is to identify the status of your resources. In this quickstart, you create a policy assignment to identify non-compliant resources using REST API. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

This guide uses REST API to create a policy assignment and to identify non-compliant resources in your Azure environment. The examples in this article use PowerShell and the Azure CLI `az rest` commands. You can also run the `az rest` commands from a Bash shell like Git Bash.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Latest version of PowerShell or a Bash shell like Git Bash.
- Latest version of Azure CLI.
- Visual Studio Code.
- A resource group with at least one virtual machine that doesn't use managed disks.

Review the REST API syntax

There are two elements to run REST API commands: the REST API URI and the request body. For information, go to Policy Assignments - Create.

The following example shows the REST API URI syntax to create a policy definition.

HTTP

```
PUT
https://management.azure.com/{scope}/providers/Microsoft.Authorization/polic
```

```
yAssignments/{policyAssignmentName}?api-version=2023-04-01
```

- `scope`: A scope determines which resources or group of resources the policy assignment gets enforced on. It could range from a management group to an individual resource. Replace `{scope}` with one of the following patterns:
  - Management group: `/providers/Microsoft.Management/managementGroups/{managementGroup}`
  - Subscription: `/subscriptions/{subscriptionId}`
  - Resource group: `/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}`
  - Resource: `/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/{resourceProviderNamespace}/[{parentResourcePath}/]{resourceType}/{resourceName}`
- `policyAssignmentName`: Creates the policy assignment name for your assignment. The name is included in the policy assignment's `policyAssignmentId` property.

The following example is the JSON to create a request body file.

JSON

```
{
  "properties": {
    "displayName": "",
    "description": "",
    "policyDefinitionId":
"/providers/Microsoft.Authorization/policyDefinitions/11111111-1111-1111-
1111-111111111111",
    "nonComplianceMessages": [
      {
        "message": ""
      }
    ]
  }
}
```

- `displayName`: Display name for the policy assignment.
- `description`: Can be used to add context about the policy assignment.
- `policyDefinitionId`: The policy definition ID that to create the assignment.
- `nonComplianceMessages`: Set the message to use when a resource is evaluated as non-compliant. For more information, see assignment non-compliance messages.

Connect to Azure

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

Azure CLI

```azurecli
az login

# Run these commands if you have multiple subscriptions
az account list --output table
az account set --subscription <subscriptionID>
```

Use `az login` even if you're using PowerShell because the examples use Azure CLI az rest commands.

Create a policy assignment

In this example, you create a policy assignment and assign the Audit VMs that do not use managed disks definition.

A request body is needed to create the assignment. Save the following JSON in a file named *request-body.json*.

JSON

```json
{
  "properties": {
    "displayName": "Audit VM managed disks",
    "description": "Policy assignment to resource group scope created with
REST API",
    "policyDefinitionId":
"/providers/Microsoft.Authorization/policyDefinitions/06a78e20-9358-41c9-
923c-fb736d382a4d",
    "nonComplianceMessages": [
      {
        "message": "Virtual machines should use managed disks"
      }
    ]
  }
}
```

To create your policy assignment in an existing resource group scope, use the following REST API URI with a file for the request body. Replace `{subscriptionId}` and `{resourceGroupName}` with your values. The command displays JSON output in your shell.

Azure PowerShell

```
az rest --method put --uri
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Authorization/policyAssignments/audit
-vm-managed-disks?api-version=2023-04-01 --body `@request-body.json
```

In PowerShell, the backtick (`) is needed to escape the `at sign` (`@`) to specify a filename. In a Bash shell like Git Bash, omit the backtick.

For information, go to Policy Assignments - Create.

Identify non-compliant resources

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state. You use REST API to display the non-compliant resources for this policy assignment and the output is in JSON.

To identify non-compliant resources, run the following command. Replace `{subscriptionId}` and `{resourceGroupName}` with your values used when you created the policy assignment.

Azure PowerShell

```
az rest --method post --uri
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.PolicyInsights/policyStates/latest/qu
eryResults?api-version=2019-10-01 --uri-parameters `$filter="complianceState
eq 'NonCompliant' and PolicyAssignmentName eq 'audit-vm-managed-disks'"
```

The `filter` queries for resources that are evaluated as non-compliant with the policy definition named *audit-vm-managed-disks* that you created with the policy assignment. Again, notice the backtick is used to escape the dollar sign (`$`) in the filter. For a Bash client, a backslash (`\`) is a common escape character.

Your results resemble the following example:

JSON

```
{
  "@odata.context":
"https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.PolicyInsights/policyStates/$metadat
a#latest",
  "@odata.count": 1,
  "@odata.nextLink": null,
  "value": [
```

```
    {
      "@odata.context":
"https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/
{resourceGroupName}/providers/Microsoft.PolicyInsights/policyStates/$metadat
a#latest/$entity",
      "@odata.id": null,
      "complianceReasonCode": "",
      "complianceState": "NonCompliant",
      "effectiveParameters": "",
      "isCompliant": false,
      "managementGroupIds": "",
      "policyAssignmentId":
"/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provider
s/microsoft.authorization/policyassignments/audit-vm-managed-disks",
      "policyAssignmentName": "audit-vm-managed-disks",
      "policyAssignmentOwner": "tbd",
      "policyAssignmentParameters": "",
      "policyAssignmentScope":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}",
      "policyAssignmentVersion": "",
      "policyDefinitionAction": "audit",
      "policyDefinitionCategory": "tbd",
      "policyDefinitionGroupNames": [
        ""
      ],
      "policyDefinitionId":
"/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-
923c-fb736d382a4d",
      "policyDefinitionName": "06a78e20-9358-41c9-923c-fb736d382a4d",
      "policyDefinitionReferenceId": "",
      "policyDefinitionVersion": "1.0.0",
      "policySetDefinitionCategory": "",
      "policySetDefinitionId": "",
      "policySetDefinitionName": "",
      "policySetDefinitionOwner": "",
      "policySetDefinitionParameters": "",
      "policySetDefinitionVersion": "",
      "resourceGroup": "{resourceGroupName}",
      "resourceId":
"/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provider
s/microsoft.compute/virtualmachines/{vmName}>",
      "resourceLocation": "westus3",
      "resourceTags": "tbd",
      "resourceType": "Microsoft.Compute/virtualMachines",
      "subscriptionId": "{subscriptionId}",
      "timestamp": "2024-03-26T02:19:28.3720191Z"
    }
  ]
}
```


For more information, go to Policy States - List Query Results For Resource Group.

Clean up resources

To remove the policy assignment, use the following command. Replace `{subscriptionId}` and `{resourceGroupName}` with your values used when you created the policy assignment. The command displays JSON output in your shell.

Azure PowerShell

```
az rest --method delete --uri
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Authorization/policyAssignments/audit
-vm-managed-disks?api-version=2023-04-01
```

You can verify the policy assignment was deleted with the following command. A message is displayed in your shell.

Azure PowerShell

```
az rest --method get --uri
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Authorization/policyAssignments/audit
-vm-managed-disks?api-version=2023-04-01
```

Output

```
The policy assignment 'audit-vm-managed-disks' is not found.
```

For more information, go to Policy Assignments - Delete and Policy Assignments - Get.

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify non-compliant resources by using ARM template

Article • 03/04/2025

In this quickstart, you use an Azure Resource Manager template (ARM template) to create a policy assignment that validates resource's compliance with an Azure policy. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template opens in the Azure portal.



When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Azure PowerShell or Azure CLI.
- Visual Studio Code and the Azure Resource Manager (ARM) Tools.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Review the template

The ARM template creates a policy assignment for a resource group scope and assigns the built-in policy definition Audit VMs that do not use managed disks.

Create the following ARM template as *policy-assignment.json*.

1. Open Visual Studio Code and select **File** > **New Text File**.
2. Copy and paste the ARM template into Visual Studio Code.
3. Select **File** > **Save** and use the filename *policy-assignment.json*.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "policyAssignmentName": {
      "type": "string",
      "defaultValue": "audit-vm-managed-disks",
      "metadata": {
        "description": "Policy assignment name used in assignment's resource
ID"
      }
    },
    "policyDefinitionID": {
      "type": "string",
      "defaultValue":
"/providers/Microsoft.Authorization/policyDefinitions/06a78e20-9358-41c9-
923c-fb736d382a4d",
      "metadata": {
        "description": "Policy definition ID"
      }
    },
    "policyDisplayName": {
      "type": "string",
      "defaultValue": "Audit VM managed disks",
      "metadata": {
        "description": "Display name for Azure portal"
      }
    }
  },
  "resources": [
    {
      "type": "Microsoft.Authorization/policyAssignments",
      "apiVersion": "2023-04-01",
      "name": "[parameters('policyAssignmentName')]",
      "properties": {
        "policyDefinitionId": "[parameters('policyDefinitionID')]",
        "description": "Policy assignment to resource group scope created
with ARM template",
        "displayName": "[parameters('policyDisplayName')]",
        "nonComplianceMessages": [
          {
```

```
            "message": "Virtual machines should use managed disks"
          }
        ]
      }
    }
  ],
  "outputs": {
    "assignmentId": {
      "type": "string",
      "value": "[resourceId('Microsoft.Authorization/policyAssignments',
parameters('policyAssignmentName'))]"
    }
  }
}
```

The resource type defined in the ARM template is Microsoft.Authorization/policyAssignments.

The template uses three parameters to deploy the policy assignment:

- `policyAssignmentName` creates the policy assignment named *audit-vm-managed-disks*.
- `policyDefinitionID` uses the ID of the built-in policy definition. For reference, the commands to get the ID are in the section to deploy the template.
- `policyDisplayName` creates a display name that's visible in Azure portal.

For more information about ARM template files:

- To find more ARM template samples, go to Browse code samples.
- To learn more about template reference's for deployments, go to Azure template reference.
- To learn how to develop ARM templates, go to ARM template documentation.
- To learn about subscription-level deployments, go to Subscription deployments with ARM templates.

Deploy the ARM template

You can deploy the ARM template with Azure PowerShell or Azure CLI.

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

PowerShell

Azure PowerShell

```
Connect-AzAccount

# Run these commands if you have multiple subscriptions
Get-AzSubScription
Set-AzContext -Subscription <subscriptionID>
```

You can verify if `Microsoft.PolicyInsights` is registered. If it isn't, you can run a command to register the resource provider.

PowerShell

Azure PowerShell

```
Get-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights' |
   Select-Object -Property ResourceTypes, RegistrationState

Register-AzResourceProvider -ProviderNamespace
'Microsoft.PolicyInsights'
```

For more information, go to Get-AzResourceProvider and Register-AzResourceProvider.

The following commands display the `policyDefinitionID` parameter's value:

PowerShell

Azure PowerShell

```
(Get-AzPolicyDefinition |
  Where-Object { $_.Properties.DisplayName -eq 'Audit VMs that do not
use managed disks' }).ResourceId
```

The following commands deploy the policy definition to your resource group. Replace `<resourceGroupName>` with your resource group name:

PowerShell

Azure PowerShell

```
$rg = Get-AzResourceGroup -Name '<resourceGroupName>'
```

```
$deployparms = @{
Name = 'PolicyDeployment'
ResourceGroupName = $rg.ResourceGroupName
TemplateFile = 'policy-assignment.json'
}

New-AzResourceGroupDeployment @deployparms
```

The `$rg` variable stores properties for the resource group. The `$deployparms` variable uses splatting to create parameter values and improve readability. The `New-AzResourceGroupDeployment` command uses the parameter values defined in the `$deployparms` variable.

- `Name` is the deployment name displayed in the output and in Azure for the resource group's deployments.
- `ResourceGroupName` uses the `$rg.ResourceGroupName` property to get the name of your resource group where the policy is assigned.
- `TemplateFile` specifies the ARM template's name and location on your local computer.

You can verify the policy assignment's deployment with the following command:

PowerShell

The command uses the `$rg.ResourceId` property to get the resource group's ID.

Azure PowerShell

```
Get-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope
$rg.ResourceId
```

Output

```
Name               : audit-vm-managed-disks
ResourceId         :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks
ResourceName       : audit-vm-managed-disks
ResourceGroupName  : {resourceGroupName}
ResourceType       : Microsoft.Authorization/policyAssignments
SubscriptionId     : {subscriptionId}
PolicyAssignmentId :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks
Properties         :
```

```
Microsoft.Azure.Commands.ResourceManager.Cmdlets.Implementation.Policy.P
sPolicyAssignmentProperties
```

For more information, go to Get-AzPolicyAssignment.

Identify non-compliant resources

After the policy assignment is deployed, virtual machines that are deployed to the resource group are audited for compliance with the managed disk policy.

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

PowerShell

Azure PowerShell

```
$complianceparms = @{
ResourceGroupName = $rg.ResourceGroupName
PolicyAssignmentName = 'audit-vm-managed-disks'
Filter = 'IsCompliant eq false'
}

Get-AzPolicyState @complianceparms
```

The `$complianceparms` variable creates parameter values used in the `Get-AzPolicyState` command.

- `ResourceGroupName` gets the resource group name from the `$rg.ResourceGroupName` property.
- `PolicyAssignmentName` specifies the name used when the policy assignment was created.
- `Filter` uses an expression to find resources that aren't compliant with the policy assignment.

Your results resemble the following example and `ComplianceState` shows `NonCompliant`:

Output

```
Timestamp                : 2/26/2024 19:02:56
ResourceId               :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
```

```
ders/microsoft.compute/virtualmachines/{vmId}
PolicyAssignmentId       :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/microsoft.authorization/policyassignments/audit-vm-managed-disks
PolicyDefinitionId       :
/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-
923c-fb736d382a4d
IsCompliant              : False
SubscriptionId           : {subscriptionId}
ResourceType             : Microsoft.Compute/virtualMachines
ResourceLocation         : {location}
ResourceGroup            : {resourceGroupName}
ResourceTags             : tbd
PolicyAssignmentName     : audit-vm-managed-disks
PolicyAssignmentOwner    : tbd
PolicyAssignmentScope    :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}
PolicyDefinitionName     : 06a78e20-9358-41c9-923c-fb736d382a4d
PolicyDefinitionAction   : audit
PolicyDefinitionCategory : tbd
ManagementGroupIds       : {managementGroupId}
ComplianceState          : NonCompliant
AdditionalProperties     : {[complianceReasonCode, ]}
```

For more information, go to Get-AzPolicyState.

Clean up resources

PowerShell

Azure PowerShell

```powershell
Remove-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope
$rg.ResourceId
```

To sign out of your Azure PowerShell session:

Azure PowerShell

```powershell
Disconnect-AzAccount
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Create a policy assignment to identify non-compliant resources by using a Bicep file

Article • 03/04/2025

In this quickstart, you use a Bicep file to create a policy assignment that validates resource's compliance with an Azure policy. The policy is assigned to a resource group and audits virtual machines that don't use managed disks. After you create the policy assignment, you identify non-compliant virtual machines.

Bicep is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. It provides concise syntax, reliable type safety, and support for code reuse. Bicep offers the best authoring experience for your infrastructure-as-code solutions in Azure.

When assigning a built-in policy or initiative definition, it's optional to reference a version. Policy assignments of built-in definitions default to the latest version and automatically inherit minor version changes unless otherwise specified.

Prerequisites

- If you don't have an Azure account, create a free account before you begin.
- Bicep.
- Azure PowerShell or Azure CLI.
- Visual Studio Code and the Bicep extension for Visual Studio Code.
- `Microsoft.PolicyInsights` must be registered in your Azure subscription. To register a resource provider, you must have permission to register resource providers. That permission is included in the Contributor and Owner roles.
- A resource group with at least one virtual machine that doesn't use managed disks.

Review the Bicep file

The Bicep file creates a policy assignment for a resource group scope and assigns the built-in policy definition Audit VMs that do not use managed disks.

Create the following Bicep file as *policy-assignment.bicep*.

1. Open Visual Studio Code and select **File** > **New Text File**.
2. Copy and paste the Bicep file into Visual Studio Code.

3. Select **File** > **Save** and use the filename *policy-assignment.bicep*.

Bicep

```
param policyAssignmentName string = 'audit-vm-managed-disks'
param policyDefinitionID string =
'/providers/Microsoft.Authorization/policyDefinitions/06a78e20-9358-41c9-
923c-fb736d382a4d'
param policyDisplayName string = 'Audit VM managed disks'

resource assignment 'Microsoft.Authorization/policyAssignments@2023-04-01' =
{
  name: policyAssignmentName
  scope: resourceGroup()
  properties: {
    policyDefinitionId: policyDefinitionID
    description: 'Policy assignment to resource group scope created with
Bicep file'
    displayName: policyDisplayName
    nonComplianceMessages: [
      {
        message: 'Virtual machines should use managed disks'
      }
    ]
  }
}

output assignmentId string = assignment.id
```

The resource type defined in the Bicep file is Microsoft.Authorization/policyAssignments.

The Bicep file uses three parameters to deploy the policy assignment:

- `policyAssignmentName` creates the policy assignment named *audit-vm-managed-disks*.
- `policyDefinitionID` uses the ID of the built-in policy definition. For reference, the commands to get the ID are in the section to deploy the template.
- `policyDisplayName` creates a display name that's visible in Azure portal.

For more information about Bicep files:

- To find more Bicep samples, go to Browse code samples.
- To learn more about template reference's for deployments, go to Azure template reference.
- To learn how to develop Bicep files, go to Bicep documentation.
- To learn about subscription-level deployments, go to Subscription deployments with Bicep files.

Deploy the Bicep file

You can deploy the Bicep file with Azure PowerShell or Azure CLI.

From a Visual Studio Code terminal session, connect to Azure. If you have more than one subscription, run the commands to set context to your subscription. Replace `<subscriptionID>` with your Azure subscription ID.

PowerShell

Azure PowerShell

```
Connect-AzAccount

# Run these commands if you have multiple subscriptions
Get-AzSubScription
Set-AzContext -Subscription <subscriptionID>
```

You can verify if `Microsoft.PolicyInsights` is registered. If it isn't, you can run a command to register the resource provider.

PowerShell

Azure PowerShell

```
Get-AzResourceProvider -ProviderNamespace 'Microsoft.PolicyInsights' |
   Select-Object -Property ResourceTypes, RegistrationState

Register-AzResourceProvider -ProviderNamespace
'Microsoft.PolicyInsights'
```

For more information, go to Get-AzResourceProvider and Register-AzResourceProvider.

The following commands display the `policyDefinitionID` parameter's value:

PowerShell

Azure PowerShell

```
(Get-AzPolicyDefinition |
  Where-Object { $_.Properties.DisplayName -eq 'Audit VMs that do not
```

```
use managed disks' }).ResourceId
```

The following commands deploy the policy definition to your resource group. Replace `<resourceGroupName>` with your resource group name:

PowerShell

Azure PowerShell

```
$rg = Get-AzResourceGroup -Name '<resourceGroupName>'

$deployparms = @{
Name = 'PolicyDeployment'
ResourceGroupName = $rg.ResourceGroupName
TemplateFile = 'policy-assignment.bicep'
}

New-AzResourceGroupDeployment @deployparms
```

The `$rg` variable stores properties for the resource group. The `$deployparms` variable uses splatting to create parameter values and improve readability. The `New-AzResourceGroupDeployment` command uses the parameter values defined in the `$deployparms` variable.

- `Name` is the deployment name displayed in the output and in Azure for the resource group's deployments.
- `ResourceGroupName` uses the `$rg.ResourceGroupName` property to get the name of your resource group where the policy is assigned.
- `TemplateFile` specifies the Bicep file's name and location on your local computer.

You can verify the policy assignment's deployment with the following command:

PowerShell

The command uses the `$rg.ResourceId` property to get the resource group's ID.

Azure PowerShell

```
Get-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope
$rg.ResourceId
```

Output

```
Name               : audit-vm-managed-disks
ResourceId         :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks
ResourceName       : audit-vm-managed-disks
ResourceGroupName  : {resourceGroupName}
ResourceType       : Microsoft.Authorization/policyAssignments
SubscriptionId     : {subscriptionId}
PolicyAssignmentId :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/Microsoft.Authorization/policyAssignments/audit-vm-managed-disks
Properties         :
Microsoft.Azure.Commands.ResourceManager.Cmdlets.Implementation.Policy.P
sPolicyAssignmentProperties
```

For more information, go to Get-AzPolicyAssignment.

Identify non-compliant resources

After the policy assignment is deployed, virtual machines that are deployed to the resource group are audited for compliance with the managed disk policy.

The compliance state for a new policy assignment takes a few minutes to become active and provide results about the policy's state.

PowerShell

Azure PowerShell

```
$complianceparms = @{
ResourceGroupName = $rg.ResourceGroupName
PolicyAssignmentName = 'audit-vm-managed-disks'
Filter = 'IsCompliant eq false'
}

Get-AzPolicyState @complianceparms
```

The `$complianceparms` variable creates parameter values used in the `Get-AzPolicyState` command.

- `ResourceGroupName` gets the resource group name from the `$rg.ResourceGroupName` property.

- `PolicyAssignmentName` specifies the name used when the policy assignment was created.
- `Filter` uses an expression to find resources that aren't compliant with the policy assignment.

Your results resemble the following example and `ComplianceState` shows `NonCompliant`:

Output

```
Timestamp                : 2/20/2024 18:55:45
ResourceId               :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/microsoft.compute/virtualmachines/{vmId}
PolicyAssignmentId       :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/provi
ders/microsoft.authorization/policyassignments/audit-vm-managed-disks
PolicyDefinitionId       :
/providers/microsoft.authorization/policydefinitions/06a78e20-9358-41c9-
923c-fb736d382a4d
IsCompliant              : False
SubscriptionId           : {subscriptionId}
ResourceType             : Microsoft.Compute/virtualMachines
ResourceLocation         : {location}
ResourceGroup            : {resourceGroupName}
ResourceTags             : tbd
PolicyAssignmentName     : audit-vm-managed-disks
PolicyAssignmentOwner    : tbd
PolicyAssignmentScope    :
/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}
PolicyDefinitionName     : 06a78e20-9358-41c9-923c-fb736d382a4d
PolicyDefinitionAction   : audit
PolicyDefinitionCategory : tbd
ManagementGroupIds       : {managementGroupId}
ComplianceState          : NonCompliant
AdditionalProperties     : {[complianceReasonCode, ]}
```

For more information, go to Get-AzPolicyState.

Clean up resources

PowerShell

Azure PowerShell

```
Remove-AzPolicyAssignment -Name 'audit-vm-managed-disks' -Scope
```

```
$rg.ResourceId
```

To sign out of your Azure PowerShell session:

Azure PowerShell

```
Disconnect-AzAccount
```

Next steps

In this quickstart, you assigned a policy definition to identify non-compliant resources in your Azure environment.

To learn more about how to assign policies that validate resource compliance, continue to the tutorial.

Tutorial: Create and manage policies to enforce compliance

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What is Azure Bastion?

Summarize this article for me

Azure Bastion is a fully managed PaaS service that provides secure and seamless RDP/SSH connectivity to your virtual machines directly over TLS from the Azure portal, or via the native SSH or RDP client already installed on your local computer. Azure Bastion is deployed directly in your virtual network and supports all VMs in the virtual network using private IP addresses. When you connect via Azure Bastion, your virtual machines don't need a public IP address, agent, or special client software.

Azure Bastion is available in four SKUs: Developer, Basic, Standard, and Premium.

> **Note**
>
> Azure Bastion is one of the services that make up the Network Security category in Azure. Other services in this category include Azure DDoS Protection, Azure Firewall, and Azure Web Application Firewall. Each service has its own unique features and use cases. For more information on this service category, see Network Security.

Key benefits

Azure Bastion provides the following benefits:

- **Secure connectivity over TLS**: Connect to VMs using RDP/SSH over TLS on port 443. Learn more about connection methods and Kerberos authentication.
- **Protection from external threats**: Your VMs are protected from port scanning. Deploy with availability zones for additional resilience.
- **Scalability and flexibility**: Configure host scaling, use shareable links, and connect via IP address.
- **Reduced management overhead**: Deploy once and use virtual network peering to serve multiple networks.
- **Compliance and audit**: Use session recording for compliance requirements (Premium SKU).

SKUs

Azure Bastion offers four SKU tiers:

- **Premium**: Includes all Standard features plus session recording for compliance and private-only deployment.

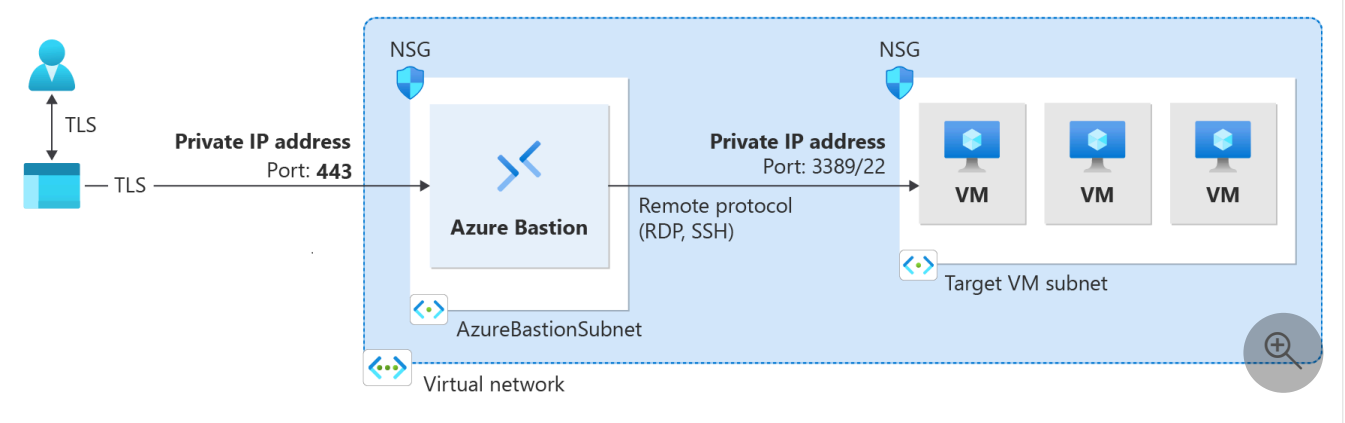
- **Standard**: Includes all Basic features plus scalability and advanced features (native client, shareable links, IP-based connections, custom ports, file transfer).
- **Basic**: Dedicated deployment with fixed capacity for production environments with moderate connection requirements.
- **Developer**: Free tier using shared infrastructure recommended for development and testing. Supports one VM at a time. Available in select regions.

For a complete feature comparison and capacity details, see Choose the right Azure Bastion SKU.

Architecture

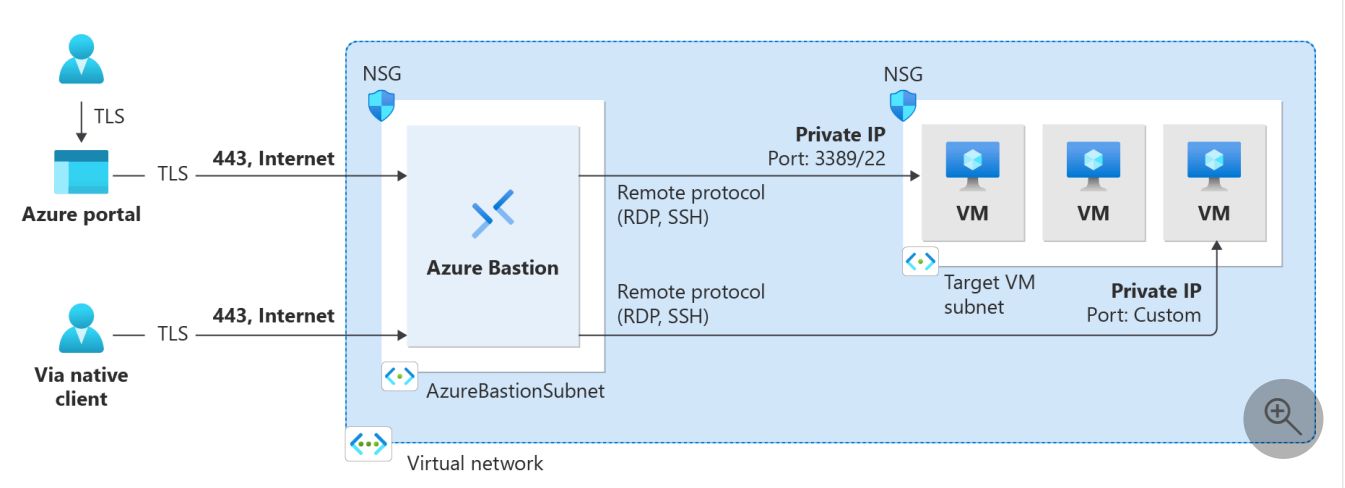
Azure Bastion offers three deployment architectures:

Private-only deployment: Premium SKU without public IP address for enhanced security.

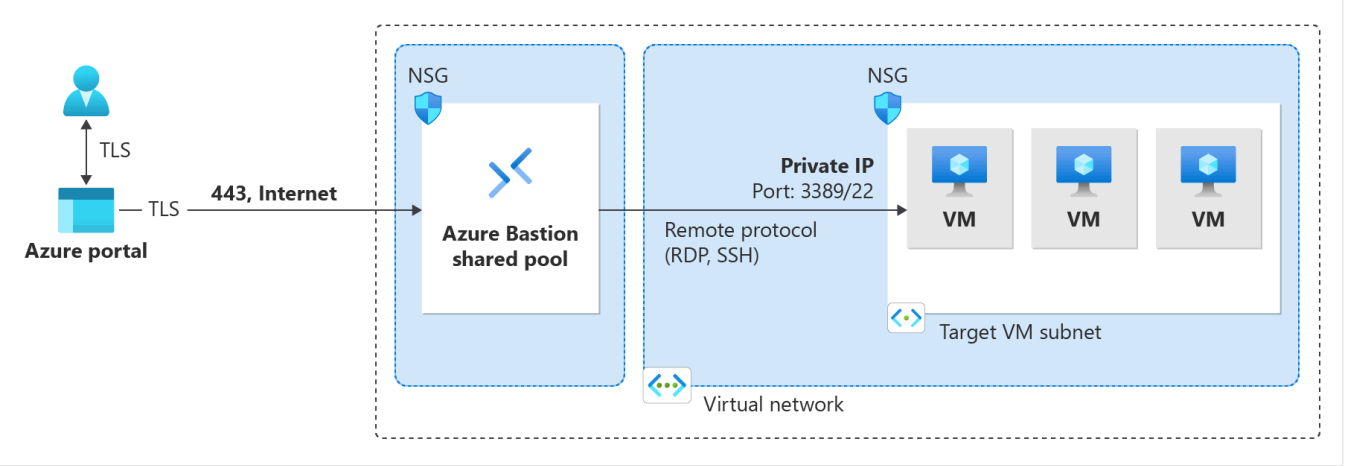


For detailed information about each architecture, deployment requirements, and network topology options, see Bastion design and architecture.

Dedicated deployment: Basic, Standard, and Premium SKUs deployed to your virtual network.



Developer: Shared infrastructure for development and testing environments.



Requirements

Deployment requirements vary by SKU. Developer uses shared infrastructure with no virtual network required. Basic, Standard, and Premium require a dedicated subnet (AzureBastionSubnet) and public IP address. Premium supports private-only deployment without a public IP.

For complete requirements including subnet sizing and NSG rules, see About Bastion configuration settings.

Connection methods

Azure Bastion supports multiple connection methods:

- **Browser-based connections**: Connect through the Azure portal using an HTML5 web client. Available for all SKU tiers. No additional client software required.
- **Native client connections**: Connect using the SSH or RDP client already installed on your local computer. Available for Standard and Premium SKUs. Supports Microsoft Entra ID authentication and file transfer.
- **Shareable links**: Create shareable links that allow users to connect to VMs without accessing the Azure portal. Available for Standard and Premium SKUs.

For more information about connection methods and authentication options, see About VM connections and features.

Pricing and SLA

Azure Bastion pricing combines hourly SKU charges with outbound data transfer costs. Billing starts from the moment Bastion is deployed, regardless of usage.

For pricing details, see Azure Bastion pricing. For SLA and reliability information, see Reliability in Azure Bastion.

What's new

Azure Bastion is continuously updated with new features and improvements. To learn about the latest updates and announcements, see What's new in Azure Bastion?

Troubleshooting and FAQ

For information about troubleshooting and frequently asked questions, see the troubleshooting guide and Azure Bastion FAQ.

Next steps

- Quickstart: Deploy Bastion automatically with default settings and Standard SKU
- Quickstart: Deploy Azure Bastion from the Azure portal
- Choose the right Azure Bastion SKU
- About Bastion configuration settings
- Azure Bastion FAQ
- Learn module: Introduction to Azure Bastion

Last updated on 02/04/2026

Quickstart: Deploy Azure Bastion from the Azure portal

In this quickstart, you learn how to deploy Azure Bastion to your virtual network from the Azure portal. You can deploy Bastion with default settings for a quick setup, configure custom settings to specify the SKU and scaling options, or use the free Developer SKU for basic connectivity. After you deploy Bastion, you can use SSH or RDP to connect to virtual machines (VMs) in the virtual network via Bastion by using the private IP addresses of the VMs. The VMs that you connect to don't need a public IP address, client software, an agent, or a special configuration. For more information about Bastion, see What is Azure Bastion?

The steps in this article help you:

- Deploy Bastion to your virtual network by using the Azure portal.
- Connect to your VM via the portal by using SSH or RDP connectivity and the VM's private IP address.
- Remove your VM's public IP address if you don't need it for anything else.

> **Important**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

Prerequisites

To complete this quickstart, you need these resources:

- An Azure subscription. If you don't already have one, you can activate your MSDN subscriber benefits or sign up for a free account.
- A virtual network to which you'll deploy Bastion.
- A virtual machine in the virtual network. This VM isn't part of the Bastion configuration and doesn't become a bastion host. You connect to this VM later in the exercise. If you don't have a VM, create one by using Quickstart: Create a Windows VM or Quickstart: Create a Linux VM.
- Required VM roles:
  - Reader role on the virtual machine

    - Reader role on the network adapter (NIC) with the private IP of the virtual machine

- Required VM inbound ports:
    - For Windows VMs: RDP (3389)
    - For Linux VMs: SSH (22)

- **For Developer SKU only**: The VM must be in a region that supports Bastion Developer.

> **Note**
>
> The use of Azure Bastion with Azure Private DNS zones is supported. However, there are restrictions. For more information, see the Azure Bastion FAQ.

Deploy Bastion

To deploy Bastion, sign in to the Azure portal and go to your VM or virtual network.

Select the tab for the deployment method you want to use:

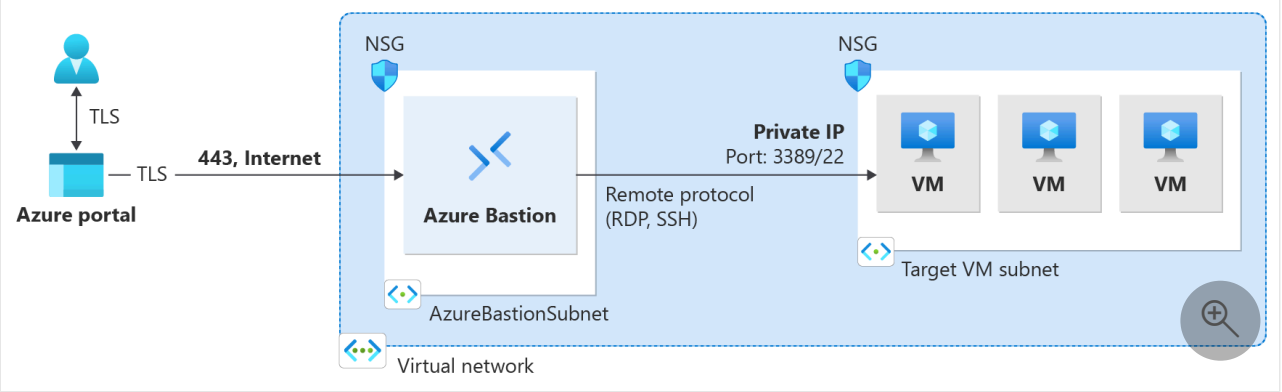
- **Default settings**: Quick one-click deployment with Standard SKU.
- **Custom settings**: Full control over SKU, scaling, availability zones, and other features.
- **Developer SKU (free)**: No-cost option with basic features for dev/test. Uses shared pool architecture. Limited to select regions.

> **Note**
>
> Dedicated deployments (Default and Custom settings) take approximately 10 minutes to complete. Developer SKU deploys in seconds.

Default settings

When you deploy Bastion using the **Deploy Bastion** option, Bastion deploys automatically with the Standard SKU and default settings based on your virtual network. You can configure additional settings or upgrade the SKU after deployment completes.

The following diagram shows the dedicated deployment architecture used by the Default settings options.



Default values:

To deploy Bastion with default settings:

1. Go to your virtual network (or VM). In the left pane, select **Connect** > **Bastion**.
2. In the **Bastion** pane, select **Deploy Bastion**.
3. Bastion deploys automatically with default settings. The deployment process takes about 10 minutes to complete.

Remove VM public IP address

When you connect to a VM by using Azure Bastion, you don't need a public IP address for your VM. If you aren't using the public IP address for anything else, you can dissociate it from your VM:

1. Go to your virtual machine. On the **Overview** page, click the **Public IP address** to open the Public IP address page.

2. On the **Public IP address** page, go to **Overview**. You can view the resource that this IP address is **Associated to**. Select **Dissociate** at the top of the pane.

3. Select **Yes** to dissociate the IP address from the VM network interface. After you dissociate the public IP address from the network interface, verify that it's no longer listed under **Associated to**.

4. After you dissociate the IP address, you can delete the public IP address resource. On the **Public IP address** pane for the VM, at the top of the **Overview** page, select **Delete**.

5. Select **Yes** to delete the public IP address.

Clean up resources

When you finish using the virtual network and the virtual machines, delete the resource group and all of the resources that it contains:

1. Enter the name of your resource group in the **Search** box at the top of the portal, and then select it from the search results.
2. Select **Delete resource group**.
3. Enter your resource group for **TYPE THE RESOURCE GROUP NAME**, and then select **Delete**.

Next steps

In this quickstart, you deployed Bastion to your virtual network. Next, you can connect to a virtual machine securely via Bastion, configure more features, and work with VM connections.

- Connect to a Windows VM: RDP | SSH
- Connect to a Linux VM: SSH | RDP
- Connect from a native client: Windows | Linux/SSH
- Connect to a scale set
- VM connections and features
- Azure Bastion configuration settings
- Bastion SKU comparison
- IP-based connections

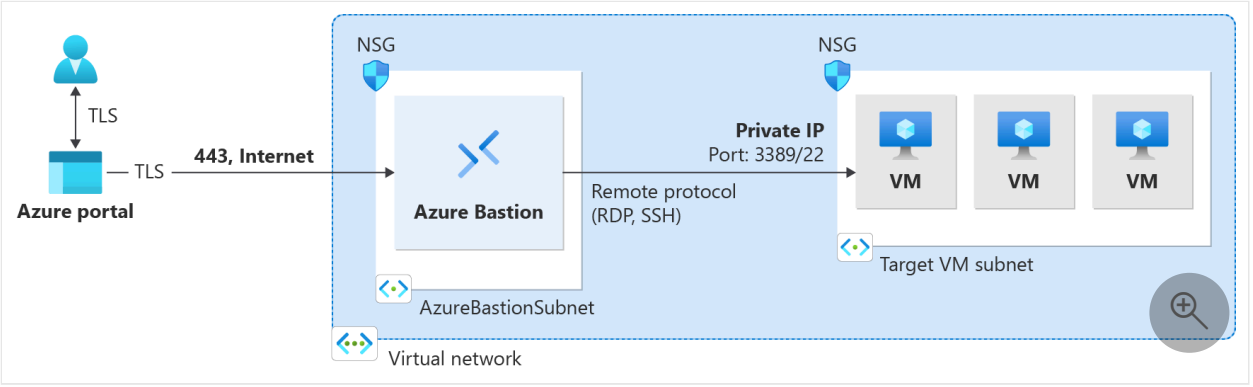
Last updated on 03/17/2026

Deploy Bastion using Azure PowerShell

Article • 03/03/2025

This article shows you how to deploy Azure Bastion using PowerShell. Azure Bastion is a PaaS service that's maintained for you, not a bastion host that you install on your VM and maintain yourself. An Azure Bastion deployment is per virtual network, not per subscription/account or virtual machine. For more information about Azure Bastion, see What is Azure Bastion?

Once you deploy Bastion to your virtual network, you can connect to your VMs via private IP address. This seamless RDP/SSH experience is available to all the VMs in the same virtual network. If your VM has a public IP address that you don't need for anything else, you can remove it.



In this article, you create a virtual network (if you don't already have one), deploy Azure Bastion using PowerShell, and connect to a VM. The examples show Bastion deployed using the Standard SKU tier, but you can use a different Bastion SKU, depending on the features you'd like to use. For more information, see Bastion SKUs.

You can also deploy Bastion by using the following other methods:

- Azure portal - Deploy Bastion with specified settings and SKUs
- Azure portal - Deploy Bastion with default settings and Standard SKU
- Deploy using Azure CLI

> **Note**
>
> The use of Azure Bastion with Azure Private DNS zones is supported. However, there are restrictions. For more information, see the Azure Bastion FAQ.

Before beginning

Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your MSDN subscriber benefits or sign up for a free account.

PowerShell

This article uses PowerShell cmdlets. To run the cmdlets, you can use Azure Cloud Shell. Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open Cloud Shell, just select **Open Cloudshell** from the upper-right corner of a code block. You can also open Cloud Shell on a separate browser tab by going to https://shell.azure.com/powershell. Select **Copy** to copy the blocks of code, paste them into Cloud Shell, and select the Enter key to run them.

You can also install and run the Azure PowerShell cmdlets locally on your computer. PowerShell cmdlets are updated frequently. If you haven't installed the latest version, the values specified in the instructions may fail. To find the versions of Azure PowerShell installed on your computer, use the `Get-Module -ListAvailable Az` cmdlet. To install or update, see Install the Azure PowerShell module.

Example values

You can use the following example values when creating this configuration, or you can substitute your own.

Example VNet and VM values:

Expand table

Name	Value
Virtual machine	TestVM
Resource group	TestRG1
Region	East US
Virtual network	VNet1
Address space	10.1.0.0/16
Subnets	FrontEnd: 10.1.0.0/24

Azure Bastion values:

Expand table

Name	Value
Name	VNet1-bastion
Subnet Name	FrontEnd
Subnet Name	AzureBastionSubnet
AzureBastionSubnet addresses	A subnet within your virtual network address space with a subnet mask /26 or larger. For example, 10.1.1.0/26.
Tier/SKU	Standard
Public IP address	Create new
Public IP address name	VNet1-ip
Public IP address SKU	Standard
Assignment	Static

Deploy Bastion

This section helps you create a virtual network, subnets, and deploy Azure Bastion using Azure PowerShell.

> **Important**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see Pricing and SKUs. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

1. Create a resource group, a virtual network, and a front end subnet to which you deploy the VMs that you'll connect to via Bastion. If you're running PowerShell locally, open your PowerShell console with elevated privileges and connect to Azure using the `Connect-AzAccount` command.

   Azure PowerShell

   ```
   New-AzResourceGroup -Name TestRG1 -Location EastUS `
   $frontendSubnet = New-AzVirtualNetworkSubnetConfig -Name FrontEnd `
   -AddressPrefix "10.1.0.0/24" `
   ```

```
$virtualNetwork = New-AzVirtualNetwork `
-Name TestVNet1 -ResourceGroupName TestRG1 `
-Location EastUS -AddressPrefix "10.1.0.0/16" `
-Subnet $frontendSubnet `
$virtualNetwork | Set-AzVirtualNetwork
```

2. Configure and set the Azure Bastion subnet for your virtual network. This subnet is reserved exclusively for Azure Bastion resources. You must create this subnet using the name value **AzureBastionSubnet**. This value lets Azure know which subnet to deploy the Bastion resources to. The example in the following section helps you add an Azure Bastion subnet to an existing VNet.

   - The smallest subnet AzureBastionSubnet size you can create is /26. We recommend that you create a /26 or larger size to accommodate host scaling.
     - For more information about scaling, see Configuration settings - Host scaling.
     - For more information about settings, see Configuration settings - AzureBastionSubnet.
   - Create the **AzureBastionSubnet** without any route tables or delegations.
   - If you use Network Security Groups on the **AzureBastionSubnet**, refer to the Work with NSGs article.

   Set the variable.

   Azure PowerShell

   ```
   $vnet = Get-AzVirtualNetwork -Name "TestVNet1" -ResourceGroupName
   "TestRG1"
   ```

   Add the subnet.

   Azure PowerShell

   ```
   Add-AzVirtualNetworkSubnetConfig `
   -Name "AzureBastionSubnet" -VirtualNetwork $vnet `
   -AddressPrefix "10.1.1.0/26" | Set-AzVirtualNetwork
   ```

3. Create a public IP address for Azure Bastion. The public IP is the public IP address of the Bastion resource on which RDP/SSH will be accessed (over port 443). The public IP address must be in the same region as the Bastion resource you're creating.

   Azure PowerShell

```
$publicip = New-AzPublicIpAddress -ResourceGroupName "TestRG1" `
-name "VNet1-ip" -location "EastUS" `
-AllocationMethod Static -Sku Standard
```

4. Create a new Azure Bastion resource in the AzureBastionSubnet using the New-AzBastion command. The following example uses the **Basic SKU**. However, you can also deploy Bastion using a different SKU by changing the -Sku value. The SKU you select determines the Bastion features and connect to VMs using more connection types. For more information, see Bastion SKUs.

   Azure PowerShell

   ```
   New-AzBastion -ResourceGroupName "TestRG1" -Name "VNet1-bastion" `
   -PublicIpAddressRgName "TestRG1" -PublicIpAddressName "VNet1-ip" `
   -VirtualNetworkRgName "TestRG1" -VirtualNetworkName "TestVNet1" `
   -Sku "Basic"
   ```

5. It takes about 10 minutes for the Bastion resources to deploy. You can create a VM in the next section while Bastion deploys to your virtual network.

Create a VM

You can create a VM using the Quickstart: Create a VM using PowerShell or Quickstart: Create a VM using the portal articles. Be sure you deploy the VM to the same virtual network to which you deployed Bastion. The VM you create in this section isn't a part of the Bastion configuration and doesn't become a bastion host. You connect to this VM later in this tutorial via Bastion.

The following required roles for your resources.

- Required VM roles:
  - Reader role on the virtual machine.
  - Reader role on the NIC with private IP of the virtual machine.

- Required inbound ports:
  - For Windows VMS - RDP (3389)
  - For Linux VMs - SSH (22)

Connect to a VM

You can use the Connection steps in the following section to connect to your VM. You can also use any of the following articles to connect to a VM. Some connection types

require the Bastion Standard SKU.

- Connect to a Windows VM
  - RDP
  - SSH
- Connect to a Linux VM
  - SSH
- Connect to a scale set
- Connect via IP address
- Connect from a native client
  - Windows client
  - Linux/SSH client

Connection steps

1. In the Azure portal, go to the virtual machine that you want to connect to.

2. At the top of the pane, select **Connect** > **Bastion** to go to the **Bastion** pane. You can also go to the **Bastion** pane by using the left menu.

3. The options available on the **Bastion** pane depend on the Bastion SKU.

   If you're using the **Basic SKU**, you connect to a Windows computer by using RDP and port 3389. Also for the Basic SKU, you connect to a Linux computer by using SSH and port 22. You don't have options to change the port number or the protocol. However, you can change the keyboard language for RDP by expanding **Connection Settings** on this pane.

   If you're using the **Standard SKU**, you have more connection protocol and port options available. Expand **Connection Settings** to see the options. Typically, unless you configure different settings for your VM, you connect to a Windows computer by using RDP and port 3389. You connect to a Linux computer by using SSH and port 22.

4. For **Authentication Type**, select the authentication type from the dropdown list. The protocol determines the available authentication types. Complete the required authentication values.

5. To open the VM session in a new browser tab, leave **Open in new browser tab** selected.

6. Select **Connect** to connect to the VM.

7. Confirm that the connection to the virtual machine opens directly in the Azure portal (over HTML5) by using port 443 and the Bastion service.

Using keyboard shortcut keys while you're connected to a VM might not result in the same behavior as shortcut keys on a local computer. For example, when you're connected to a Windows VM from a Windows client, Ctrl+Alt+End is the keyboard shortcut for Ctrl+Alt+Delete on a local computer. To do this from a Mac while you're connected to a Windows VM, the keyboard shortcut is fn+control+option+delete.

To enable audio output

You can enable remote audio output for your VM. Some VMs automatically enable this setting, whereas others require you to enable audio settings manually. The settings are changed on the VM itself. Your Bastion deployment doesn't need any special configuration settings to enable remote audio output. Audio input is not supported at the moment.

> ⓘ **Note**
>
> Audio output uses bandwidth on your internet connection.

To enable remote audio output on a Windows VM:

1. After you're connected to the VM, an audio button appears on the lower-right corner of the toolbar. Right-click the audio button, and then select **Sounds**.
2. A pop-up message asks if you want to enable the Windows Audio Service. Select **Yes**. You can configure more audio options in **Sound preferences**.
3. To verify sound output, hover over the audio button on the toolbar.

Remove VM public IP address

Azure Bastion doesn't use the public IP address to connect to the client VM. If you don't need the public IP address for your VM, you can disassociate the public IP address. See Dissociate a public IP address from an Azure VM.

Next steps

- To use Network Security Groups with the Azure Bastion subnet, see Work with NSGs.
- To understand VNet peering, see Virtual Network peering and Azure Bastion.

Feedback

Was this page helpful? Yes No

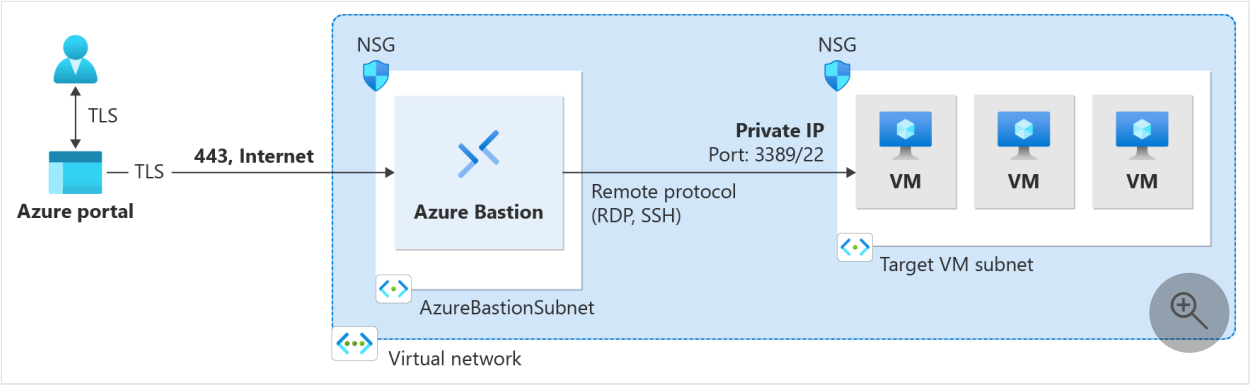
Provide product feedback | Get help at Microsoft Q&A

Deploy Bastion using Azure CLI

Article • 04/05/2024

This article shows you how to deploy Azure Bastion using CLI. Azure Bastion is a PaaS service that's maintained for you, not a bastion host that you install on your VM and maintain yourself. An Azure Bastion deployment is per virtual network, not per subscription/account or virtual machine. For more information about Azure Bastion, see What is Azure Bastion?

Once you deploy Bastion to your virtual network, you can connect to your VMs via private IP address. This seamless RDP/SSH experience is available to all the VMs in the same virtual network. If your VM has a public IP address that you don't need for anything else, you can remove it.



In this article, you create a virtual network (if you don't already have one), deploy Azure Bastion using CLI, and connect to a VM. You can also deploy Bastion by using the following other methods:

- Azure portal
- Azure PowerShell
- Quickstart - deploy with default settings

> **Note**
>
> The use of Azure Bastion with Azure Private DNS zones is supported. However, there are restrictions. For more information, see the Azure Bastion FAQ.

Before beginning

Azure subscription

Verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your MSDN subscriber benefits or sign up for a free account.

Azure CLI

This article uses the Azure CLI. To run commands, you can use Azure Cloud Shell. The Azure Cloud Shell is a free interactive shell that you can use to run the steps in this article. It has common Azure tools preinstalled and configured to use with your account.

To open the Cloud Shell, just select **Try it** from the upper right corner of a code block. You can also launch Cloud Shell in a separate browser tab by going to https://shell.azure.com and toggle the dropdown in the left corner to reflect Bash or PowerShell. Select **Copy** to copy the blocks of code, paste it into the Cloud Shell, and press enter to run it.

Deploy Bastion

This section helps you deploy Azure Bastion using Azure CLI.

> **Important**
>
> Hourly pricing starts from the moment that Bastion is deployed, regardless of outbound data usage. For more information, see **Pricing** and **SKUs**. If you're deploying Bastion as part of a tutorial or test, we recommend that you delete this resource after you finish using it.

1. If you don't already have a virtual network, create a resource group and a virtual network using az group create and az network vnet create.

   Azure CLI

   ```
   az group create --name TestRG1 --location eastus
   ```

   Azure CLI

   ```
   az network vnet create --resource-group TestRG1 --name VNet1 --address-prefix 10.1.0.0/16 --subnet-name default --subnet-prefix 10.1.0.0/24
   ```

2. Use az network vnet subnet create to create the subnet to which Bastion will be deployed. The subnet you create must be named **AzureBastionSubnet**. This subnet

is reserve exclusively for Azure Bastion resources. If you don't have a subnet with the naming value **AzureBastionSubnet**, Bastion won't deploy.

- The smallest subnet AzureBastionSubnet size you can create is /26. We recommend that you create a /26 or larger size to accommodate host scaling.
  - For more information about scaling, see Configuration settings - Host scaling.
  - For more information about settings, see Configuration settings - AzureBastionSubnet.
- Create the **AzureBastionSubnet** without any route tables or delegations.
- If you use Network Security Groups on the **AzureBastionSubnet**, refer to the Work with NSGs article.

Azure CLI

```
az network vnet subnet create --name AzureBastionSubnet --resource-
group TestRG1 --vnet-name VNet1 --address-prefix 10.1.1.0/26
```

3. Create a public IP address for Azure Bastion. The public IP is the public IP address the Bastion resource on which RDP/SSH will be accessed (over port 443). The public IP address must be in the same region as the Bastion resource you're creating. For this reason, pay particular attention to the `--location` value that you specify.

Azure CLI

```
az network public-ip create --resource-group TestRG1 --name VNet1-ip --
sku Standard --location eastus
```

4. Use az network bastion create to create a new Azure Bastion resource for your virtual network. It takes about 10 minutes for the Bastion resource to create and deploy.

   The following example deploys Bastion using the **Basic** SKU tier. You can also deploy using other SKUs. The SKU determines the features that your Bastion deployment supports. If you don't specify a SKU in your command, the SKU defaults to Standard. For more information, see Bastion SKUs.

Azure CLI

```
az network bastion create --name VNet1-bastion --public-ip-address
VNet1-ip --resource-group TestRG1 --vnet-name VNet1 --location eastus -
-sku Basic
```

Connect to a VM

If you don't already have VMs in your virtual network, you can create a VM using Quickstart: Create a Windows VM, or Quickstart: Create a Linux VM

You can use any of the following articles, or the steps in the following section, to help you connect to a VM. Some connection types require the Bastion Standard SKU or higher.

- Connect to a Windows VM
  - RDP
  - SSH
- Connect to a Linux VM
  - SSH
- Connect to a scale set
- Connect via IP address
- Connect from a native client
  - Windows client
  - Linux/SSH client

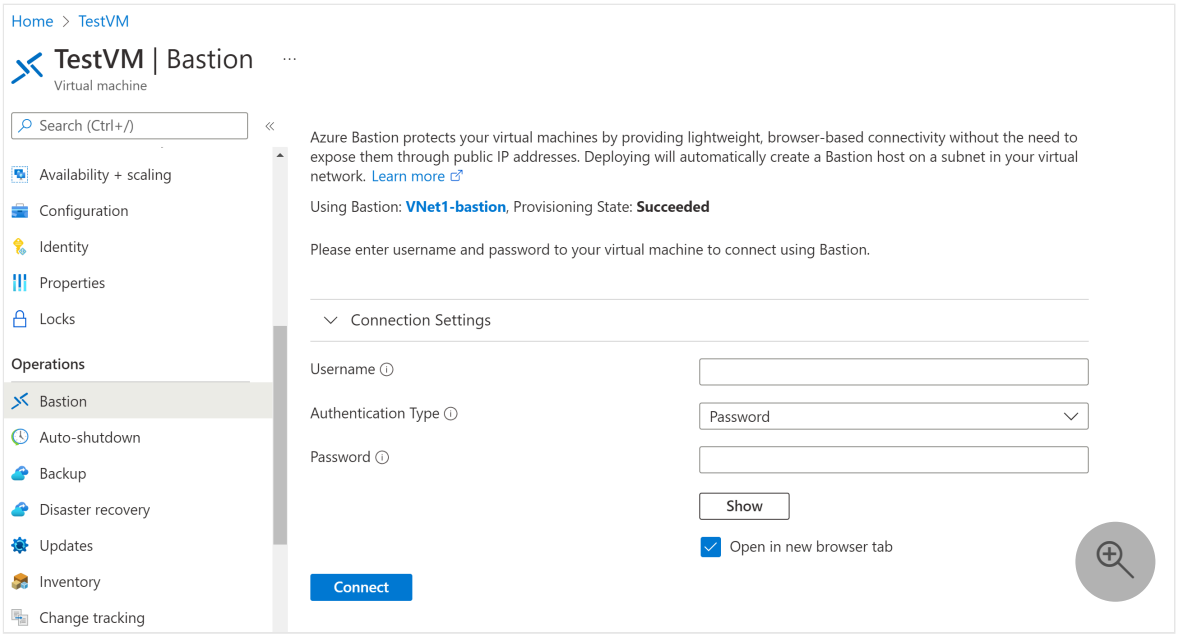
Connect using the portal

The following steps walk you through one type of connection using the Azure portal.

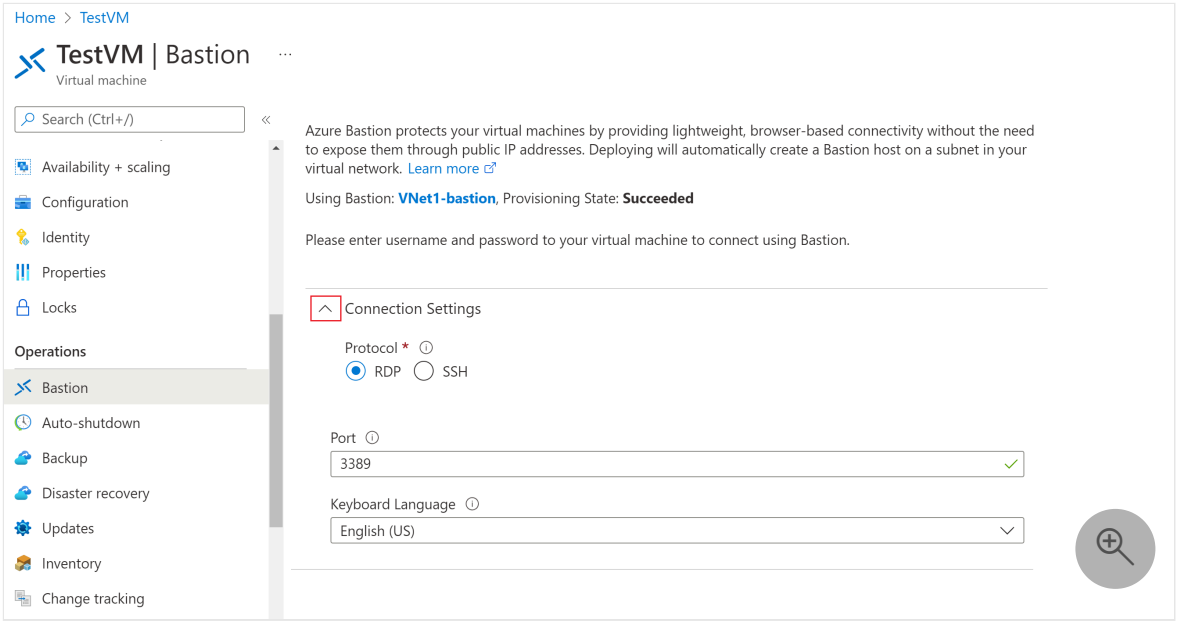
1. In the Azure portal, go to the virtual machine that you want to connect to.

2. At the top of the pane, select **Connect** > **Bastion** to go to the **Bastion** pane. You can also go to the **Bastion** pane by using the left menu.

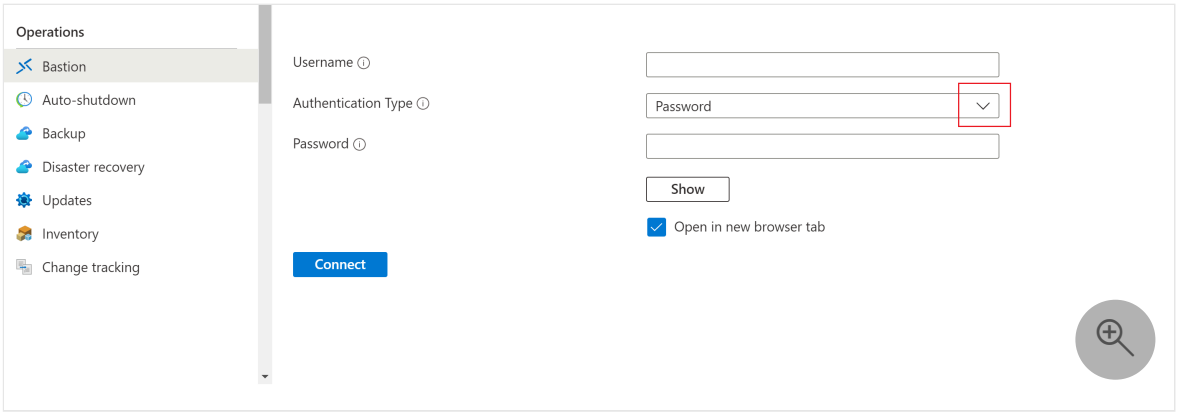
3. The options available on the **Bastion** pane depend on the Bastion SKU. If you're using the Basic SKU, you connect to a Windows computer by using RDP and port 3389. Also for the Basic SKU, you connect to a Linux computer by using SSH and port 22. You don't have options to change the port number or the protocol. However, you can change the keyboard language for RDP by expanding **Connection Settings**.



If you're using the Standard SKU, you have more connection protocol and port options available. Expand **Connection Settings** to see the options. Typically, unless you configure different settings for your VM, you connect to a Windows computer by using RDP and port 3389. You connect to a Linux computer by using SSH and port 22.



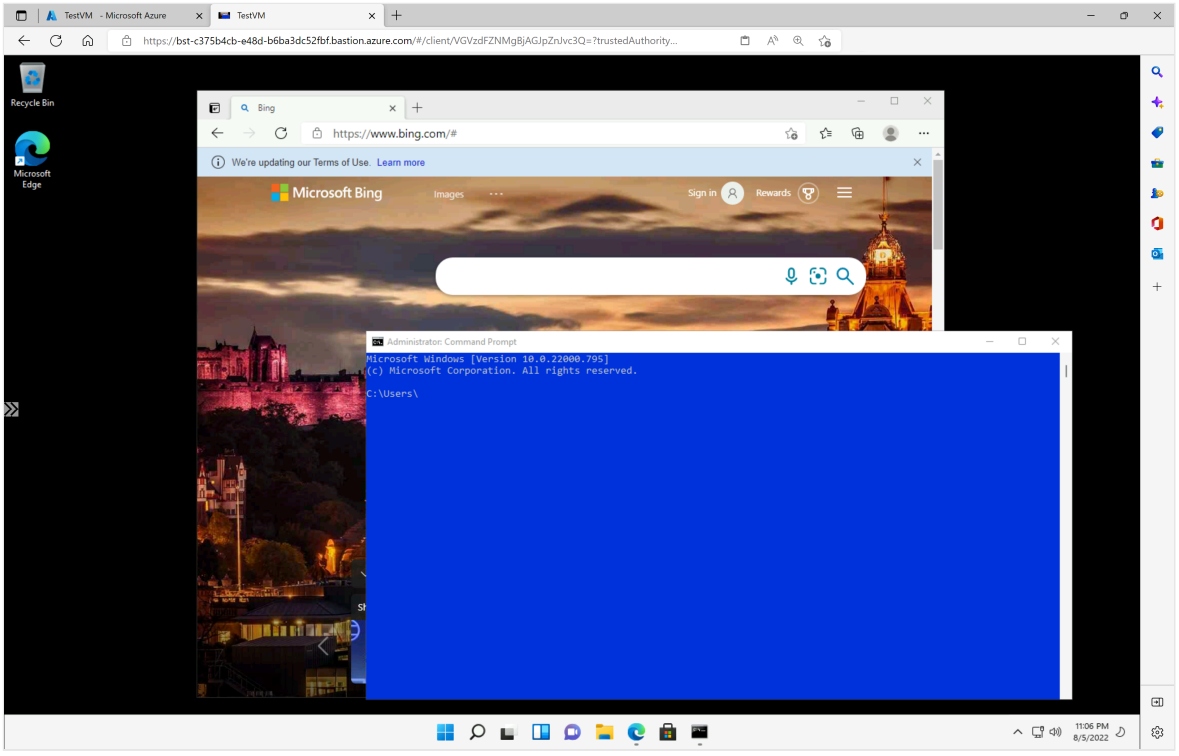
4. For **Authentication Type**, select from the dropdown list. The protocol determines the available authentication types. Complete the required authentication values.



5. To open the VM session in a new browser tab, leave **Open in new browser tab** selected.

6. Select **Connect** to connect to the VM.

7. Confirm that the connection to the virtual machine opens directly in the Azure portal (over HTML5) by using port 443 and the Bastion service.



> **Note**
>
> When you connect, the desktop of the VM will look different from the example screenshot.

Using keyboard shortcut keys while you're connected to a VM might not result in the same behavior as shortcut keys on a local computer. For example, when you're connected to a Windows VM from a Windows client, Ctrl+Alt+End is the keyboard

shortcut for Ctrl+Alt+Delete on a local computer. To do this from a Mac while you're connected to a Windows VM, the keyboard shortcut is Fn+Ctrl+Alt+Backspace.

To enable audio output

You can enable remote audio output for your VM. Some VMs automatically enable this setting, whereas others require you to enable audio settings manually. The settings are changed on the VM itself. Your Bastion deployment doesn't need any special configuration settings to enable remote audio output.

> **Note**
>
> Audio output uses bandwidth on your internet connection.

To enable remote audio output on a Windows VM:

1. After you're connected to the VM, an audio button appears on the lower-right corner of the toolbar. Right-click the audio button, and then select **Sounds**.
2. A pop-up message asks if you want to enable the Windows Audio Service. Select **Yes**. You can configure more audio options in **Sound preferences**.
3. To verify sound output, hover over the audio button on the toolbar.

Remove VM public IP address

Azure Bastion doesn't use the public IP address to connect to the client VM. If you don't need the public IP address for your VM, you can disassociate the public IP address. See Dissociate a public IP address from an Azure VM.

Next steps

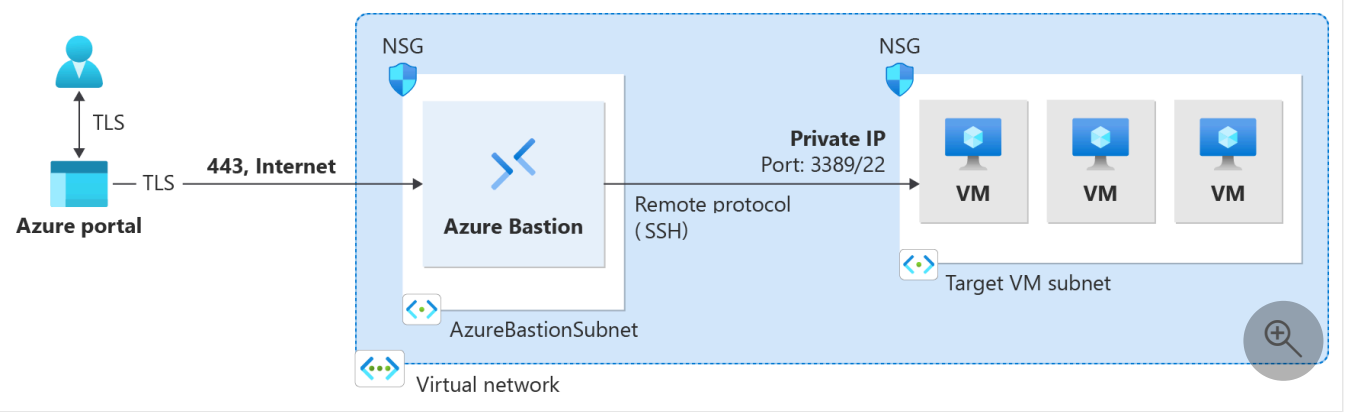
- To use Network Security Groups with the Azure Bastion subnet, see Work with NSGs.
- To understand VNet peering, see VNet peering and Azure Bastion.

Create an SSH connection to a Linux VM using Azure Bastion

This article describes how to create a secure SSH connection to your Linux virtual machines using Azure Bastion. You can connect through the Azure portal (browser-based), via a specified IP address, or using a native client on your local computer. When you use Azure Bastion, your virtual machines don't require a client, agent, or additional software. Azure Bastion securely connects to all virtual machines in the virtual network without exposing RDP/SSH ports to the public internet. For more information, see What is Azure Bastion?

For native client connections using Azure CLI, see Connect to a VM using a Linux native client or Connect to a VM using a Windows native client. To connect to a Linux VM using RDP, see Create an RDP connection to a Linux VM.

The following diagram shows the dedicated deployment architecture using an SSH connection.



Prerequisites

Before you begin, verify that you meet the following criteria:

- An Azure Bastion host deployed in the virtual network where the virtual machine is located, or in a peered virtual network. To set up a Bastion host, see Create a bastion host. The SKU you need depends on your connection method:

Expand table

Connection method	Minimum SKU	Additional configuration
Azure portal (browser)	Basic	None
Azure portal with custom ports	Standard	None
IP-based connection	Standard	IP-based connection enabled

Connection method	Minimum SKU	Additional configuration
Native client (SSH)	Standard	Native client support enabled

- A Linux virtual machine in the virtual network (or reachable from the virtual network for IP-based connections).
- **Required roles:**
  - Reader role on the virtual machine.
  - Reader role on the NIC with private IP of the virtual machine.
  - Reader role on the Azure Bastion resource.
  - Reader role on the virtual network of the target virtual machine (if the Bastion deployment is in a peered virtual network).
  - Virtual Machine Administrator Login or Virtual Machine User Login role (only required for Microsoft Entra ID authentication).
- **Ports:** In order to connect to the Linux VM via SSH, you must have the following ports open on your VM:
  - Inbound port: SSH (22) ***or***
  - Inbound port: Custom value (you'll then need to specify this custom port when you connect to the VM via Azure Bastion). This setting isn't available for the Basic or Developer SKU.

See the Azure Bastion FAQ for additional requirements.

Authentication methods

The following table shows which authentication methods are available for each connection method.

Expand table

Authentication method	Supported connection methods	Minimum SKU
Microsoft Entra ID	Azure portal, native client	Basic (portal), Standard (native client)
Username and password	Azure portal, IP address (portal), native client	Basic (portal), Standard (IP address, native client)
Password from Azure Key Vault	Azure portal	Basic
SSH private key from local file	Azure portal, IP address (portal),	Basic (portal), Standard (IP address,

Authentication method	Supported connection methods	Minimum SKU
	native client	native client)
SSH private key from Azure Key Vault	Azure portal	Basic

Authentication details

Configure the authentication settings for your connection. Not all authentication methods are available for every connection method. See the authentication methods table for availability.

Microsoft Entra ID

Available for: Azure portal, native client. Not supported for IP-based connections.

For prerequisites, setup steps, and connection instructions, see Configure Microsoft Entra ID authentication for Azure Bastion.

Key Vault configuration

If you're using Azure Key Vault to store a password or SSH private key, configure your Key Vault using the following requirements:

- If you didn't set up an Azure Key Vault resource, see Create a key vault and store your secret (password or SSH private key) as the value of a new Key Vault secret.
- Make sure you have **List** and **Get** access to the secrets stored in the Key Vault resource. To assign and modify access policies for your Key Vault resource, see Assign a Key Vault access policy.
- Store your secret in Azure Key Vault using the **PowerShell** or **Azure CLI** experience. Storing your secret via the Azure Key Vault portal experience interferes with the formatting and results in unsuccessful login. If you stored your private key as a secret using the portal experience and no longer have access to the original private key file, see Update SSH key to update access to your target VM with a new SSH key pair.

Connect to a virtual machine using SSH

Choose your connection method tab below to see the navigation steps for connecting to your VM. For available authentication methods per connection method, see the authentication methods table.

Azure portal

Use the Azure portal to create a browser-based SSH connection to your Linux virtual machine. This method connects directly through your browser. No native SSH client or additional software is required on your local computer. The Basic SKU or higher is required, or the Standard SKU if you need custom ports.

1. In the Azure portal, go to the virtual machine to which you want to connect. At the top of the virtual machine **Overview** page, select **Connect**, then select **Connect via Bastion** from the dropdown. This opens the **Bastion** page. You can also go to the Bastion page directly in the left pane.

2. On the **Bastion** page, the settings that you can configure depend on the Bastion SKU that your bastion host has been configured to use.

   - If you're using a SKU higher than the Basic SKU, **Connection Settings** values (ports and protocols) are visible and can be configured.
   - If you're using the Basic SKU or Developer SKU, you can't configure **Connection Settings** values. Instead, your connection uses the following default settings: SSH and port 22.
   - To view and select an available **Authentication Type**, use the dropdown.

3. Configure your authentication settings. For details, see Authentication details. Select **Connect**.

Limitations

- **IP-based connections:** IP-based connection doesn't work with force tunneling over VPN, or when a default route is advertised over an ExpressRoute circuit. Azure Bastion requires access to the internet. Force tunneling or default route advertisement results in traffic being dropped.
- **IP-based connections:** UDR isn't supported on the Bastion subnet, including with IP-based connections.
- **IP-based connections:** Microsoft Entra ID authentication isn't supported for IP-based SSH connections. For more information, see Microsoft Entra ID authentication.
- **Native client:** Signing in using an SSH private key stored in Azure Key Vault isn't supported with native client connections.
- **Native client:** This feature isn't supported in Cloud Shell.

Next steps

- Connect to a Linux VM using RDP
- What is Azure Bastion?
- Configure Microsoft Entra ID authentication for identity-based access.
- Transfer files to your virtual machine using a native client.
- Configure a shareable link for users without Azure portal access.
- Azure Bastion FAQ

Last updated on 03/17/2026

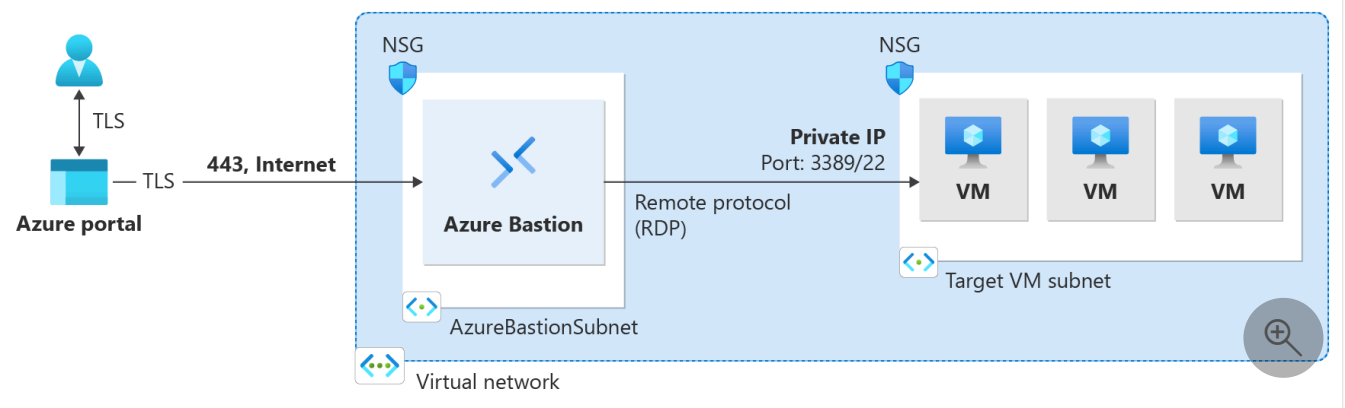
Create an RDP connection to a Windows VM using Azure Bastion

Summarize this article for me

This article describes how to create a secure RDP connection to your Windows virtual machines using Azure Bastion. You can connect through the Azure portal (browser-based), via a specified IP address, or using a native client on your local Windows computer. When you use Azure Bastion, your virtual machines don't require a client, agent, or additional software. Azure Bastion securely connects to all virtual machines in the virtual network without exposing RDP/SSH ports to the public internet. For more information, see What is Azure Bastion?

For native client connections using Azure CLI (including SSH and tunnel), see Connect to a VM using a native client. To connect to a Windows virtual machine using SSH, see Create an SSH connection to a Windows VM.

The following diagram shows the dedicated deployment architecture using an RDP connection.



Prerequisites

Before you begin, verify that you meet the following criteria:

- An Azure Bastion host deployed in the virtual network where the virtual machine is located, or in a peered virtual network. To set up a Bastion host, see Create a bastion host. The SKU you need depends on your connection method:

Expand table

Connection method	Minimum SKU	Additional configuration
Azure portal (browser)	Basic	None
Azure portal with custom ports	Standard	None

Connection method	Minimum SKU	Additional configuration
IP-based connection	Standard	IP-based connection enabled
Native client (RDP)	Standard	Native client support enabled

- Users connecting via RDP must have rights on the target virtual machine. If the user isn't a local administrator, add them to the **Remote Desktop Users** group.
- Azure Bastion uses RDP port 3389 by default. Custom ports require the Standard SKU or higher. To upgrade, see Upgrade a SKU.
- A Windows virtual machine in the virtual network (or reachable from the virtual network for IP-based connections).
- **Required roles:**
  - Reader role on the virtual machine.
  - Reader role on the NIC with the IP of the virtual machine.
  - Reader role on the Azure Bastion resource.
  - Reader role on the virtual network of the target virtual machine (if the Bastion deployment is in a peered virtual network).
  - Virtual Machine Administrator Login or Virtual Machine User Login role (only required for Microsoft Entra ID authentication).

See the Azure Bastion FAQ for additional requirements.

Authentication methods

The following authentication methods are available for RDP connections through Azure Bastion. Select an authentication method to see the corresponding steps.

Expand table

Authentication method	Supported connection methods	Minimum SKU
Microsoft Entra ID (Preview) (Preview for RDP)	Azure portal, native client	Basic (portal), Standard (native client)
Username and password	Azure portal, IP address (portal), native client	Basic (portal), Standard (IP address, native client)
Kerberos	Azure portal	Basic

Connect to a virtual machine using RDP

Select a connection method to see the corresponding steps. After you navigate to the Bastion connection page, choose your authentication method.

Azure portal

Use the Azure portal to create a browser-based RDP connection to your Windows virtual machine. This method connects directly through your browser. No native RDP client or additional software is required on your local computer. The Basic SKU or higher is required, or the Standard SKU if you need custom ports.

1. In the Azure portal, select your virtual machine. On the left pane select **Connect**, then select **Bastion**.
2. In the **Connection settings** tab, select **RDP** as the protocol, and enter the port number if you changed it from the default of 3389.
3. Select your authentication method. Microsoft Entra ID (Preview) is recommended. For other options, see Authentication methods.
4. Select **Connect** to open the RDP connection to your virtual machine in a new browser tab.

> **Note**
>
> For troubleshooting tips, see Troubleshooting RDP connections and Troubleshoot Microsoft Entra sign in for a Windows virtual machine in Azure or Arc-enabled Windows Server

Limitations

- **IP-based connections:** IP-based connection doesn't work with force tunneling over VPN, or when a default route is advertised over an ExpressRoute circuit. Azure Bastion requires access to the Internet and force tunneling, or the default route advertisement, results in traffic blackholing.
- **IP-based connections:** UDR isn't supported on the Bastion subnet, including with IP-based connections.
- **IP-based connections:** Custom ports and protocols aren't currently supported when connecting to a virtual machine via native client with IP-based connections.

- **Microsoft Entra ID:** Microsoft Entra authentication isn't supported for IP-based RDP connections. IP-based SSH connections via native client do support Entra ID authentication. For Entra ID auth details, see About Microsoft Entra ID authentication.
- **Session recording:** RDP + Entra ID authentication in the portal can't be used concurrently with graphical session recording.

Next steps

- Connect to a Windows VM using SSH
- What is Azure Bastion?
- Configure Microsoft Entra ID authentication for identity-based access.
- Configure Kerberos authentication for domain-joined virtual machines.
- Transfer files to your virtual machine using a native client.
- Configure a shareable link for users without Azure portal access.
- Azure Bastion FAQ

Last updated on 03/11/2026

What is Azure role-based access control (Azure RBAC)?

Article • 03/12/2024

Access management for cloud resources is a critical function for any organization that is using the cloud. Azure role-based access control (Azure RBAC) helps you manage who has access to Azure resources, what they can do with those resources, and what areas they have access to.

Azure RBAC is an authorization system built on Azure Resource Manager that provides fine-grained access management to Azure resources.

This video provides a quick overview of Azure RBAC.
https://www.youtube-nocookie.com/embed/Dzhm-garKBM

What can I do with Azure RBAC?

Here are some examples of what you can do with Azure RBAC:

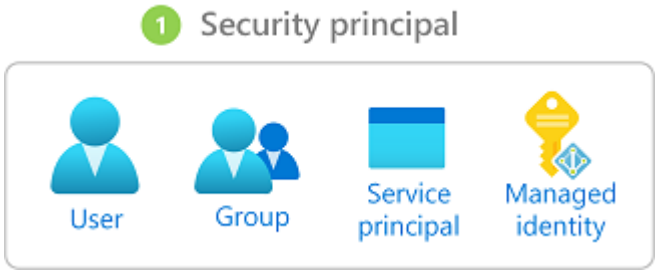
- Allow one user to manage virtual machines in a subscription and another user to manage virtual networks
- Allow a DBA group to manage SQL databases in a subscription
- Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets
- Allow an application to access all resources in a resource group

How Azure RBAC works

The way you control access to resources using Azure RBAC is to assign Azure roles. This is a key concept to understand – it's how permissions are enforced. A role assignment consists of three elements: security principal, role definition, and scope.

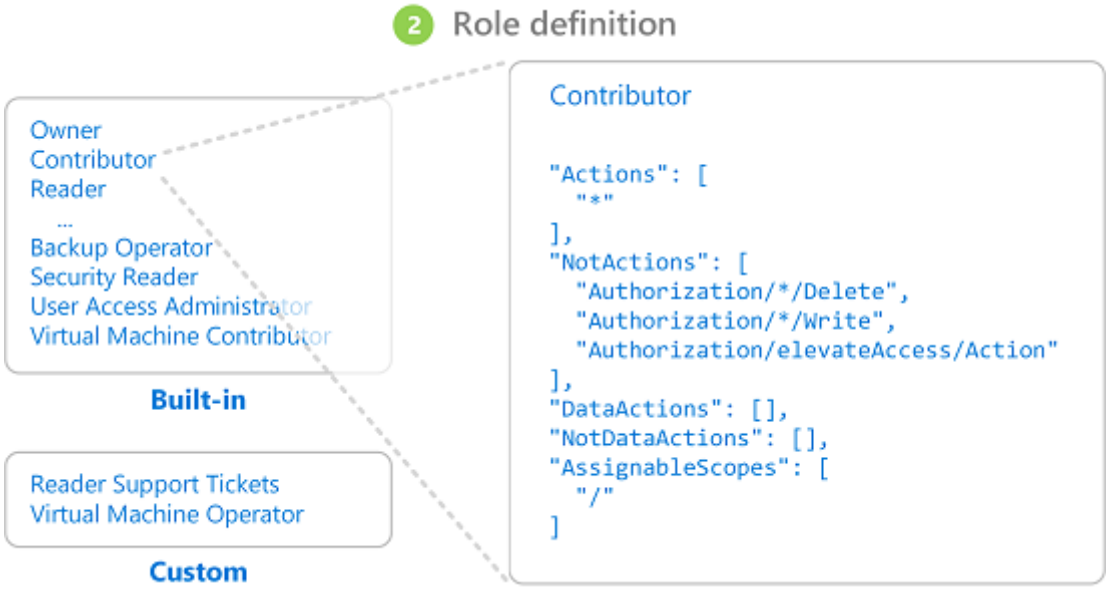
Security principal

A *security principal* is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources. You can assign a role to any of these security principals.



Role definition

A *role definition* is a collection of permissions. It's typically just called a *role*. A role definition lists the actions that can be performed, such as read, write, and delete. Roles can be high-level, like owner, or specific, like virtual machine reader.



Azure includes several built-in roles that you can use. For example, the Virtual Machine Contributor role allows a user to create and manage virtual machines. If the built-in roles don't meet the specific needs of your organization, you can create your own Azure custom roles.

This video provides a quick overview of built-in roles and custom roles.
https://www.youtube-nocookie.com/embed/I1mefHptRgo

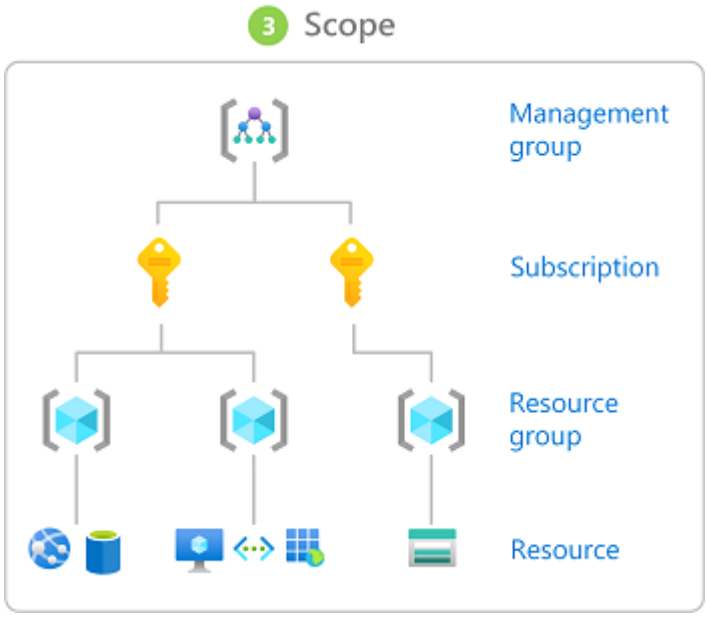
Azure has data actions that enable you to grant access to data within an object. For example, if a user has read data access to a storage account, then they can read the blobs or messages within that storage account.

For more information, see Understand Azure role definitions.

Scope

Scope is the set of resources that the access applies to. When you assign a role, you can further limit the actions allowed by defining a scope. This is helpful if you want to make someone a Website Contributor, but only for one resource group.

In Azure, you can specify a scope at four levels: management group, subscription, resource group, or resource. Scopes are structured in a parent-child relationship. You can assign roles at any of these levels of scope.

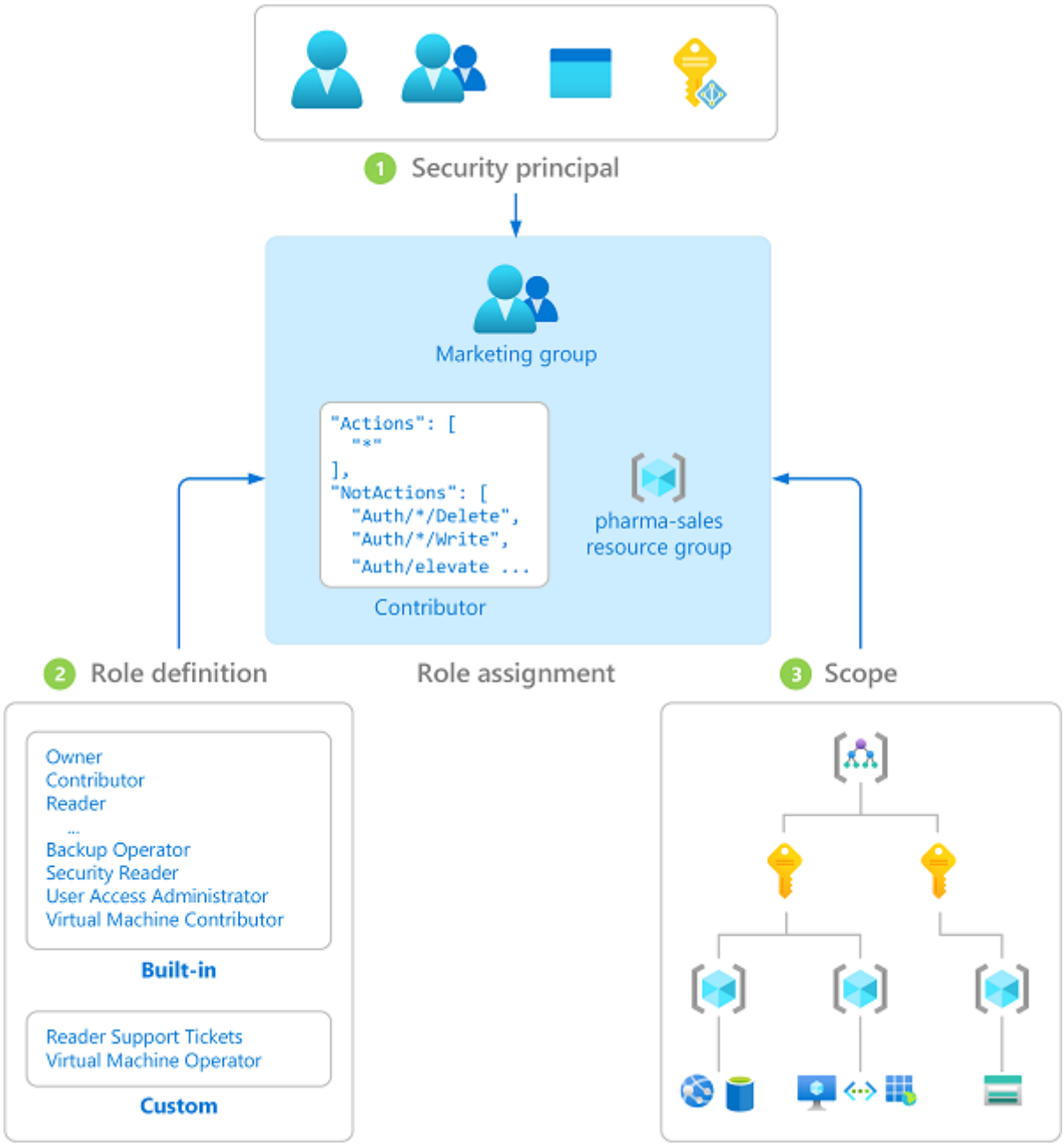


For more information about scope, see Understand scope.

Role assignments

A *role assignment* is the process of attaching a role definition to a user, group, service principal, or managed identity at a particular scope for the purpose of granting access. Access is granted by creating a role assignment, and access is revoked by removing a role assignment.

The following diagram shows an example of a role assignment. In this example, the Marketing group has been assigned the Contributor role for the pharma-sales resource group. This means that users in the Marketing group can create or manage any Azure resource in the pharma-sales resource group. Marketing users don't have access to resources outside the pharma-sales resource group, unless they're part of another role assignment.

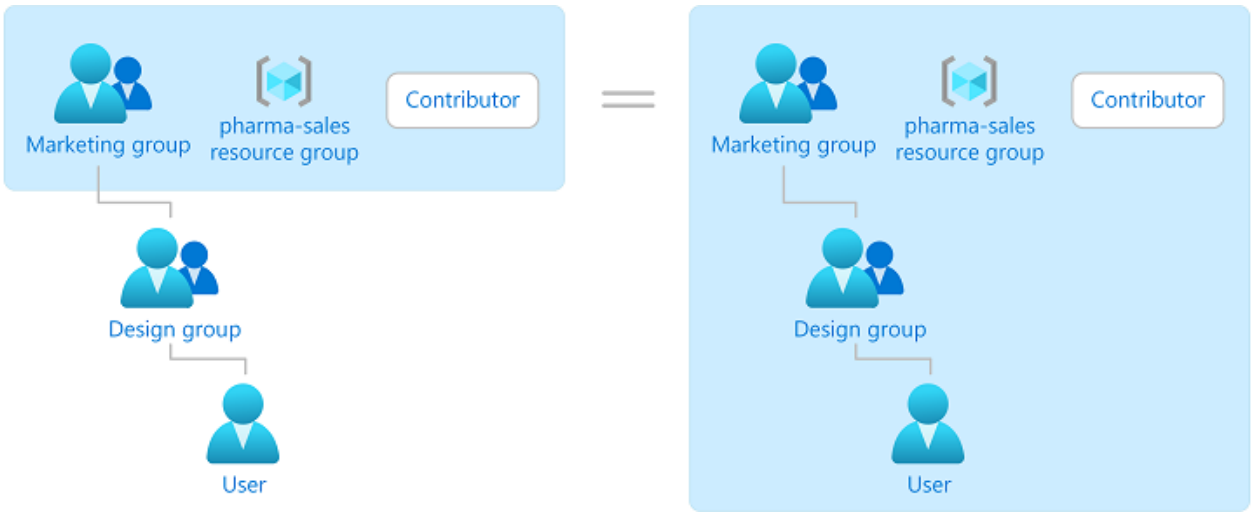


You can assign roles using the Azure portal, Azure CLI, Azure PowerShell, Azure SDKs, or REST APIs.

For more information, see Steps to assign an Azure role.

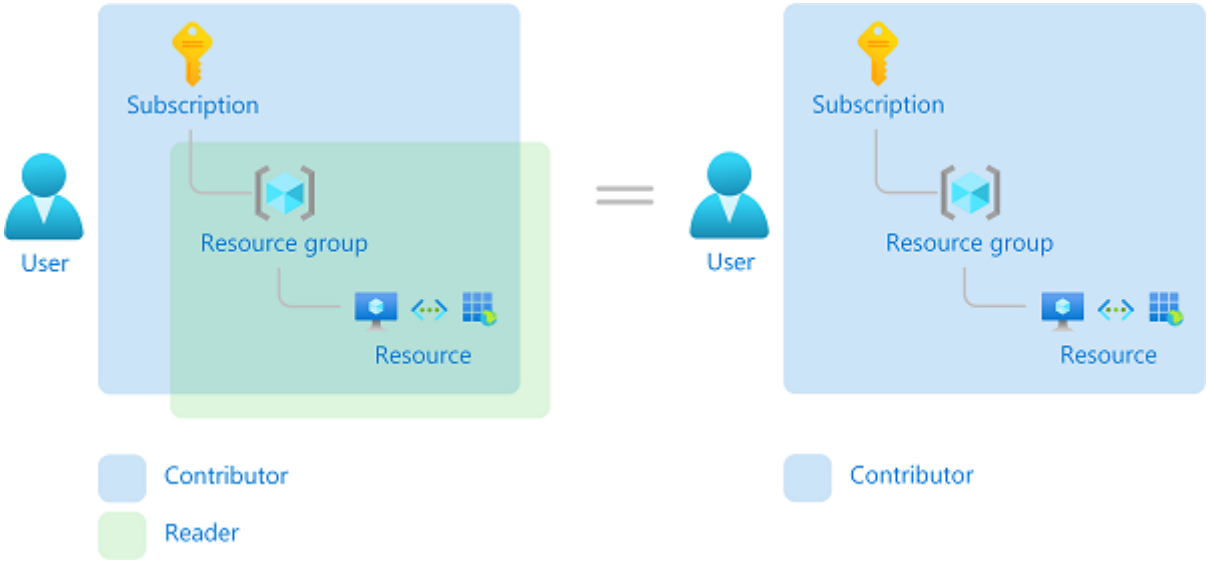
Groups

Role assignments are transitive for groups, which means that if a user is a member of a group and that group is a member of another group that has a role assignment, the user has the permissions in the role assignment.



Multiple role assignments

So what happens if you have multiple overlapping role assignments? Azure RBAC is an additive model, so your effective permissions are the sum of your role assignments. Consider the following example where a user is granted the Contributor role at the subscription scope and the Reader role on a resource group. The sum of the Contributor permissions and the Reader permissions is effectively the Contributor role for the subscription. Therefore, in this case, the Reader role assignment has no impact.



How Azure RBAC determines if a user has access to a resource

The following are the high-level steps that Azure RBAC uses to determine if you have access to a resource. These steps apply to Azure Resource Manager or data plane services integrated with Azure RBAC. This is helpful to understand if you're trying to troubleshoot an access issue.

1. A user (or service principal) acquires a token for Azure Resource Manager.

   The token includes the user's group memberships (including transitive group memberships).

2. The user makes a REST API call to Azure Resource Manager with the token attached.

3. Azure Resource Manager retrieves all the role assignments and deny assignments that apply to the resource upon which the action is being taken.

4. If a deny assignment applies, access is blocked. Otherwise, evaluation continues.

5. Azure Resource Manager narrows the role assignments that apply to this user or their group and determines what roles the user has for this resource.

6. Azure Resource Manager determines if the action in the API call is included in the roles the user has for this resource. If the roles include `Actions` that have a wildcard (`*`), the effective permissions are computed by subtracting the `NotActions` from the allowed `Actions`. Similarly, the same subtraction is done for any data actions.

   ```
   Actions - NotActions = Effective management permissions
   ```

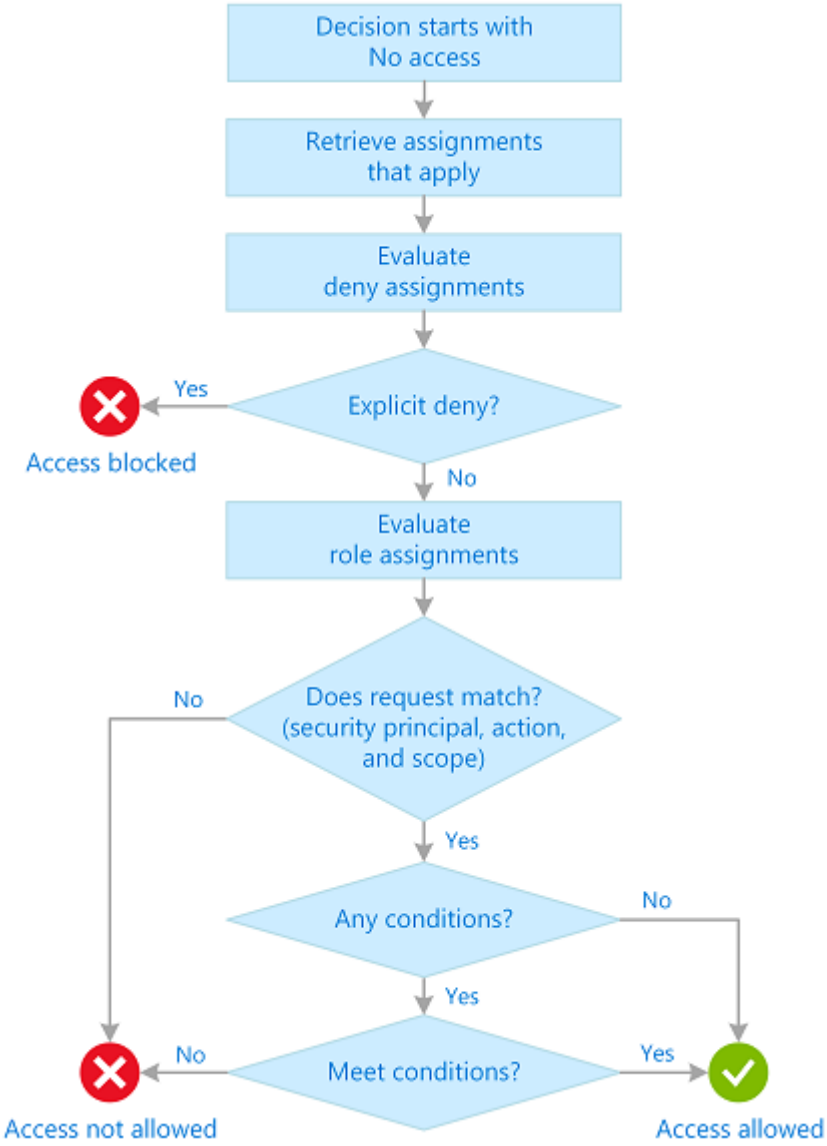
   ```
   DataActions - NotDataActions = Effective data permissions
   ```

7. If the user doesn't have a role with the action at the requested scope, access isn't allowed. Otherwise, any conditions are evaluated.

8. If the role assignment includes conditions, they're evaluated. Otherwise access is allowed.

9. If conditions are met, access is allowed. Otherwise access isn't allowed.

The following diagram is a summary of the evaluation logic.



Where is Azure RBAC data stored?

Role definitions, role assignments, and deny assignments are stored globally to ensure that you have access to your resources regardless of the region you created the resource.

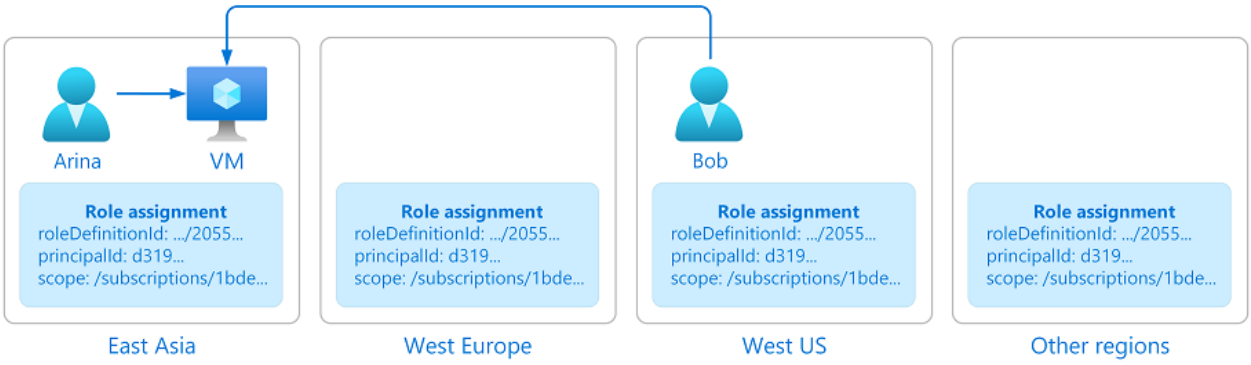
When a role assignment or any other Azure RBAC data is deleted, the data is globally deleted. Principals that had access to a resource via Azure RBAC data will lose their access.

Why is Azure RBAC data global?

Azure RBAC data is global to ensure that customers can timely access resources regardless from where they're accessing. Azure RBAC is enforced by Azure Resource Manager, which has a global endpoint and requests are routed to the nearest region for speed and resilience. Therefore, Azure RBAC must be enforced in all regions and the

data is replicated to all regions. For more information, see Resiliency of Azure Resource Manager.

Consider the following example. Arina creates a virtual machine in East Asia. Bob, who is a member of Arina's team, works in the United States. Bob needs to access the virtual machine that was created in East Asia. To grant Bob timely access to the virtual machine, Azure needs to globally replicate the role assignment that grants Bob access to the virtual machine from anywhere Bob is.



License requirements

Using this feature is free and included in your Azure subscription.

Next steps

- Assign Azure roles using the Azure portal
- Understand the different roles
- Cloud Adoption Framework: Resource access management in Azure

How to set up Key Vault for virtual machines with the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

In the Azure Resource Manager stack, secrets/certificates are modeled as resources that are provided by Key Vault. To learn more about Azure Key Vault, see What is Azure Key Vault? In order for Key Vault to be used with Azure Resource Manager VMs, the *EnabledForDeployment* property on Key Vault must be set to true. This article shows you how to set up Key Vault for use with Azure virtual machines (VMs) using the Azure CLI.

To perform these steps, you need the latest Azure CLI installed and logged in to an Azure account using az login.

Create a Key Vault

Create a key vault and assign the deployment policy with az keyvault create. The following example creates a key vault named `myKeyVault` in the `myResourceGroup` resource group:

Azure CLI

```
az keyvault create -l westus -n myKeyVault -g myResourceGroup --enabled-for-deployment true
```

Update a Key Vault for use with VMs

Set the deployment policy on an existing key vault with az keyvault update. The following updates the key vault named `myKeyVault` in the `myResourceGroup` resource group:

Azure CLI

```
az keyvault update -n myKeyVault -g myResourceGroup --set properties.enabledForDeployment=true
```

Use templates to set up Key Vault

When you use a template, you need to set the `enabledForDeployment` property to `true` for the Key Vault resource as follows:

JSON

```
{
    "type": "Microsoft.KeyVault/vaults",
    "name": "ContosoKeyVault",
    "apiVersion": "2015-06-01",
    "location": "<location-of-key-vault>",
    "properties": {
    "enabledForDeployment": "true",
    ....
    ....
    }
}
```

Next steps

For other options that you can configure when you create a Key Vault by using templates, see Create a key vault.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Set up Key Vault for virtual machines using Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

> ⓘ **Note**
>
> Azure has two different deployment models you can use to create and work with resources: Azure Resource Manager and classic. This article covers the use of the Resource Manager deployment model. We recommend the Resource Manager deployment model for new deployments instead of the classic deployment model.

In Azure Resource Manager stack, secrets/certificates are modeled as resources that are provided by the resource provider of Key Vault. To learn more about Key Vault, see What is Azure Key Vault?

> ⓘ **Note**
>
> 1. In order for Key Vault to be used with Azure Resource Manager virtual machines, the **EnabledForDeployment** property on Key Vault must be set to true. You can do this in various clients.
> 2. The Key Vault needs to be created in the same subscription and location as the Virtual Machine.

Use PowerShell to set up Key Vault

To create a key vault by using PowerShell, see Set and retrieve a secret from Azure Key Vault using PowerShell.

For new key vaults, you can use this PowerShell cmdlet:

Azure PowerShell

```
New-AzKeyVault -VaultName 'ContosoKeyVault' -ResourceGroupName
'ContosoResourceGroup' -Location 'East Asia' -EnabledForDeployment
```

For existing key vaults, you can use this PowerShell cmdlet:

Azure PowerShell

```
Set-AzKeyVaultAccessPolicy -VaultName 'ContosoKeyVault' -
EnabledForDeployment
```

Use CLI to set up Key Vault

To create a key vault by using the command-line interface (CLI), see Manage Key Vault using CLI.

For CLI, you have to create the key vault before you assign the deployment policy. You can do this by using the following command:

Azure CLI

```
az keyvault create --name "ContosoKeyVault" --resource-group
"ContosoResourceGroup" --location "EastAsia"
```

Then to enable Key Vault for use with template deployment, run the following command:

Azure CLI

```
az keyvault update --name "ContosoKeyVault" --resource-group
"ContosoResourceGroup" --enabled-for-deployment "true"
```

Use templates to set up Key Vault

While you use a template, you need to set the `enabledForDeployment` property to `true` for the Key Vault resource.

config

```
{
  "type": "Microsoft.KeyVault/vaults",
  "name": "ContosoKeyVault",
  "apiVersion": "2015-06-01",
  "location": "<location-of-key-vault>",
  "properties": {
    "enabledForDeployment": "true",
    ....
    ....
  }
}
```

For other options that you can configure when you create a key vault by using templates, see Create a key vault.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Guidance for mitigating silicon based micro-architectural and speculative execution side-channel vulnerabilities

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article provides guidance for a new class of silicon based micro-architectural and speculative execution side-channel vulnerabilities that affect many modern processors and operating systems. This includes Intel, AMD, and ARM. Specific details for these silicon-based vulnerabilities can be found in the following security advisories and CVEs:

- ADV180002 - Guidance to mitigate speculative execution side-channel vulnerabilities
- ADV180012 - Microsoft Guidance for Speculative Store Bypass
- ADV180013 - Microsoft Guidance for Rogue System Register Read
- ADV180016 - Microsoft Guidance for Lazy FP State Restore
- ADV180018 - Microsoft Guidance to mitigate L1TF variant
- ADV190013 - Microsoft Guidance to mitigate Microarchitectural Data Sampling vulnerabilities
- ADV220002 - Microsoft Guidance on Intel Processor MMIO Stale Data Vulnerabilities
- CVE-2022-23816 - AMD CPU Branch Type Confusion
- CVE-2022-21123 - AMD CPU Branch Type Confusion

The disclosure of these CPU vulnerabilities has resulted in questions from customers seeking more clarity.

Microsoft has deployed mitigations across all our cloud services. The infrastructure that runs Azure and isolates customer workloads from each other is protected. This means that a potential attacker using the same infrastructure can't attack your application using these vulnerabilities.

Azure is using memory preserving maintenance whenever possible, to minimize customer impact and eliminate the need for reboots. Azure will continue utilizing these methods when making systemwide updates to the host and protect our customers.

More information about how security is integrated into every aspect of Azure is available on the Azure Security Documentation site.

> **Note**
>
> Since this document was first published, multiple variants of this vulnerability class have been disclosed. Microsoft continues to be heavily invested in protecting our customers and providing guidance. This page will be updated as we continue to release further fixes.
>
> **Customers that are running untrusted code within their VM** need to take action to protect against these vulnerabilities by reading below for more guidance on all vulnerabilities.
>
> Other customers should evaluate these vulnerabilities from a Defense in Depth perspective and consider the security and performance implications of their chosen configuration.

Keeping your operating systems up-to-date

While an OS update isn't required to isolate your applications running on Azure from other Azure customers, it is always a best practice to keep your software up-to-date. The latest Security Updates for Windows contain mitigations for these vulnerabilities. Similarly, Linux distributions have released multiple updates to address these vulnerabilities. Here are our recommended actions to update your operating system:

Expand table

Offering	Recommended Action
Azure Cloud Services	Enable auto update or ensure you're running the newest Guest OS.
Azure Linux Virtual Machines	Install updates from your operating system provider. For more information, see Linux later in this document.
Azure Windows Virtual Machines	Install the latest security rollup.
Other Azure PaaS Services	There's no action needed for customers using these services. Azure automatically keeps your OS versions up-to-date.

Additional guidance if you're running untrusted code

Customers who allow untrusted users to execute arbitrary code may wish to implement some extra security features inside their Azure Virtual Machines or Cloud Services. These features protect against the intra-process disclosure vectors that several speculative execution vulnerabilities describe.

Example scenarios where more security features are recommended:

- You allow code that you don't trust to run inside your VM.
  - *For example, you allow one of your customers to upload a binary or script that you then execute within your application*.
- You allow users that you don't trust to log into your VM using low privileged accounts.
  - *For example, you allow a low-privileged user to log into one of your VMs using remote desktop or SSH*.
- You allow untrusted users access to virtual machines implemented via nested virtualization.
  - *For example, you control the Hyper-V host, but allocate the VMs to untrusted users*.

Customers who don't implement a scenario involving untrusted code don't need to enable these extra security features.

Enabling additional security

You can enable more security features inside your VM or Cloud Service if you're running untrusted code. In parallel, ensure your operating system is up-to-date to enable security features inside your VM or Cloud Service

Windows

Your target operating system must be up-to-date to enable these extra security features. While numerous mitigations are enabled by default, the extra features described here must be enabled manually and may cause a performance impact.

Option 1

Step 1: Follow the instructions in KB4072698 to verify protections are enabled using the SpeculationControl PowerShell module.

> **Note**

If you previously downloaded this module, you will need to install the newest version.

To validate enabled protections against these vulnerabilities, see Understanding Get-SpeculationControlSettings PowerShell script output.

If protections aren't enabled, please contact Azure Support to enable additional controls on your Azure VM.

Step 2: To enable Kernel Virtual Address Shadowing (KVAS) and Branch Target Injection (BTI) OS support, follow the instructions in KB4072698 to enable protections using the `Session Manager` registry keys. A reboot is required.

Step 3: For deployments that are using nested virtualization (D3 and E3 only): These instructions apply inside the VM you're using as a Hyper-V host.

1. Follow the instructions in KB4072698 to enable protections using the `MinVmVersionForCpuBasedMitigations` registry keys.
2. Set the hypervisor scheduler type to `Core` by following the instructions here.

Option 2

Disable hyper-threading on the VM - Customers running untrusted code on a hyper-threaded VM might choose to disable hyper-threading or move to a non-hyper-threaded VM size. Reference this doc for a list of hyper-threaded VM sizes (where ratio of vCPU to Core is 2:1). To check if your VM has hyper-threading enabled, refer to the below script using the Windows command line from within the VM.

Type `wmic` to enter the interactive interface. Then type the following command to view the amount of physical and logical processors on the VM.

```
CPU Get NumberOfCores,NumberOfLogicalProcessors /Format:List
```

If the number of logical processors is greater than physical processors (cores), then hyper-threading is enabled. If you're running a hyper-threaded VM, contact Azure Support to get hyper-threading disabled. Once hyper-threading is disabled, support will require a full VM reboot. Refer to Core count to understand why your VM core count decreased.

Option 3

For CVE-2022-23816 and CVE-2022-21123 (AMD CPU Branch Type Confusion), follow both **Option 1** and **Option 2** above.

Linux

Enabling the set of extra security features inside requires that the target operating system be fully up-to-date. Some mitigations will be enabled by default. The following section describes the features which are off by default and/or reliant on hardware support (microcode). Enabling these features may cause a performance impact. Reference your operating system provider's documentation for further instructions

Step 1: Disable hyper-threading on the VM - Customers running untrusted code on a hyper-threaded VM will need to disable hyper-threading or move to a non-hyper-threaded VM. Reference this doc for a list of hyper-threaded VM sizes (where ratio of vCPU to Core is 2:1). To check if you're running a hyper-threaded VM, run the `lscpu` command in the Linux VM.

If `Thread(s) per core = 2`, then hyper-threading has been enabled.

If `Thread(s) per core = 1`, then hyper-threading has been disabled.

Sample output for a VM with hyper-threading enabled:

Console

```
CPU Architecture:      x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
```

If you're running a hyper-threaded VM, contact Azure Support to get hyper-threading disabled. Once hyper-threading is disabled, **support will require a full VM reboot**. Refer to Core count to understand why your VM core count decreased.

Step 2: To mitigate against any of the below CPU based memory vulnerabilities, refer to your operating system provider's documentation:

- Redhat
- SUSE

- Ubuntu

Core count

When a hyper-threaded VM is created, Azure allocates 2 threads per core - these are called vCPUs. When hyper-threading is disabled, Azure removes a thread and surfaces up single threaded cores (physical cores). The ratio of vCPU to CPU is 2:1, so once hyper-threading is disabled, the CPU count in the VM will appear to have decreased by half. For example, a D8_v3 VM is a hyper-threaded VM running on 8 vCPUs (2 threads per core x 4 cores). When hyper-threading is disabled, CPUs will drop to 4 physical cores with 1 thread per core.

Next steps

For more information about how security is integrated into every aspect of Azure, see Azure Security Documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Join a Red Hat Enterprise Linux virtual machine to a Microsoft Entra Domain Services managed domain

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to a Microsoft Entra Domain Services managed domain. When you join a VM to a Domain Services managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join a Red Hat Enterprise Linux (RHEL) VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- A Microsoft Entra tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create a Microsoft Entra tenant or associate an Azure subscription with your account.
- A Microsoft Entra Domain Services managed domain enabled and configured in your Microsoft Entra tenant.
  - If needed, the first tutorial creates and configures a Microsoft Entra Domain Services managed domain.
- A user account that's a part of the managed domain.
- Unique Linux VM names that are a maximum of 15 characters to avoid truncated names that might cause conflicts in Active Directory.

Create and connect to a RHEL Linux VM

If you have an existing RHEL Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create a RHEL Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Microsoft Entra admin center
- Azure CLI

- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Microsoft Entra Domain Services.
- Deploy the VM into a different subnet than your Microsoft Entra Domain Services managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and set the hostname:

Bash

```
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *rhel* is the hostname of your RHEL VM that you're joining to the managed domain.

Update these names with your own values:

config

```
127.0.0.1 rhel rhel.aaddscontoso.com
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

RHEL 6

> **Important**
>
> Keep in consideration Red Hat Enterprise Linux 6.X and Oracle Linux 6.x is already EOL. RHEL 6.10 has available ELS support, which ended on 06/2024.

Install required packages

The VM needs some additional packages to join the VM to the managed domain. To install and configure these packages, update and install the domain-join tools using `yum`.

```
Bash
sudo yum install adcli sssd authconfig krb5-workstation
```

Join VM to the managed domain

Now that the required packages are installed on the VM, join the VM to the managed domain.

1. Use the `adcli info` command to discover the managed domain. The following example discovers the realm *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   ```
   Bash
   sudo adcli info aaddscontoso.com
   ```

   If the `adcli info` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network are updated to point to the domain controllers of the managed domain.

2. First, join the domain using the `adcli join` command. This command also creates the keytab to authenticate the machine. Use a user account that's a part of the managed domain.

   ```
   Bash
   sudo adcli join aaddscontoso.com -U contosoadmin
   ```

3. Now configure the `/ect/krb5.conf` and create the `/etc/sssd/sssd.conf` files to use the `aaddscontoso.com` Active Directory domain. Make sure that `AADDSCONTOSO.COM` is replaced by your own domain name:

   Open the `/etc/krb5.conf` file with an editor:

Bash

```
sudo vi /etc/krb5.conf
```

Update the `krb5.conf` file to match the following sample:

config

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = AADDSCONTOSO.COM
 dns_lookup_realm = true
 dns_lookup_kdc = true
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true

[realms]
 AADDSCONTOSO.COM = {
 kdc = AADDSCONTOSO.COM
 admin_server = AADDSCONTOSO.COM
 }

[domain_realm]
 .AADDSCONTOSO.COM = AADDSCONTOSO.COM
 AADDSCONTOSO.COM = AADDSCONTOSO.COM
```

Create the `/etc/sssd/sssd.conf` file:

Bash

```
sudo vi /etc/sssd/sssd.conf
```

Update the `sssd.conf` file to match the following sample:

config

```
[sssd]
 services = nss, pam, ssh, autofs
 config_file_version = 2
 domains = AADDSCONTOSO.COM

[domain/AADDSCONTOSO.COM]

 id_provider = ad
```

4. Make sure `/etc/sssd/sssd.conf` permissions are 600 and is owned by root user:

```Bash
sudo chmod 600 /etc/sssd/sssd.conf
sudo chown root:root /etc/sssd/sssd.conf
```

5. Use `authconfig` to instruct the VM about the AD Linux integration:

```Bash
sudo authconfig --enablesssd --enablesssd auth --update
```

6. Start and enable the sssd service:

```Bash
sudo service sssd start
sudo chkconfig sssd on
```

If your VM can't successfully complete the domain-join process, make sure that the VM's network security group allows outbound Kerberos traffic on TCP + UDP port 464 to the virtual network subnet for your managed domain.

Now check if you can query user AD information using `getent`

```Bash
sudo getent passwd contosoadmin
```

Allow password authentication for SSH

By default, users can only sign in to a VM using SSH public key-based authentication. Password-based authentication fails. When you join the VM to a managed domain, those domain accounts need to use password-based authentication. Update the SSH configuration to allow password-based authentication as follows.

1. Open the *sshd_conf* file with an editor:

```Bash
sudo vi /etc/ssh/sshd_config
```

2. Update the line for *PasswordAuthentication* to *yes*:

```
config
PasswordAuthentication yes
```

When done, save and exit the *sshd_conf* file using the `:wq` command of the editor.

3. To apply the changes and let users sign in using a password, restart the SSH service for your RHEL distro version:

```
Bash
sudo service sshd restart
```

Grant the 'AAD DC Administrators' group sudo privileges

To grant members of the *AAD DC Administrators* group administrative privileges on the RHEL VM, you add an entry to the */etc/sudoers*. Once added, members of the *AAD DC Administrators* group can use the `sudo` command on the RHEL VM.

1. Open the *sudoers* file for editing:

```
Bash
sudo visudo
```

2. Add the following entry to the end of */etc/sudoers* file. The *AAD DC Administrators* group contains whitespace in the name, so include the backslash escape character in the group name. Add your own domain name, such as *aaddscontoso.com*:

```
config
# Add 'AAD DC Administrators' group members as admins.
%AAD\ DC\ Administrators@aaddscontoso.com ALL=(ALL) NOPASSWD:ALL
```

When done, save and exit the editor using the `:wq` command of the editor.

Sign in to the VM using a domain account

To verify that the VM successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory is created, and that group

membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *rhel.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

   ```
   Bash
   ssh -l contosoadmin@AADDSCONTOSO.com rhel.aaddscontoso.com
   ```

2. When you've successfully connected to the VM, verify that the home directory was initialized correctly:

   ```
   Bash
   pwd
   ```

   You should be in the */home* directory with your own directory that matches the user account.

3. Now check that the group memberships are being resolved correctly:

   ```
   Bash
   id
   ```

   You should see your group memberships from the managed domain.

4. If you signed in to the VM as a member of the *AAD DC Administrators* group, check that you can correctly use the `sudo` command:

   ```
   Bash
   sudo yum update
   ```

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Last updated on 02/13/2025

Join an Ubuntu Linux virtual machine to a Microsoft Entra Domain Services managed domain

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to a Microsoft Entra Domain Services managed domain. When you join a VM to a Domain Services managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join an Ubuntu Linux VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- A Microsoft Entra tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create a Microsoft Entra tenant or associate an Azure subscription with your account.
- A Microsoft Entra Domain Services managed domain enabled and configured in your Microsoft Entra tenant.
  - If needed, the first tutorial creates and configures a Microsoft Entra Domain Services managed domain.
- A user account that's a part of the managed domain. Make sure the SAMAccountName attribute for the user isn't autogenerated. If multiple user accounts in the Microsoft Entra tenant have the same mailNickname attribute, the SAMAccountName attribute for each user is autogenerated. For more information, see How objects and credentials are synchronized in a Microsoft Entra Domain Services managed domain.
- Unique Linux VM names that are a maximum of 15 characters to avoid truncated names that might cause conflicts in Active Directory.

Create and connect to an Ubuntu Linux VM

If you have an existing Ubuntu Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create an Ubuntu Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Microsoft Entra admin center
- Azure CLI
- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Microsoft Entra Domain Services.
- Deploy the VM into a different subnet than your Microsoft Entra Domain Services managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and set the hostname:

```Bash
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *ubuntu* is the hostname of your Ubuntu VM that you're joining to the managed domain.

Update these names with your own values:

```config
127.0.0.1 ubuntu.aaddscontoso.com ubuntu
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

Install required packages

The VM needs some additional packages to join the VM to the managed domain. To install and configure these packages, update and install the domain-join tools using `apt-get`

During the Kerberos installation, the *krb5-user* package prompts for the realm name in ALL UPPERCASE. For example, if the name of your managed domain is *aaddscontoso.com*, enter *AADDSCONTOSO.COM* as the realm. The installation writes the `[realm]` and `[domain_realm]` sections in */etc/krb5.conf* configuration file. Make sure that you specify the realm an ALL UPPERCASE:

Bash

```
sudo apt-get update
sudo apt-get install krb5-user samba sssd sssd-tools libnss-sss libpam-sss ntp
ntpdate realmd adcli
```

Configure Network Time Protocol (NTP)

For domain communication to work correctly, the date and time of your Ubuntu VM must synchronize with the managed domain. Add your managed domain's NTP hostname to the */etc/ntp.conf* file.

1. Open the *ntp.conf* file with an editor:

   Bash

   ```
   sudo vi /etc/ntp.conf
   ```

2. In the *ntp.conf* file, create a line to add your managed domain's DNS name. In the following example, an entry for *aaddscontoso.com* is added. Use your own DNS name:

   config

   ```
   server aaddscontoso.com
   ```

   When done, save and exit the *ntp.conf* file using the `:wq` command of the editor.

3. To make sure that the VM is synchronized with the managed domain, the following steps are needed:

   - Stop the NTP server
   - Update the date and time from the managed domain
   - Start the NTP service

   To complete these steps, run the following commands. Use your own DNS name with the `ntpdate` command:

Bash

```
sudo systemctl stop ntp
sudo ntpdate aaddscontoso.com
sudo systemctl start ntp
```

Join VM to the managed domain

Now that the required packages are installed on the VM and NTP is configured, join the VM to the managed domain.

1. Use the `realm discover` command to discover the managed domain. The following example discovers the realm *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   Bash

   ```
   sudo realm discover AADDSCONTOSO.COM
   ```

   If the `realm discover` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.

2. Now initialize Kerberos using the `kinit` command. Specify a user that's a part of the managed domain. If needed, add a user account to a group in Microsoft Entra ID.

   Again, the managed domain name must be entered in ALL UPPERCASE. In the following example, the account named `contosoadmin@aaddscontoso.com` is used to initialize Kerberos. Enter your own user account that's a part of the managed domain:

   Bash

   ```
   sudo kinit -V contosoadmin@AADDSCONTOSO.COM
   ```

3. Finally, join the VM to the managed domain using the `realm join` command. Use the same user account that's a part of the managed domain that you specified in the previous `kinit` command, such as `contosoadmin@AADDSCONTOSO.COM`:

```bash
sudo realm join --verbose AADDSCONTOSO.COM -U 'contosoadmin@AADDSCONTOSO.COM' --install=/
```

It takes a few moments to join the VM to the managed domain. The following example output shows the VM has successfully joined to the managed domain:

```output
Successfully enrolled machine in realm
```

If your VM can't successfully complete the domain-join process, make sure that the VM's network security group allows outbound Kerberos traffic on TCP + UDP port 464 to the virtual network subnet for your managed domain.

If you received the error *Unspecified GSS failure. Minor code can provide more information (Server not found in Kerberos database)*, open the file */etc/krb5.conf* and add the following code in `[libdefaults]` section and try again:

```config
rdns=false
```

Update the SSSD configuration

One of the packages installed in a previous step was for System Security Services Daemon (SSSD). When a user tries to sign in to a VM using domain credentials, SSSD relays the request to an authentication provider. In this scenario, SSSD uses Domain Services to authenticate the request.

1. Open the *sssd.conf* file with an editor:

   ```bash
   sudo vi /etc/sssd/sssd.conf
   ```

2. Comment out the line for *use_fully_qualified_names* as follows:

   ```config
   # use_fully_qualified_names = True
   ```

   When done, save and exit the *sssd.conf* file using the `:wq` command of the editor.

3. To apply the change, restart the SSSD service:

```bash
sudo systemctl restart sssd
```

Configure user account and group settings

With the VM joined to the managed domain and configured for authentication, there are a few user configuration options to complete. These configuration changes include allowing password-based authentication, and automatically creating home directories on the local VM when domain users first sign in.

Allow password authentication for SSH

By default, users can only sign in to a VM using SSH public key-based authentication. Password-based authentication fails. When you join the VM to a managed domain, those domain accounts need to use password-based authentication. Update the SSH configuration to allow password-based authentication as follows.

> **Note**
>
> Ubuntu marketplace images usually have a few configuration options set under /etc/ssh/sshd_config.d, including *PasswordAuthentication* in the 50-cloud-init.conf file, so make sure you also update that file to avoid it overwriting the one set with the steps below.

1. Open the *sshd_conf* file with an editor:

```bash
sudo vi /etc/ssh/sshd_config
```

2. Update the line for *PasswordAuthentication* to *yes*:

```config
PasswordAuthentication yes
```

   When done, save and exit the *sshd_conf* file using the `:wq` command of the editor.

3. To apply the changes and let users sign in using a password, restart the SSH service:

```bash
sudo systemctl restart ssh
```

Configure automatic home directory creation

To enable automatic creation of the home directory when a user first signs in, complete the following steps:

1. Open the `/etc/pam.d/common-session` file in an editor:

   ```bash
   sudo vi /etc/pam.d/common-session
   ```

2. Add the following line in this file below the line `session optional pam_sss.so`:

   ```config
   session required pam_mkhomedir.so skel=/etc/skel/ umask=0077
   ```

   When done, save and exit the *common-session* file using the `:wq` command of the editor.

Grant the 'AAD DC Administrators' group sudo privileges

To grant members of the *AAD DC Administrators* group administrative privileges on the Ubuntu VM, you add an entry to the */etc/sudoers*. Once added, members of the *AAD DC Administrators* group can use the `sudo` command on the Ubuntu VM.

1. Open the *sudoers* file for editing:

   ```bash
   sudo visudo
   ```

2. Add the following entry to the end of */etc/sudoers* file:

   ```config
   # Add 'AAD DC Administrators' group members as admins.
   %AAD\ DC\ Administrators ALL=(ALL) NOPASSWD:ALL
   ```

   When done, save and exit the editor using the `Ctrl-X` command.

Sign in to the VM using a domain account

To verify that the VM successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory is created, and that group membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *ubuntu.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

   ```Bash
   sudo ssh -l contosoadmin@AADDSCONTOSO.com ubuntu.aaddscontoso.com
   ```

2. When you've successfully connected to the VM, verify that the home directory was initialized correctly:

   ```Bash
   sudo pwd
   ```

   You should be in the */home* directory with your own directory that matches the user account.

3. Now check that the group memberships are being resolved correctly:

   ```Bash
   sudo id
   ```

   You should see your group memberships from the managed domain.

4. If you signed in to the VM as a member of the *AAD DC Administrators* group, check that you can correctly use the `sudo` command:

   ```Bash
   sudo apt-get update
   ```

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Last updated on 02/13/2025

Join an Ubuntu Linux virtual machine to a Microsoft Entra Domain Services managed domain

To let users sign in to virtual machines (VMs) in Azure using a single set of credentials, you can join VMs to a Microsoft Entra Domain Services managed domain. When you join a VM to a Domain Services managed domain, user accounts and credentials from the domain can be used to sign in and manage servers. Group memberships from the managed domain are also applied to let you control access to files or services on the VM.

This article shows you how to join an Ubuntu Linux VM to a managed domain.

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- An active Azure subscription.
  - If you don't have an Azure subscription, create an account.
- A Microsoft Entra tenant associated with your subscription, either synchronized with an on-premises directory or a cloud-only directory.
  - If needed, create a Microsoft Entra tenant or associate an Azure subscription with your account.
- A Microsoft Entra Domain Services managed domain enabled and configured in your Microsoft Entra tenant.
  - If needed, the first tutorial creates and configures a Microsoft Entra Domain Services managed domain.
- A user account that's a part of the managed domain. Make sure the SAMAccountName attribute for the user isn't autogenerated. If multiple user accounts in the Microsoft Entra tenant have the same mailNickname attribute, the SAMAccountName attribute for each user is autogenerated. For more information, see How objects and credentials are synchronized in a Microsoft Entra Domain Services managed domain.
- Unique Linux VM names that are a maximum of 15 characters to avoid truncated names that might cause conflicts in Active Directory.

Create and connect to an Ubuntu Linux VM

If you have an existing Ubuntu Linux VM in Azure, connect to it using SSH, then continue on to the next step to start configuring the VM.

If you need to create an Ubuntu Linux VM, or want to create a test VM for use with this article, you can use one of the following methods:

- Microsoft Entra admin center
- Azure CLI
- Azure PowerShell

When you create the VM, pay attention to the virtual network settings to make sure that the VM can communicate with the managed domain:

- Deploy the VM into the same, or a peered, virtual network in which you have enabled Microsoft Entra Domain Services.
- Deploy the VM into a different subnet than your Microsoft Entra Domain Services managed domain.

Once the VM is deployed, follow the steps to connect to the VM using SSH.

Configure the hosts file

To make sure that the VM host name is correctly configured for the managed domain, edit the */etc/hosts* file and set the hostname:

```Bash
sudo vi /etc/hosts
```

In the *hosts* file, update the *localhost* address. In the following example:

- *aaddscontoso.com* is the DNS domain name of your managed domain.
- *ubuntu* is the hostname of your Ubuntu VM that you're joining to the managed domain.

Update these names with your own values:

```config
127.0.0.1 ubuntu.aaddscontoso.com ubuntu
```

When done, save and exit the *hosts* file using the `:wq` command of the editor.

Install required packages

The VM needs some additional packages to join the VM to the managed domain. To install and configure these packages, update and install the domain-join tools using `apt-get`

During the Kerberos installation, the *krb5-user* package prompts for the realm name in ALL UPPERCASE. For example, if the name of your managed domain is *aaddscontoso.com*, enter *AADDSCONTOSO.COM* as the realm. The installation writes the `[realm]` and `[domain_realm]` sections in */etc/krb5.conf* configuration file. Make sure that you specify the realm an ALL UPPERCASE:

Bash

```
sudo apt-get update
sudo apt-get install krb5-user samba sssd sssd-tools libnss-sss libpam-sss ntp
ntpdate realmd adcli
```

Configure Network Time Protocol (NTP)

For domain communication to work correctly, the date and time of your Ubuntu VM must synchronize with the managed domain. Add your managed domain's NTP hostname to the */etc/ntp.conf* file.

1. Open the *ntp.conf* file with an editor:

   Bash

   ```
   sudo vi /etc/ntp.conf
   ```

2. In the *ntp.conf* file, create a line to add your managed domain's DNS name. In the following example, an entry for *aaddscontoso.com* is added. Use your own DNS name:

   config

   ```
   server aaddscontoso.com
   ```

   When done, save and exit the *ntp.conf* file using the `:wq` command of the editor.

3. To make sure that the VM is synchronized with the managed domain, the following steps are needed:

   - Stop the NTP server
   - Update the date and time from the managed domain
   - Start the NTP service

   To complete these steps, run the following commands. Use your own DNS name with the `ntpdate` command:

Bash

```
sudo systemctl stop ntp
sudo ntpdate aaddscontoso.com
sudo systemctl start ntp
```

Join VM to the managed domain

Now that the required packages are installed on the VM and NTP is configured, join the VM to the managed domain.

1. Use the `realm discover` command to discover the managed domain. The following example discovers the realm *AADDSCONTOSO.COM*. Specify your own managed domain name in ALL UPPERCASE:

   Bash

   ```
   sudo realm discover AADDSCONTOSO.COM
   ```

   If the `realm discover` command can't find your managed domain, review the following troubleshooting steps:

   - Make sure that the domain is reachable from the VM. Try `ping aaddscontoso.com` to see if a positive reply is returned.
   - Check that the VM is deployed to the same, or a peered, virtual network in which the managed domain is available.
   - Confirm that the DNS server settings for the virtual network have been updated to point to the domain controllers of the managed domain.

2. Now initialize Kerberos using the `kinit` command. Specify a user that's a part of the managed domain. If needed, add a user account to a group in Microsoft Entra ID.

   Again, the managed domain name must be entered in ALL UPPERCASE. In the following example, the account named `contosoadmin@aaddscontoso.com` is used to initialize Kerberos. Enter your own user account that's a part of the managed domain:

   Bash

   ```
   sudo kinit -V contosoadmin@AADDSCONTOSO.COM
   ```

3. Finally, join the VM to the managed domain using the `realm join` command. Use the same user account that's a part of the managed domain that you specified in the previous `kinit` command, such as `contosoadmin@AADDSCONTOSO.COM`:

```bash
sudo realm join --verbose AADDSCONTOSO.COM -U 'contosoadmin@AADDSCONTOSO.COM' --install=/
```

It takes a few moments to join the VM to the managed domain. The following example output shows the VM has successfully joined to the managed domain:

```output
Successfully enrolled machine in realm
```

If your VM can't successfully complete the domain-join process, make sure that the VM's network security group allows outbound Kerberos traffic on TCP + UDP port 464 to the virtual network subnet for your managed domain.

If you received the error *Unspecified GSS failure. Minor code can provide more information (Server not found in Kerberos database)*, open the file */etc/krb5.conf* and add the following code in `[libdefaults]` section and try again:

```config
rdns=false
```

Update the SSSD configuration

One of the packages installed in a previous step was for System Security Services Daemon (SSSD). When a user tries to sign in to a VM using domain credentials, SSSD relays the request to an authentication provider. In this scenario, SSSD uses Domain Services to authenticate the request.

1. Open the *sssd.conf* file with an editor:

   ```bash
   sudo vi /etc/sssd/sssd.conf
   ```

2. Comment out the line for *use_fully_qualified_names* as follows:

   ```config
   # use_fully_qualified_names = True
   ```

   When done, save and exit the *sssd.conf* file using the `:wq` command of the editor.

3. To apply the change, restart the SSSD service:

```bash
sudo systemctl restart sssd
```

Configure user account and group settings

With the VM joined to the managed domain and configured for authentication, there are a few user configuration options to complete. These configuration changes include allowing password-based authentication, and automatically creating home directories on the local VM when domain users first sign in.

Allow password authentication for SSH

By default, users can only sign in to a VM using SSH public key-based authentication. Password-based authentication fails. When you join the VM to a managed domain, those domain accounts need to use password-based authentication. Update the SSH configuration to allow password-based authentication as follows.

> **Note**
>
> Ubuntu marketplace images usually have a few configuration options set under /etc/ssh/sshd_config.d, including *PasswordAuthentication* in the 50-cloud-init.conf file, so make sure you also update that file to avoid it overwriting the one set with the steps below.

1. Open the *sshd_conf* file with an editor:

```bash
sudo vi /etc/ssh/sshd_config
```

2. Update the line for *PasswordAuthentication* to *yes*:

```config
PasswordAuthentication yes
```

   When done, save and exit the *sshd_conf* file using the `:wq` command of the editor.

3. To apply the changes and let users sign in using a password, restart the SSH service:

```bash
sudo systemctl restart ssh
```

Configure automatic home directory creation

To enable automatic creation of the home directory when a user first signs in, complete the following steps:

1. Open the `/etc/pam.d/common-session` file in an editor:

   ```bash
   sudo vi /etc/pam.d/common-session
   ```

2. Add the following line in this file below the line `session optional pam_sss.so`:

   ```config
   session required pam_mkhomedir.so skel=/etc/skel/ umask=0077
   ```

   When done, save and exit the *common-session* file using the `:wq` command of the editor.

Grant the 'AAD DC Administrators' group sudo privileges

To grant members of the *AAD DC Administrators* group administrative privileges on the Ubuntu VM, you add an entry to the */etc/sudoers*. Once added, members of the *AAD DC Administrators* group can use the `sudo` command on the Ubuntu VM.

1. Open the *sudoers* file for editing:

   ```bash
   sudo visudo
   ```

2. Add the following entry to the end of */etc/sudoers* file:

   ```config
   # Add 'AAD DC Administrators' group members as admins.
   %AAD\ DC\ Administrators ALL=(ALL) NOPASSWD:ALL
   ```

   When done, save and exit the editor using the `Ctrl-X` command.

Sign in to the VM using a domain account

To verify that the VM successfully joined to the managed domain, start a new SSH connection using a domain user account. Confirm that a home directory is created, and that group membership from the domain is applied.

1. Create a new SSH connection from your console. Use a domain account that belongs to the managed domain using the `ssh -l` command, such as `contosoadmin@aaddscontoso.com` and then enter the address of your VM, such as *ubuntu.aaddscontoso.com*. If you use the Azure Cloud Shell, use the public IP address of the VM rather than the internal DNS name.

   ```Bash
   sudo ssh -l contosoadmin@AADDSCONTOSO.com ubuntu.aaddscontoso.com
   ```

2. When you've successfully connected to the VM, verify that the home directory was initialized correctly:

   ```Bash
   sudo pwd
   ```

   You should be in the */home* directory with your own directory that matches the user account.

3. Now check that the group memberships are being resolved correctly:

   ```Bash
   sudo id
   ```

   You should see your group memberships from the managed domain.

4. If you signed in to the VM as a member of the *AAD DC Administrators* group, check that you can correctly use the `sudo` command:

   ```Bash
   sudo apt-get update
   ```

Next steps

If you have problems connecting the VM to the managed domain or signing in with a domain account, see Troubleshooting domain join issues.

Last updated on 02/13/2025

Configure managed identities on Azure virtual machines (VMs)

Managed identities for Azure resources provide Azure services with an automatically managed identity in Microsoft Entra ID. You can use this identity to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

For information about Azure Policy definition and details, see Use Azure Policy to assign managed identities (preview).

In this article, you learn how to enable and disable system and user-assigned managed identities for an Azure Virtual Machine (VM), using the Azure portal.

Prerequisites

- If you're unfamiliar with managed identities for Azure resources, check out the overview section.
- If you don't already have an Azure account, sign up for a free account before continuing.

System-assigned managed identity

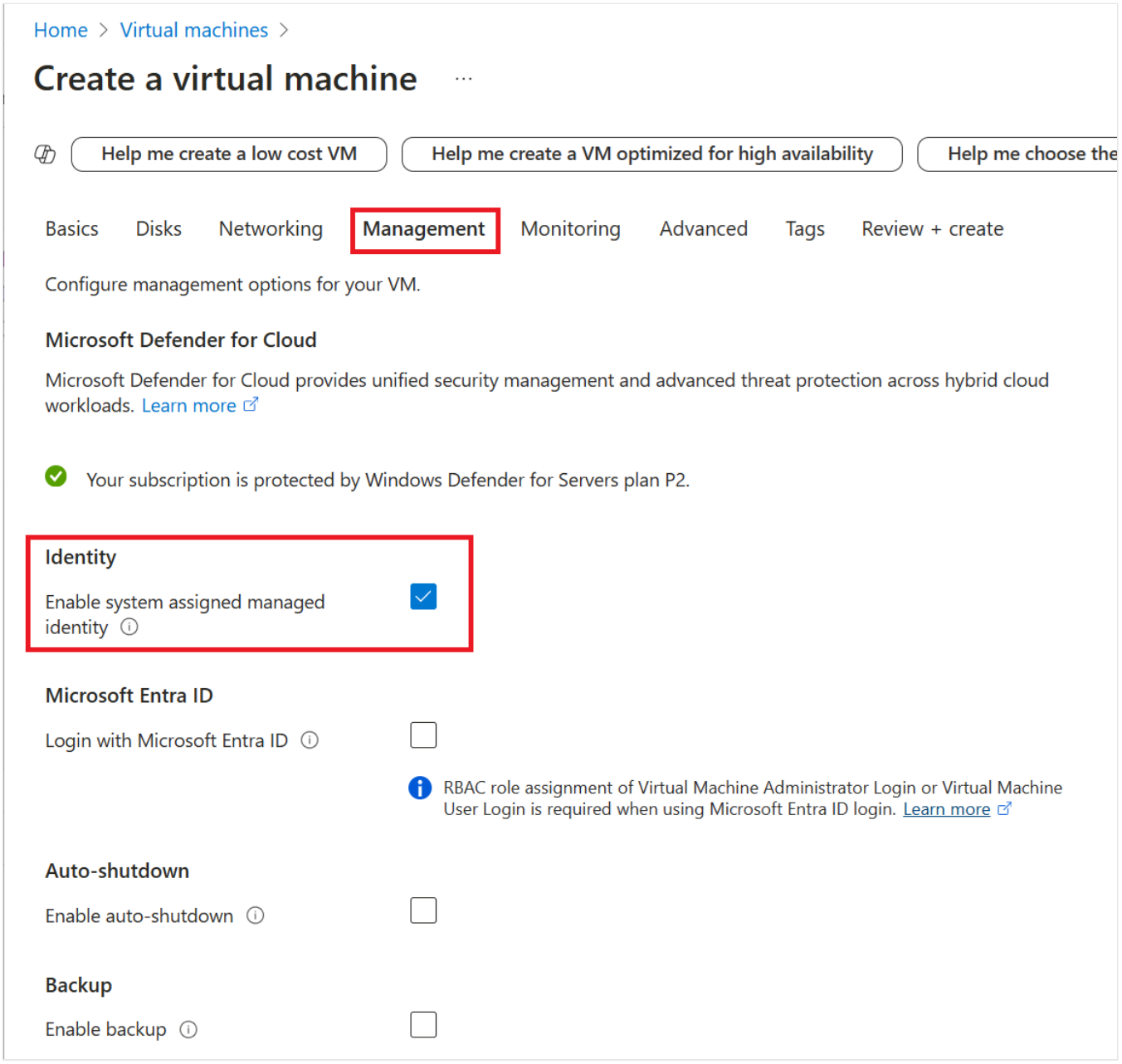
In this section, you learn how to enable and disable the system-assigned managed identity for VM using the Azure portal.

Enable system-assigned managed identity during creation of a VM

To enable system-assigned managed identity on a VM during its creation, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

When creating a Windows virtual machine or Linux virtual machine, select the **Management** tab.

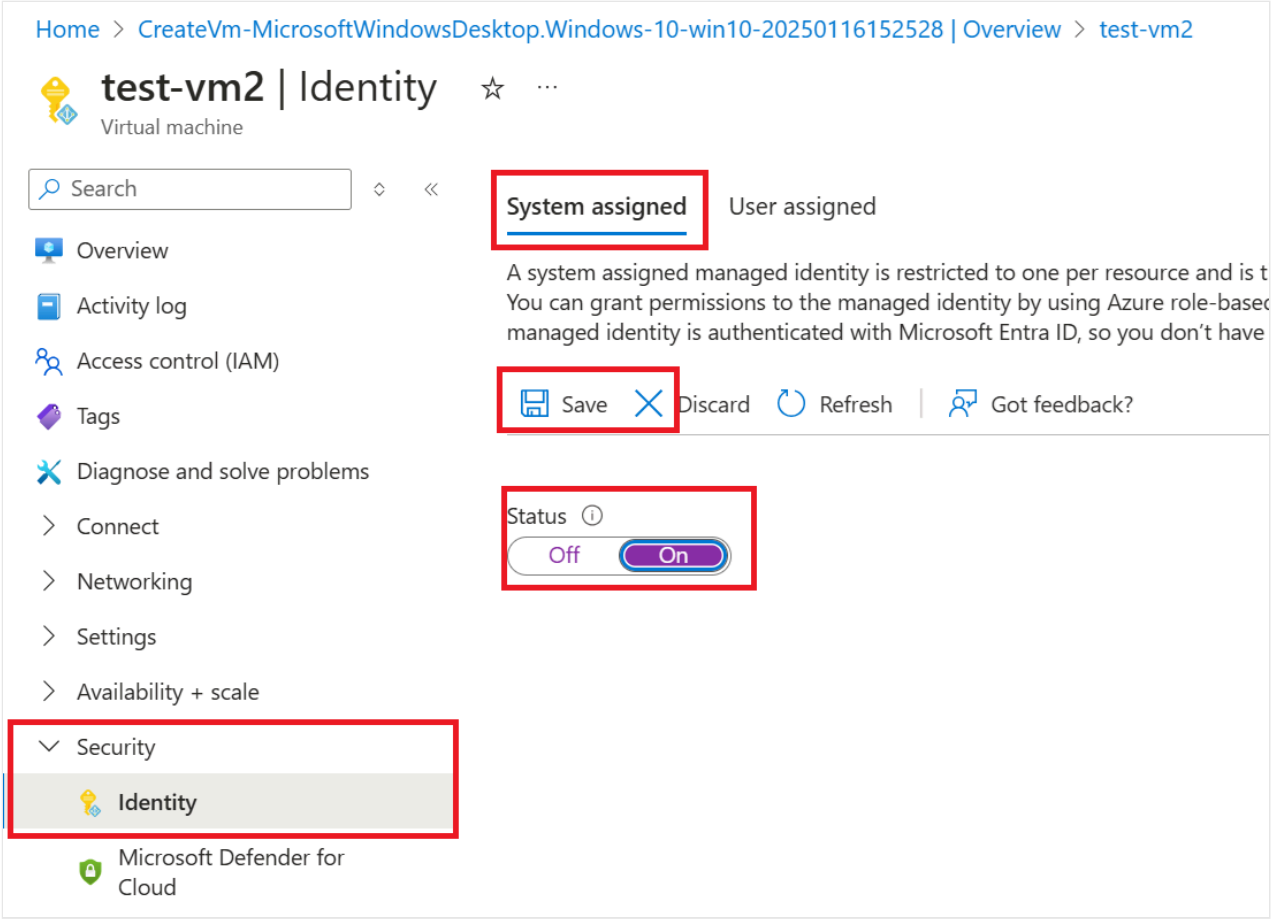
In the **Identity** section, select the **Enable system assigned managed identity** check-box.



Enable system-assigned managed identity on an existing VM

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **On** and then click **Save**:

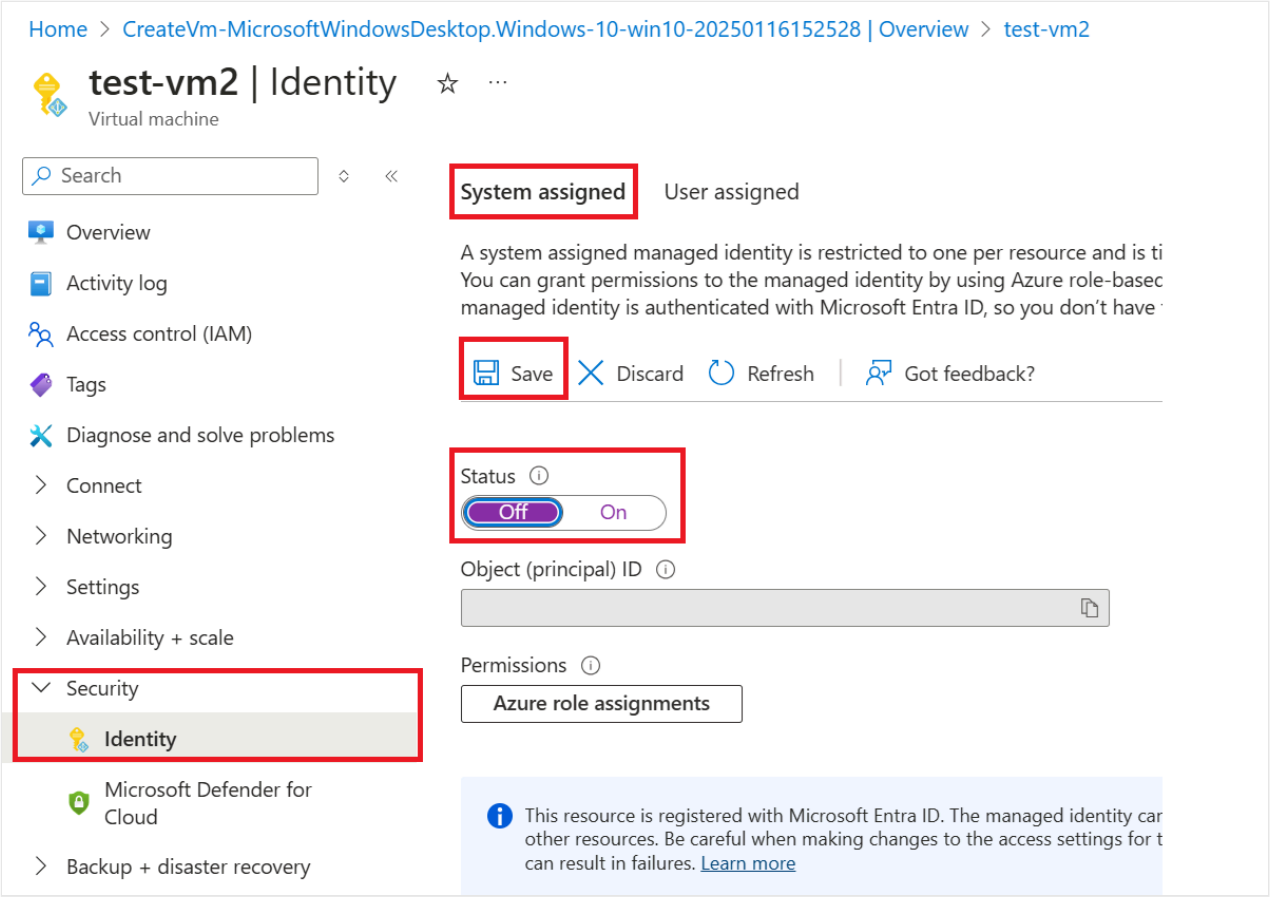


Remove system-assigned managed identity from a VM

To remove system-assigned managed identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

If you have a Virtual Machine that no longer needs system-assigned managed identity:

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **Off** and then click **Save**:



User-assigned managed identity

In this section, you learn how to add and remove a user-assigned managed identity from a VM using the Azure portal.

Assign a user-assigned identity during the creation of a VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

Currently, the Azure portal does not support assigning a user-assigned managed identity during the creation of a VM. First create a Windows virtual machine or Linux virtual machine, then assign a user-assigned managed identity to the VM.

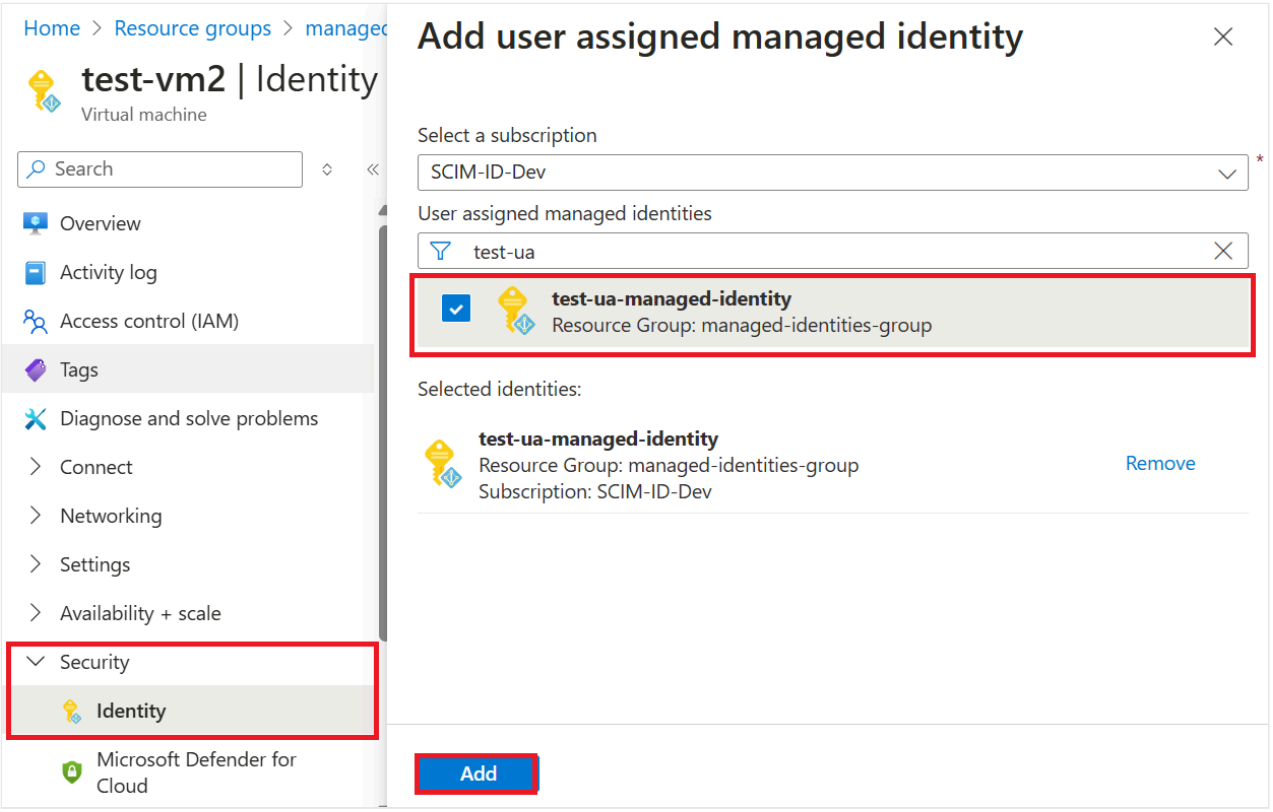
Assign a user-assigned managed identity to an existing VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

2. Navigate to the desired VM and click **Security** > **Identity**, **User assigned** and then **+Add**. Click the user-assigned identity you want to add to the VM and then click **Add**.

3. Select the previously created user assigned managed identity from the list.

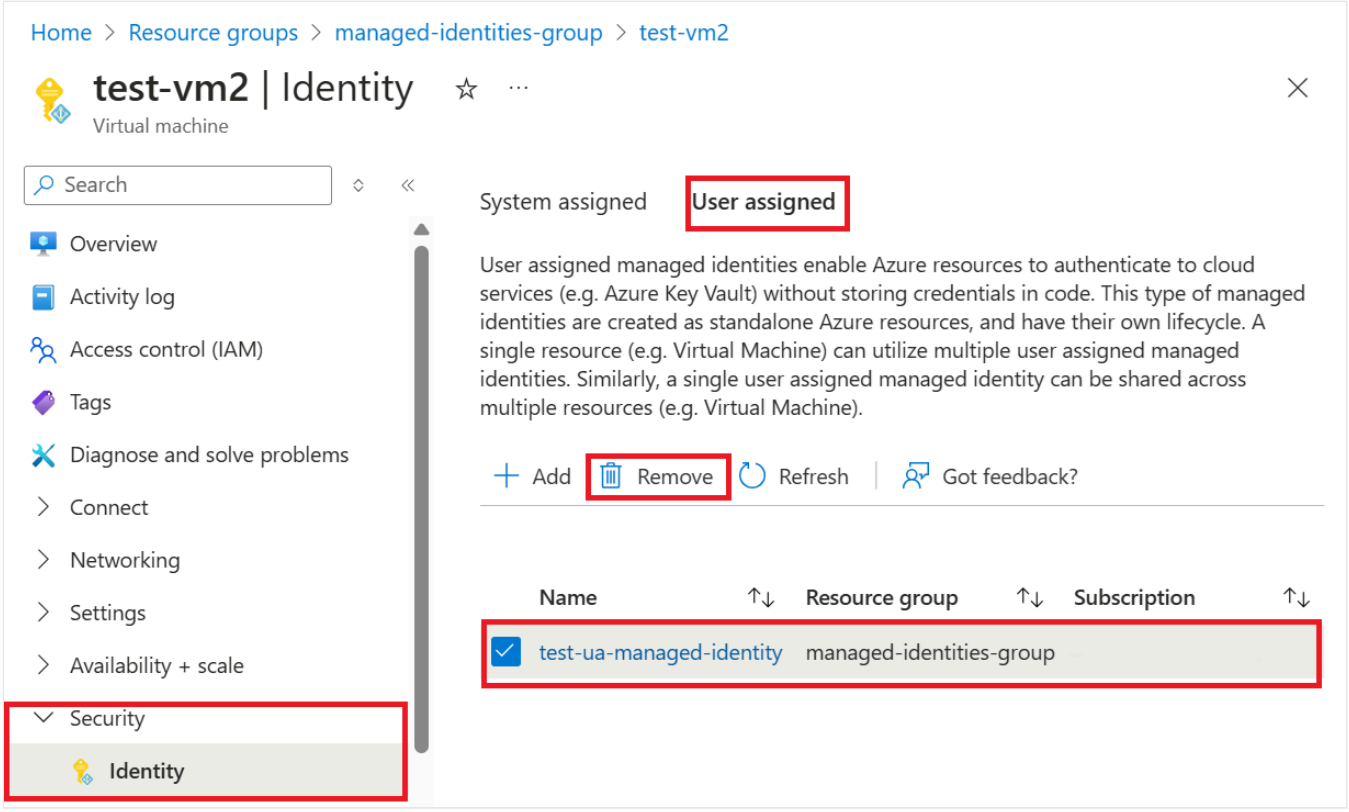


Remove a user-assigned managed identity from a VM

To remove a user-assigned identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

Navigate to the desired VM and select **Security** > **Identity**, **User assigned**, the name of the user-assigned managed identity you want to delete and then click **Remove** (click **Yes** in the confirmation pane).



Next steps

- Using the Azure portal, give an Azure VM's managed identity access to another Azure resource.

Last updated on 01/16/2025

Configure managed identities on Azure virtual machines (VMs)

Managed identities for Azure resources provide Azure services with an automatically managed identity in Microsoft Entra ID. You can use this identity to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

For information about Azure Policy definition and details, see Use Azure Policy to assign managed identities (preview).

In this article, you learn how to enable and disable system and user-assigned managed identities for an Azure Virtual Machine (VM), using the Azure portal.

Prerequisites

- If you're unfamiliar with managed identities for Azure resources, check out the overview section.
- If you don't already have an Azure account, sign up for a free account before continuing.

System-assigned managed identity

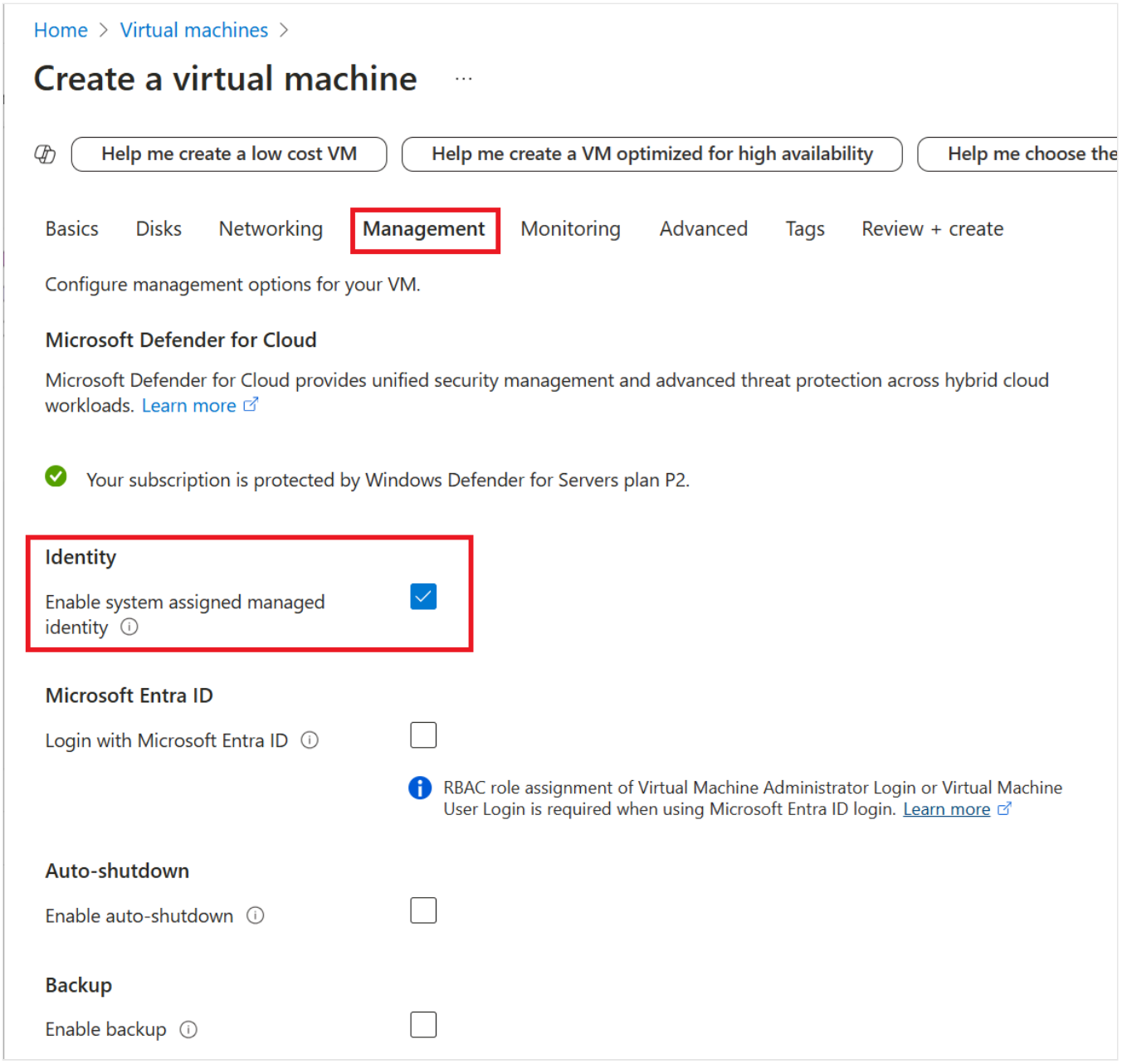
In this section, you learn how to enable and disable the system-assigned managed identity for VM using the Azure portal.

Enable system-assigned managed identity during creation of a VM

To enable system-assigned managed identity on a VM during its creation, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

When creating a Windows virtual machine or Linux virtual machine, select the **Management** tab.

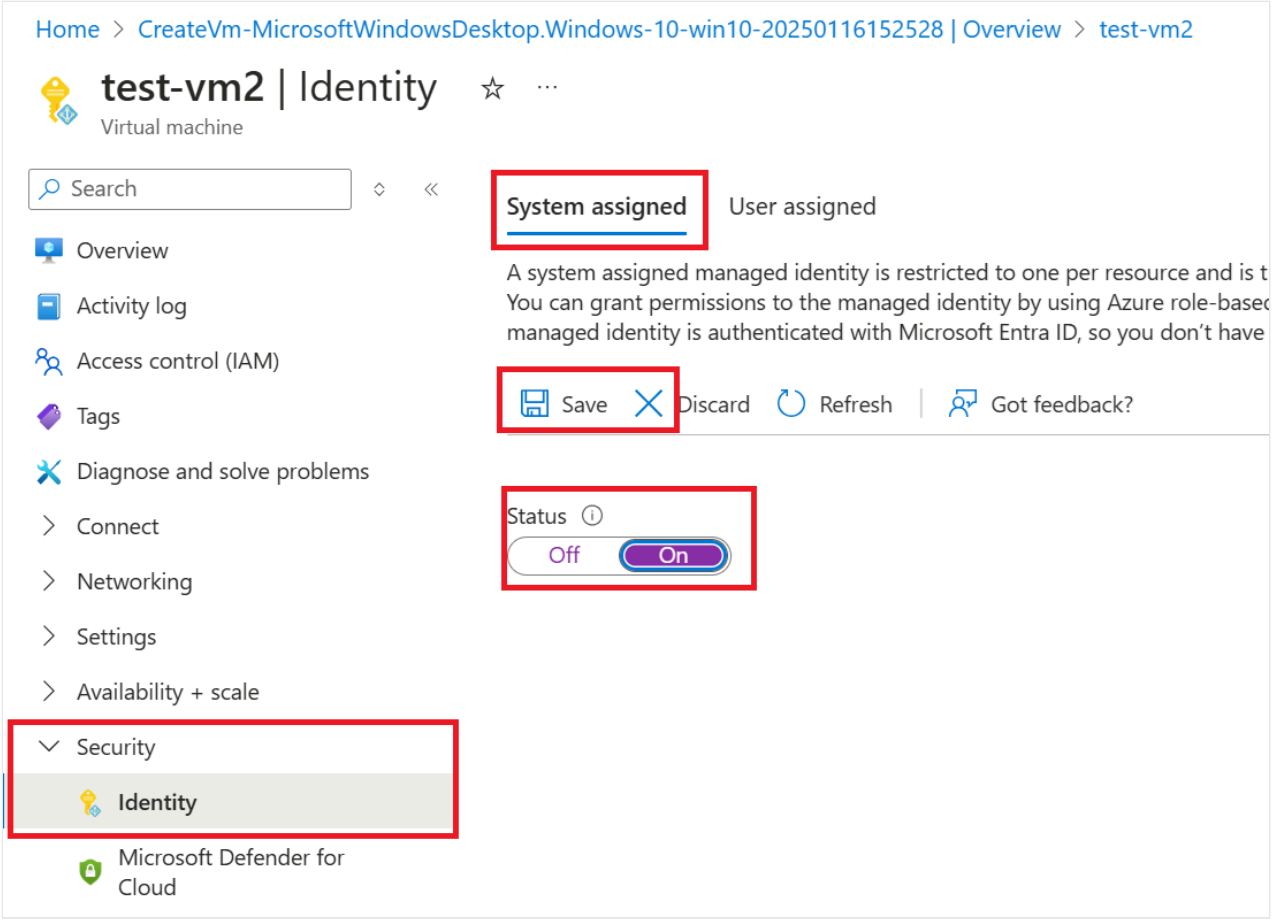
In the **Identity** section, select the **Enable system assigned managed identity** check-box.



Enable system-assigned managed identity on an existing VM

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **On** and then click **Save**:

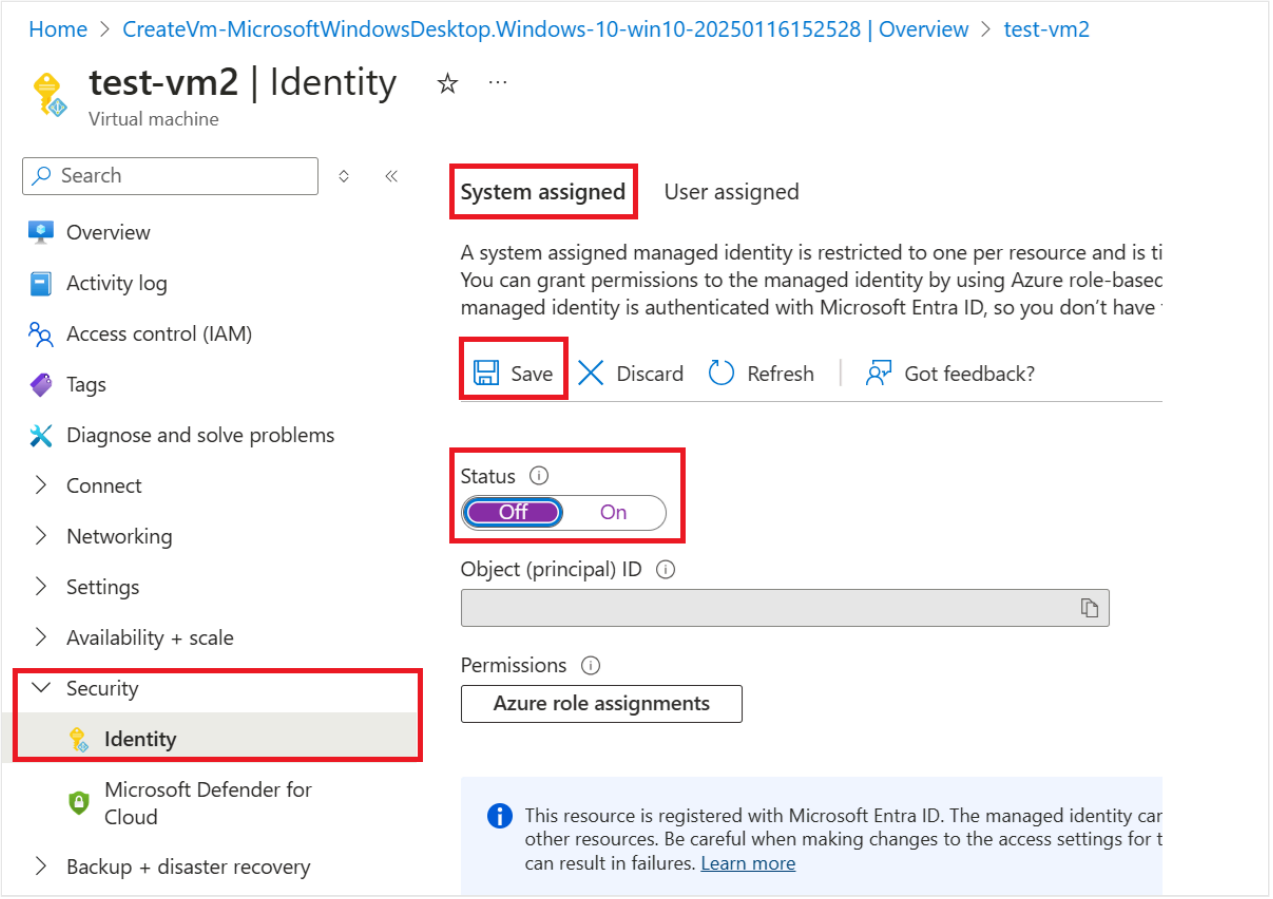


Remove system-assigned managed identity from a VM

To remove system-assigned managed identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

If you have a Virtual Machine that no longer needs system-assigned managed identity:

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **Off** and then click **Save**:



User-assigned managed identity

In this section, you learn how to add and remove a user-assigned managed identity from a VM using the Azure portal.

Assign a user-assigned identity during the creation of a VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

Currently, the Azure portal does not support assigning a user-assigned managed identity during the creation of a VM. First create a Windows virtual machine or Linux virtual machine, then assign a user-assigned managed identity to the VM.

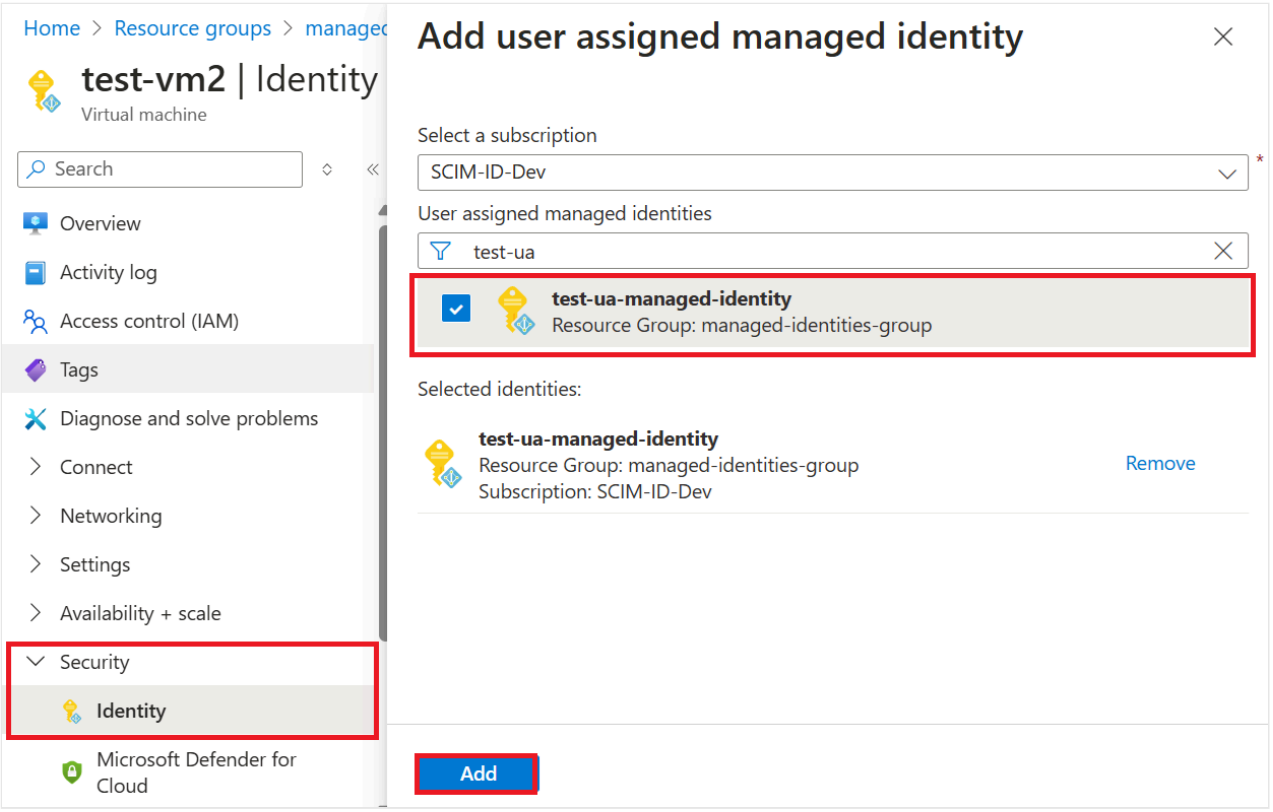
Assign a user-assigned managed identity to an existing VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

2. Navigate to the desired VM and click **Security** > **Identity**, **User assigned** and then **+Add**. Click the user-assigned identity you want to add to the VM and then click **Add**.

3. Select the previously created user assigned managed identity from the list.

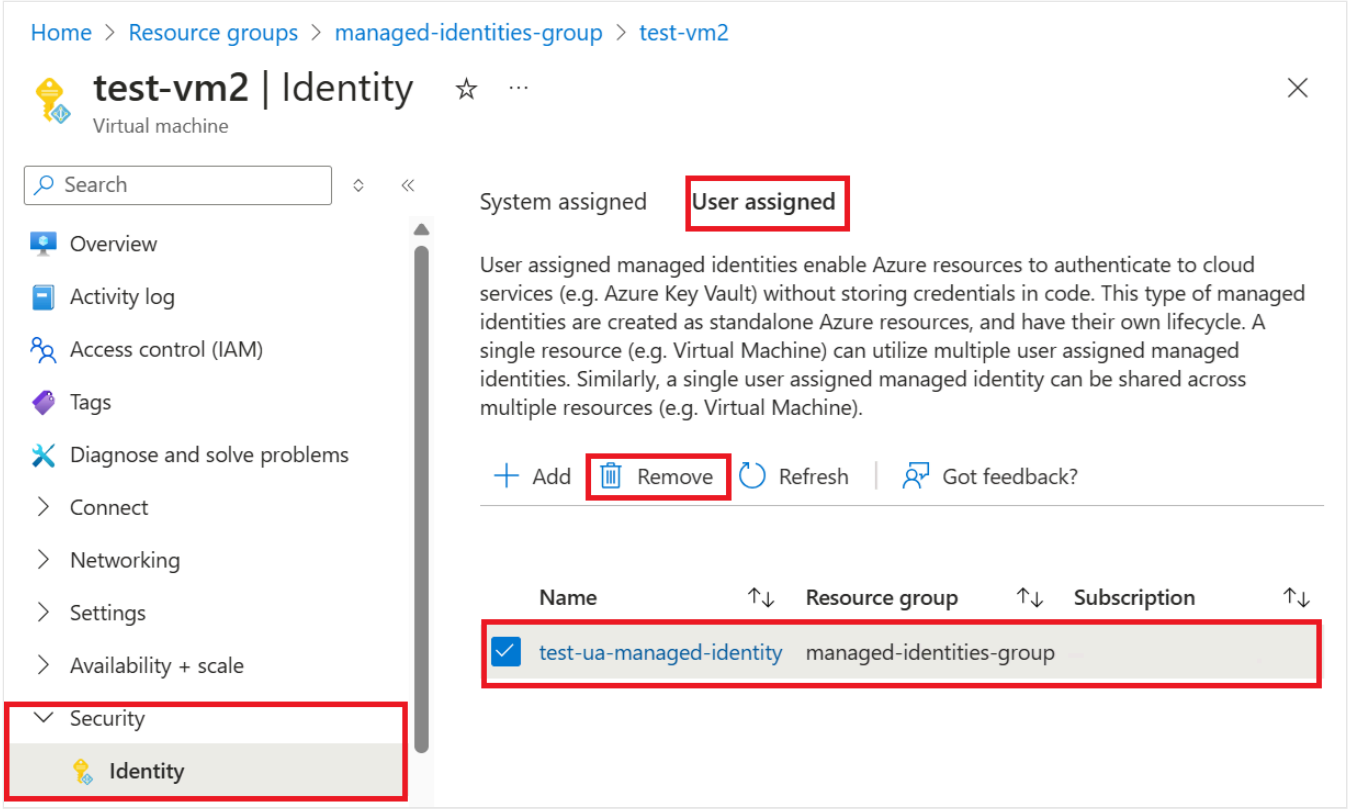


Remove a user-assigned managed identity from a VM

To remove a user-assigned identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

Navigate to the desired VM and select **Security** > **Identity**, **User assigned**, the name of the user-assigned managed identity you want to delete and then click **Remove** (click **Yes** in the confirmation pane).



Next steps

- Using the Azure portal, give an Azure VM's managed identity access to another Azure resource.

Last updated on 01/16/2025

Configure managed identities on Azure virtual machines (VMs)

Managed identities for Azure resources provide Azure services with an automatically managed identity in Microsoft Entra ID. You can use this identity to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

For information about Azure Policy definition and details, see Use Azure Policy to assign managed identities (preview).

In this article, you learn how to enable and disable system and user-assigned managed identities for an Azure Virtual Machine (VM), using the Azure portal.

Prerequisites

- If you're unfamiliar with managed identities for Azure resources, check out the overview section.
- If you don't already have an Azure account, sign up for a free account before continuing.

System-assigned managed identity

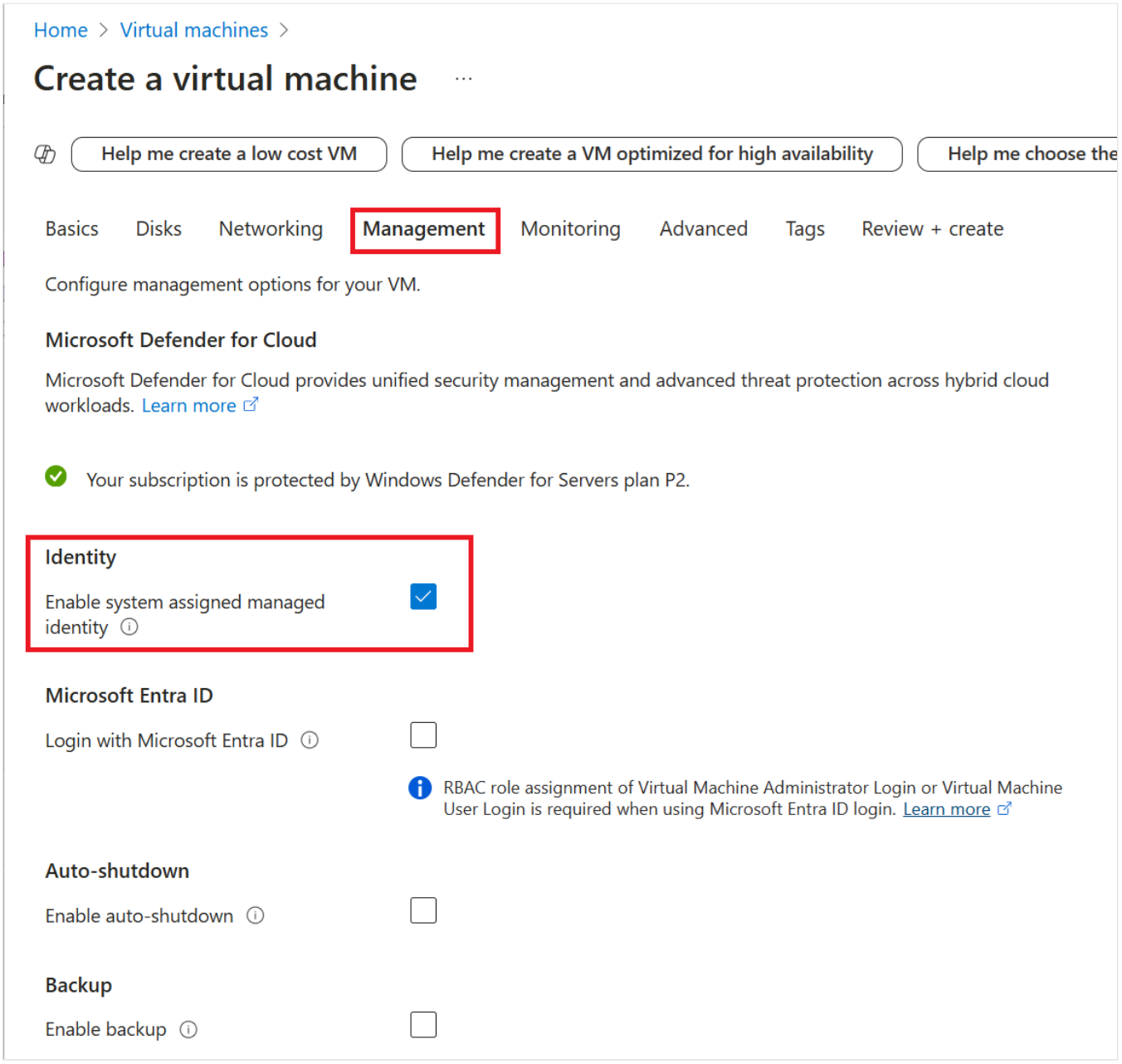
In this section, you learn how to enable and disable the system-assigned managed identity for VM using the Azure portal.

Enable system-assigned managed identity during creation of a VM

To enable system-assigned managed identity on a VM during its creation, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

When creating a Windows virtual machine or Linux virtual machine, select the **Management** tab.

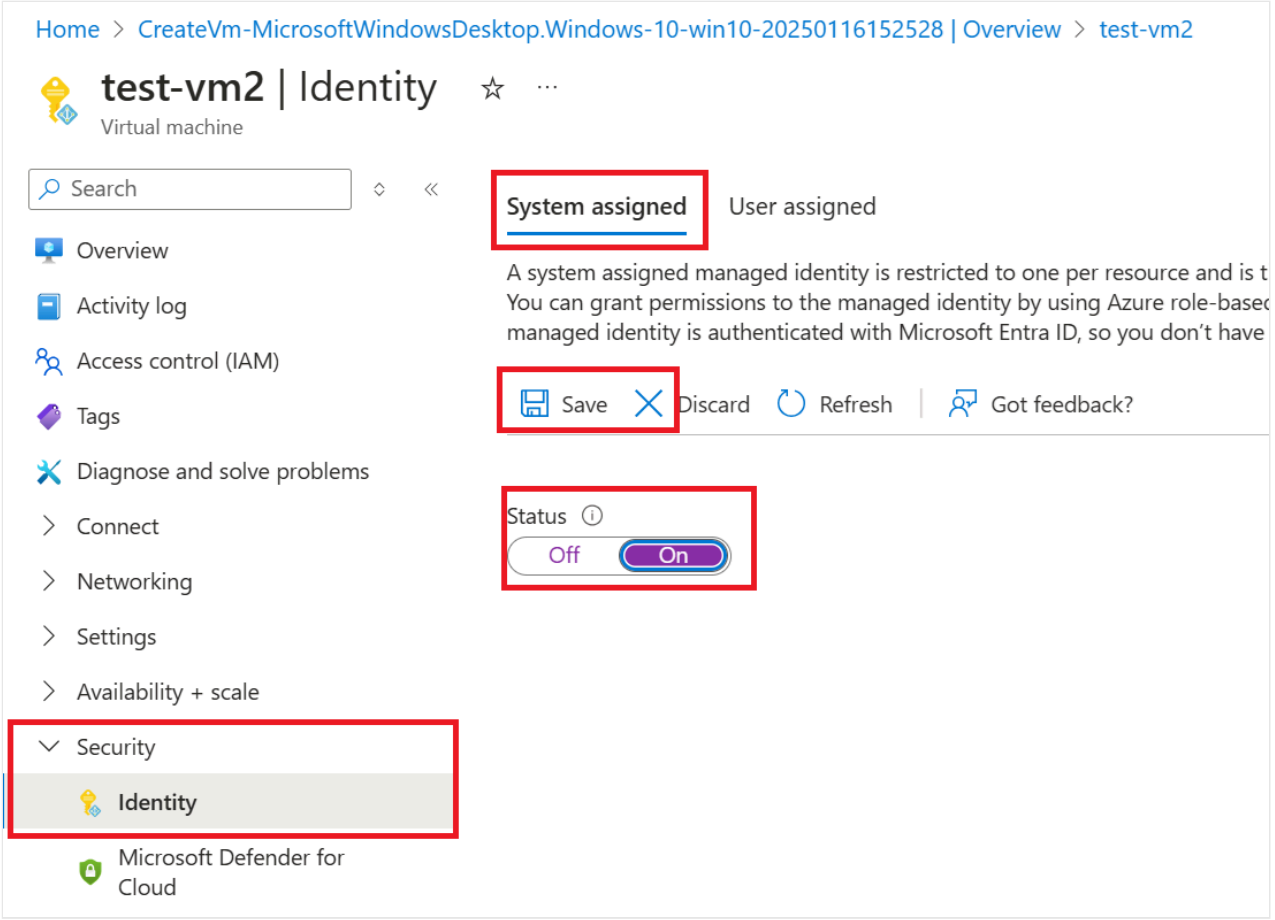
In the **Identity** section, select the **Enable system assigned managed identity** check-box.



Enable system-assigned managed identity on an existing VM

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **On** and then click **Save**:

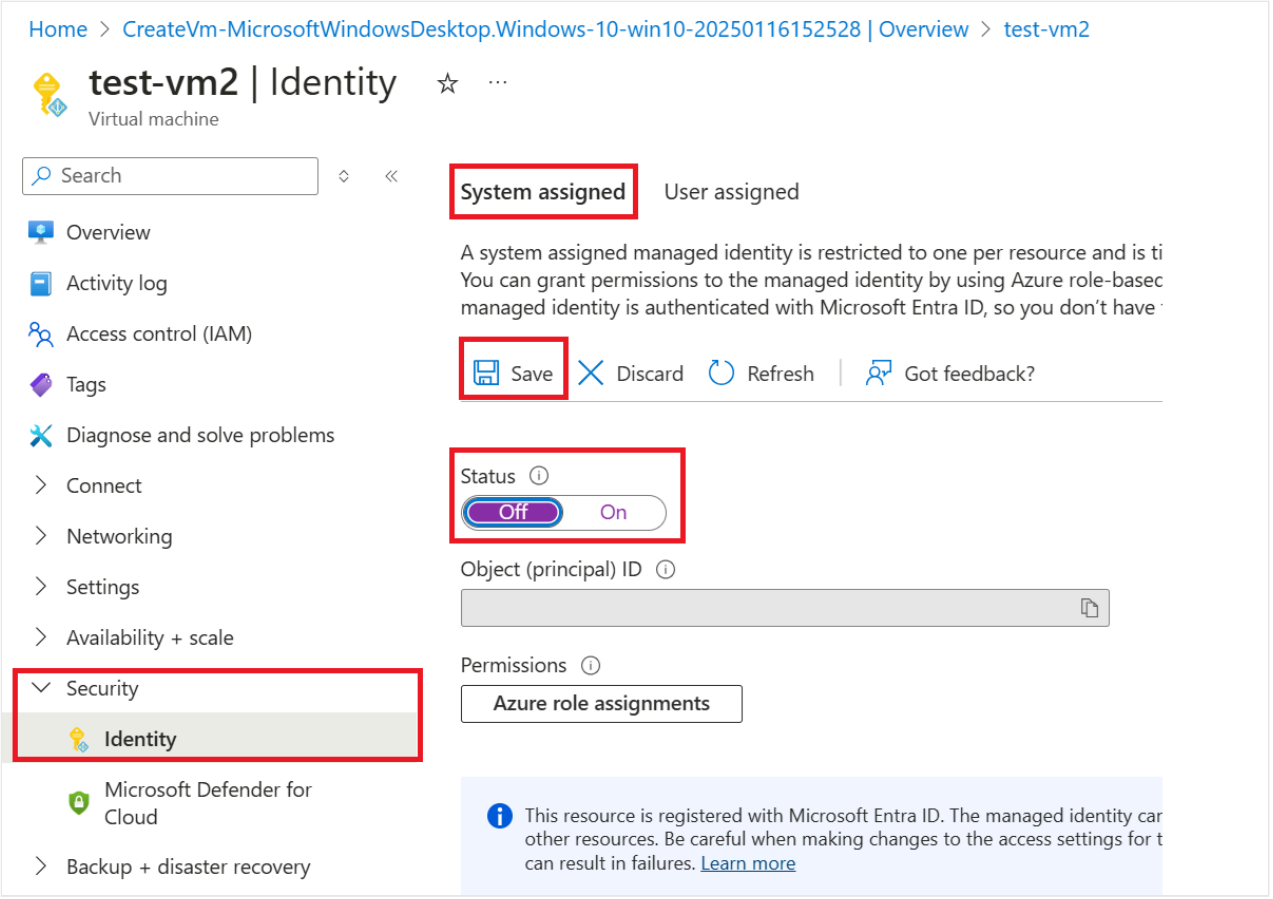


Remove system-assigned managed identity from a VM

To remove system-assigned managed identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

If you have a Virtual Machine that no longer needs system-assigned managed identity:

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **Off** and then click **Save**:



User-assigned managed identity

In this section, you learn how to add and remove a user-assigned managed identity from a VM using the Azure portal.

Assign a user-assigned identity during the creation of a VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

Currently, the Azure portal does not support assigning a user-assigned managed identity during the creation of a VM. First create a Windows virtual machine or Linux virtual machine, then assign a user-assigned managed identity to the VM.

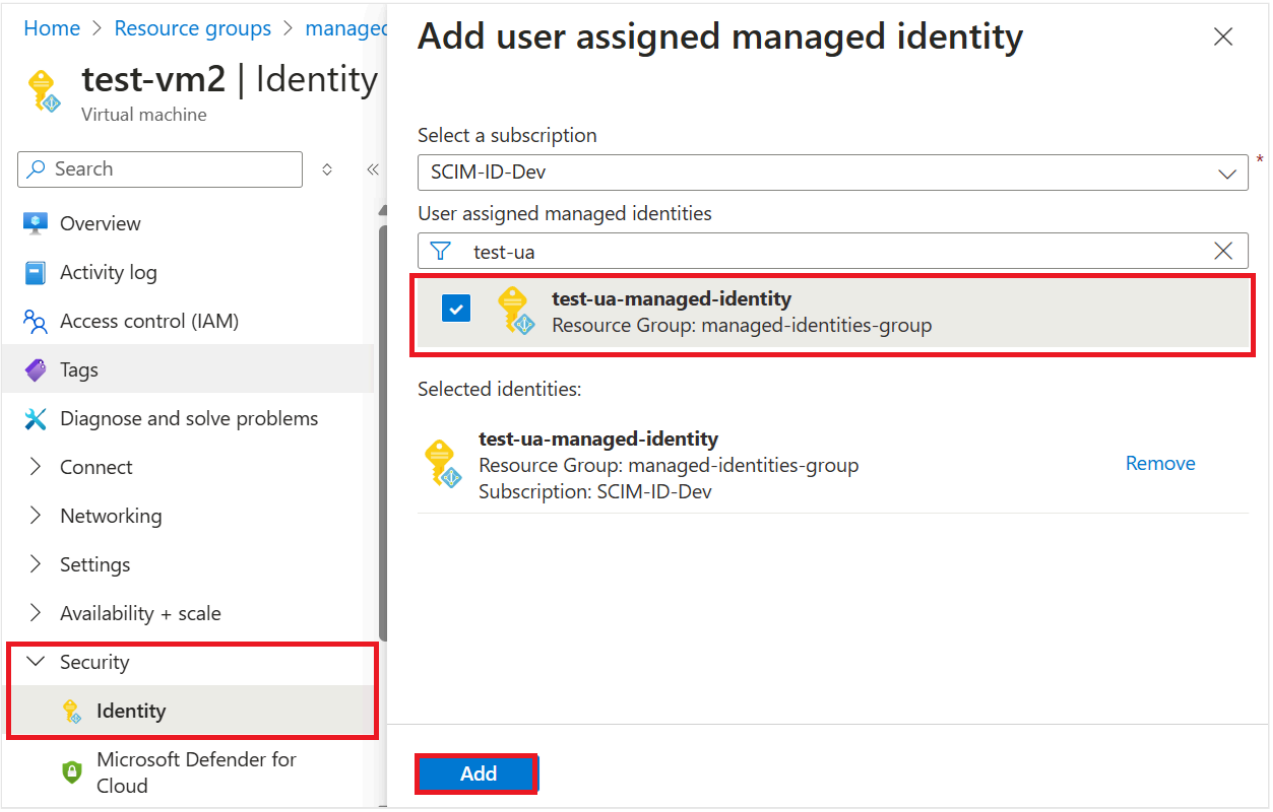
Assign a user-assigned managed identity to an existing VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

2. Navigate to the desired VM and click **Security** > **Identity**, **User assigned** and then **+Add**. Click the user-assigned identity you want to add to the VM and then click **Add**.

3. Select the previously created user assigned managed identity from the list.

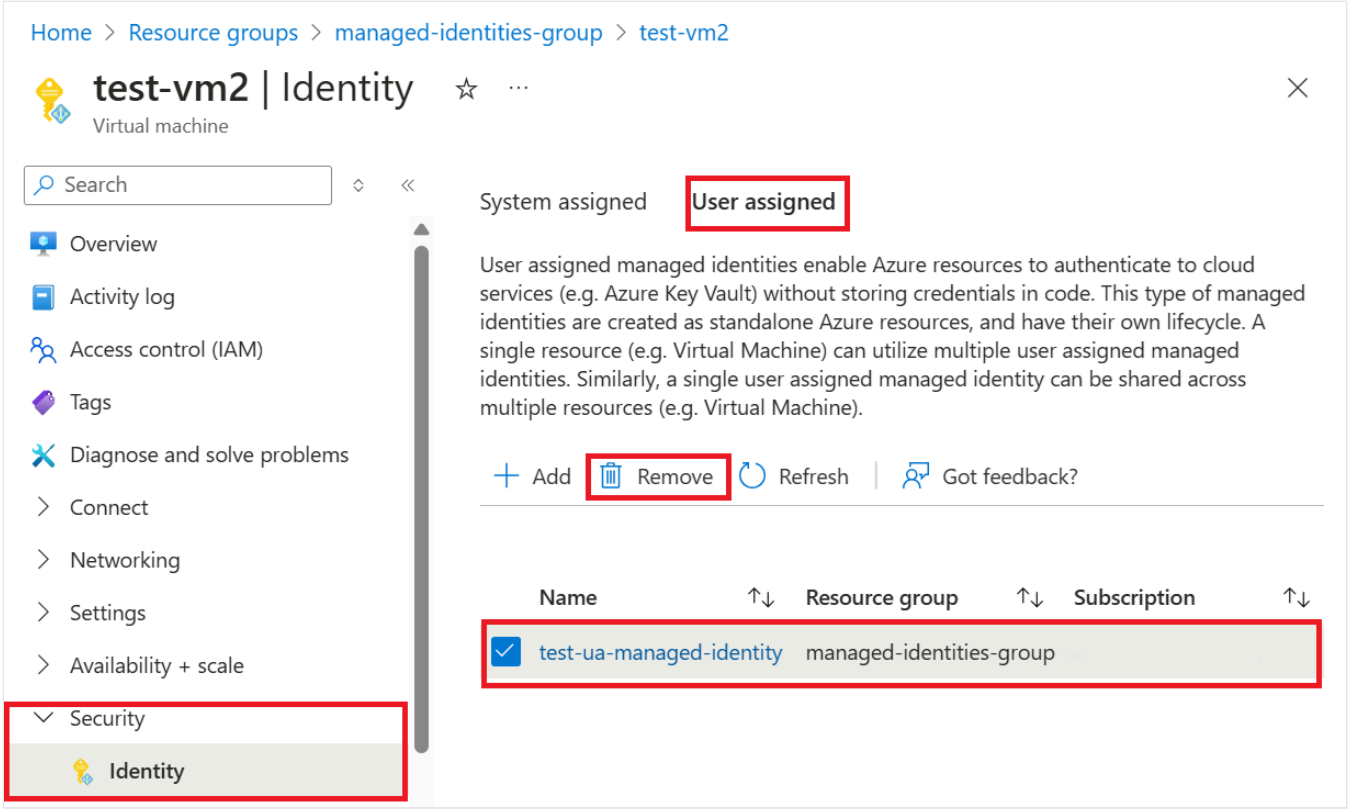


Remove a user-assigned managed identity from a VM

To remove a user-assigned identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

Navigate to the desired VM and select **Security** > **Identity**, **User assigned**, the name of the user-assigned managed identity you want to delete and then click **Remove** (click **Yes** in the confirmation pane).



Next steps

- Using the Azure portal, give an Azure VM's managed identity access to another Azure resource.

Last updated on 01/16/2025

Configure managed identities on Azure virtual machines (VMs)

Managed identities for Azure resources provide Azure services with an automatically managed identity in Microsoft Entra ID. You can use this identity to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

For information about Azure Policy definition and details, see Use Azure Policy to assign managed identities (preview).

In this article, you learn how to enable and disable system and user-assigned managed identities for an Azure Virtual Machine (VM), using the Azure portal.

Prerequisites

- If you're unfamiliar with managed identities for Azure resources, check out the overview section.
- If you don't already have an Azure account, sign up for a free account before continuing.

System-assigned managed identity

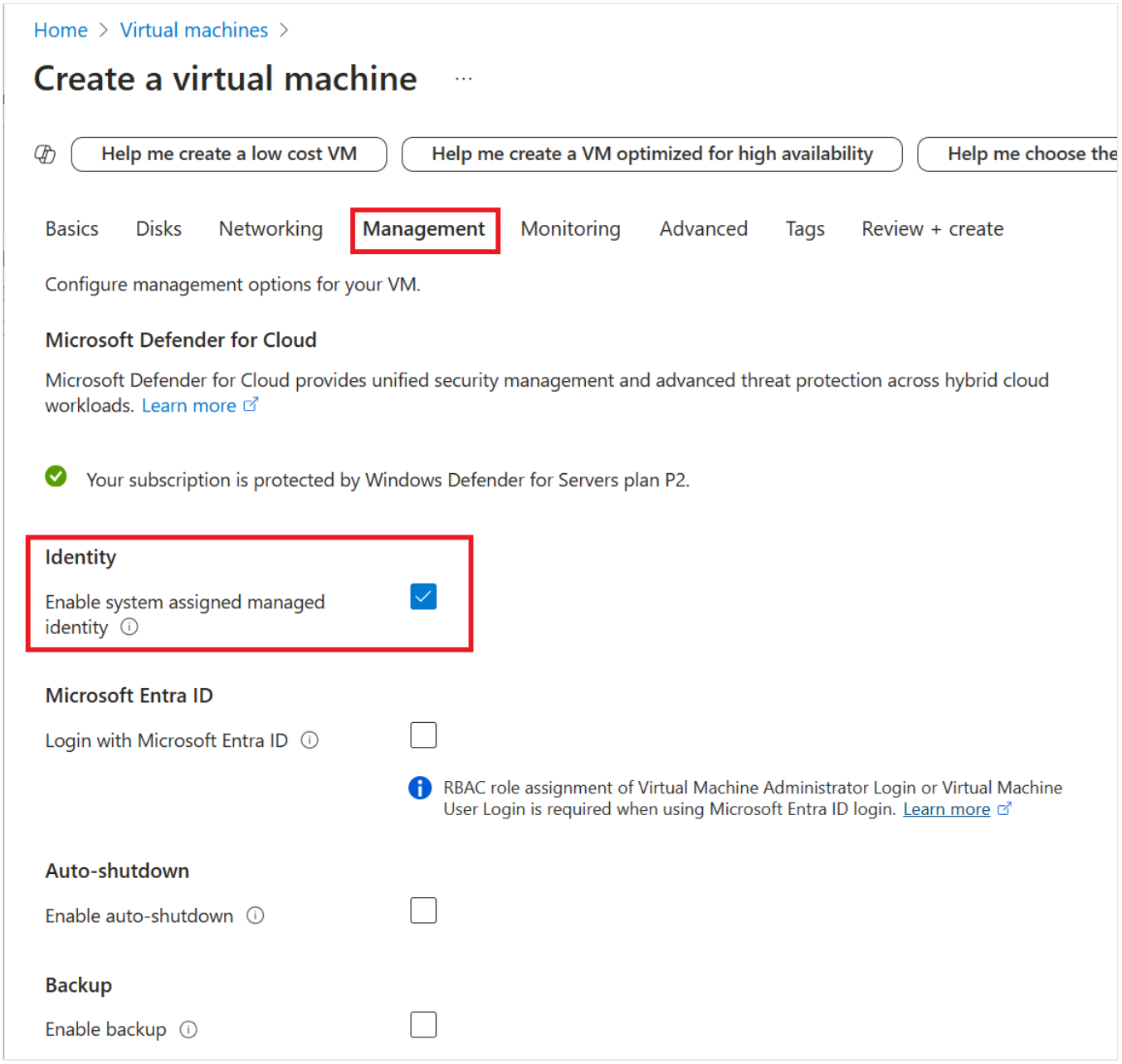
In this section, you learn how to enable and disable the system-assigned managed identity for VM using the Azure portal.

Enable system-assigned managed identity during creation of a VM

To enable system-assigned managed identity on a VM during its creation, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

When creating a Windows virtual machine or Linux virtual machine, select the **Management** tab.

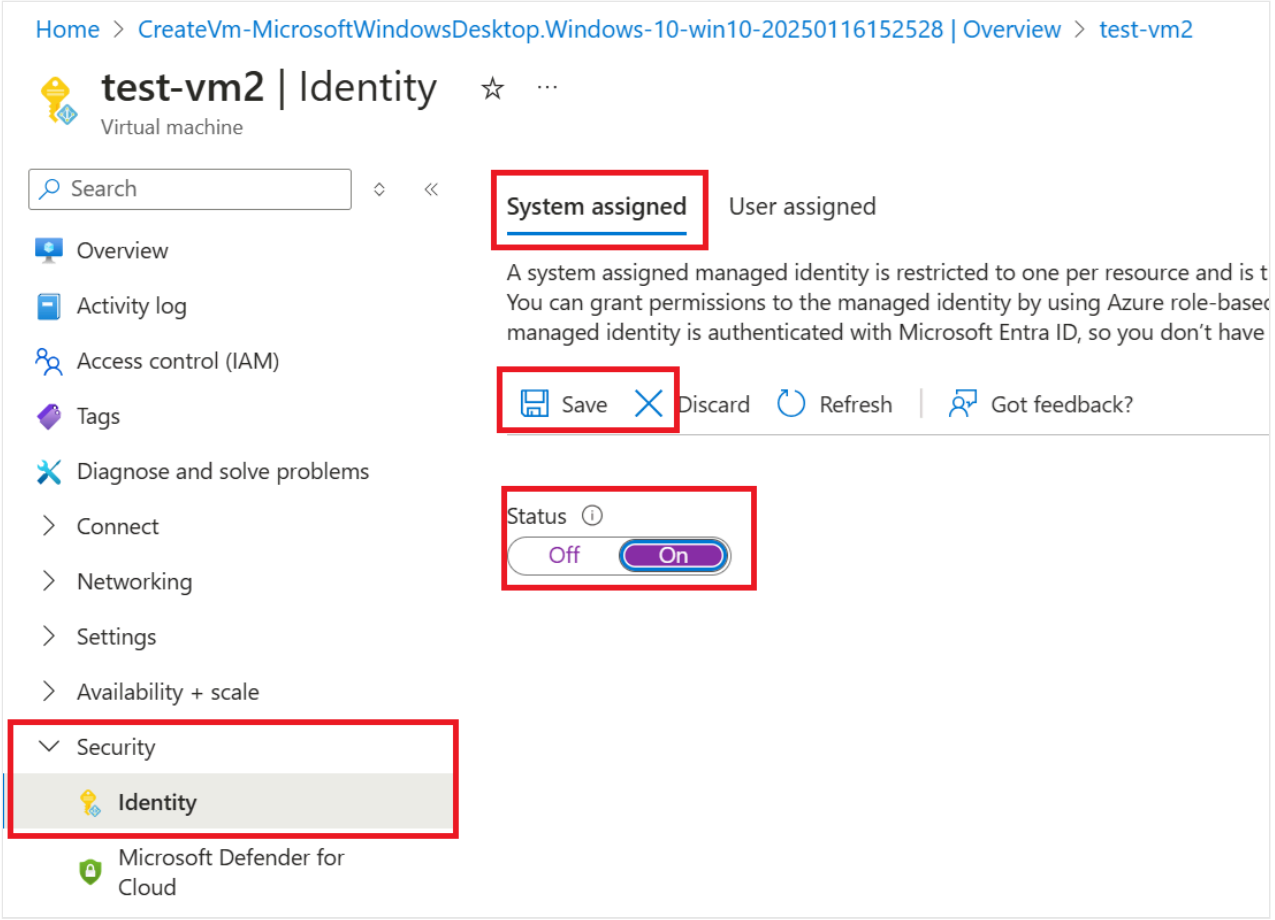
In the **Identity** section, select the **Enable system assigned managed identity** check-box.



Enable system-assigned managed identity on an existing VM

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **On** and then click **Save**:

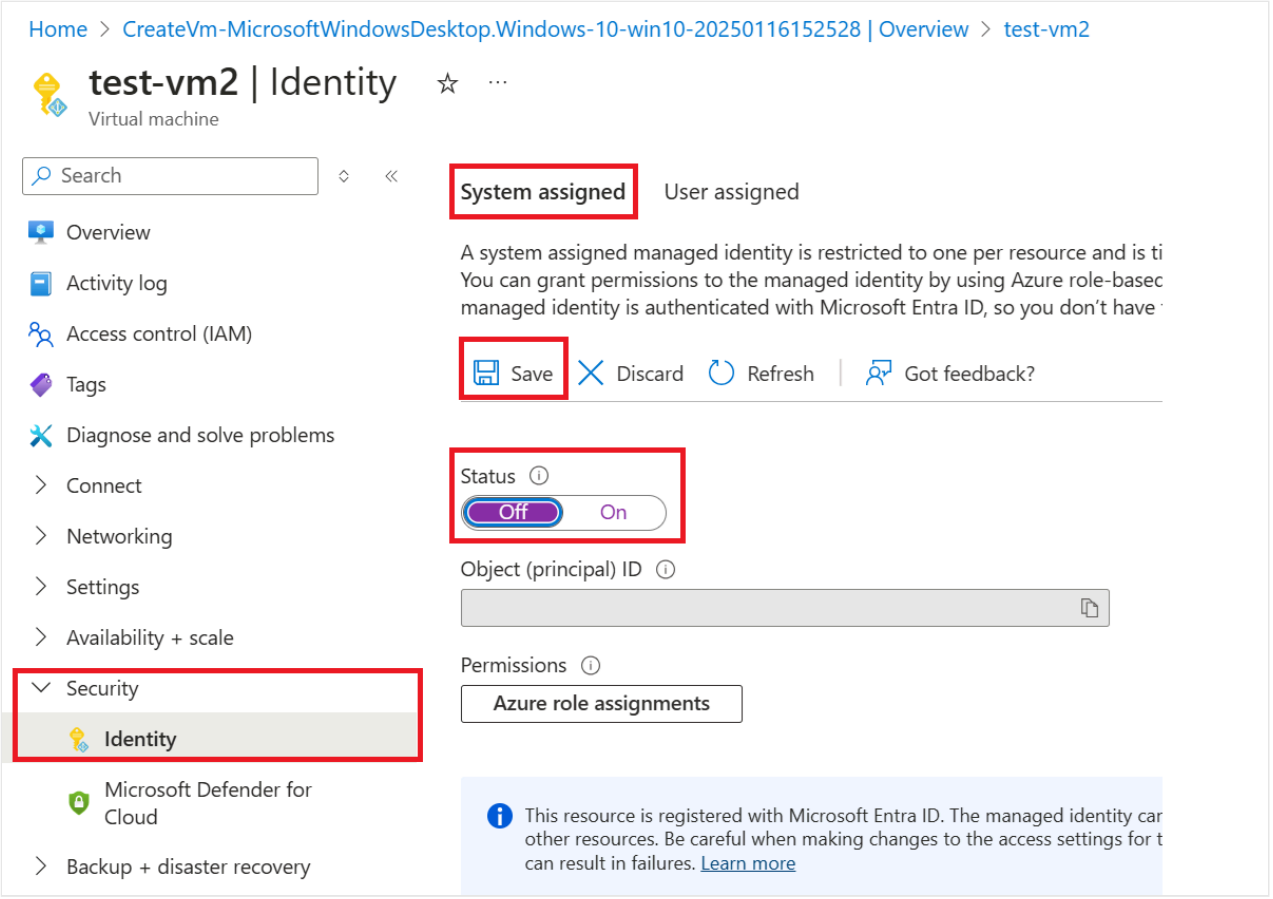


Remove system-assigned managed identity from a VM

To remove system-assigned managed identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

If you have a Virtual Machine that no longer needs system-assigned managed identity:

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **Off** and then click **Save**:



User-assigned managed identity

In this section, you learn how to add and remove a user-assigned managed identity from a VM using the Azure portal.

Assign a user-assigned identity during the creation of a VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

Currently, the Azure portal does not support assigning a user-assigned managed identity during the creation of a VM. First create a Windows virtual machine or Linux virtual machine, then assign a user-assigned managed identity to the VM.

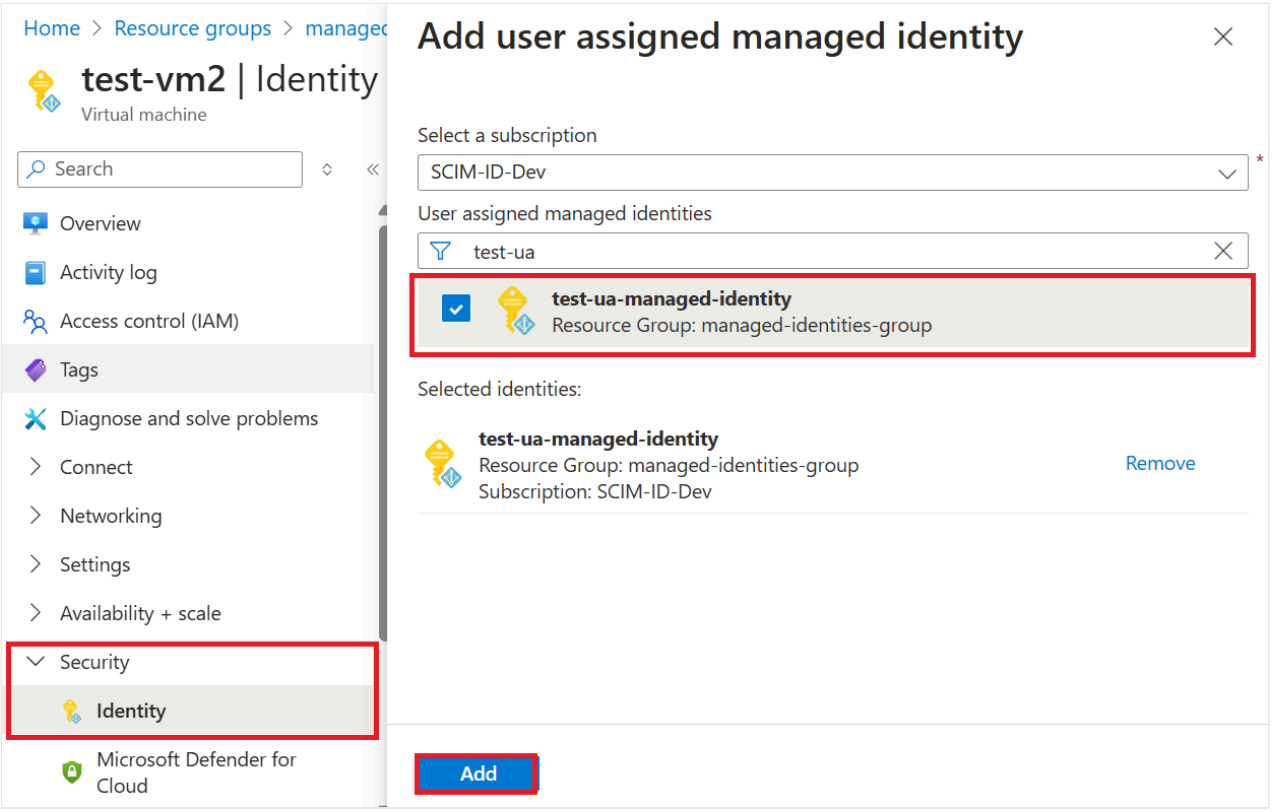
Assign a user-assigned managed identity to an existing VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

2. Navigate to the desired VM and click **Security** > **Identity**, **User assigned** and then **+Add**. Click the user-assigned identity you want to add to the VM and then click **Add**.

3. Select the previously created user assigned managed identity from the list.

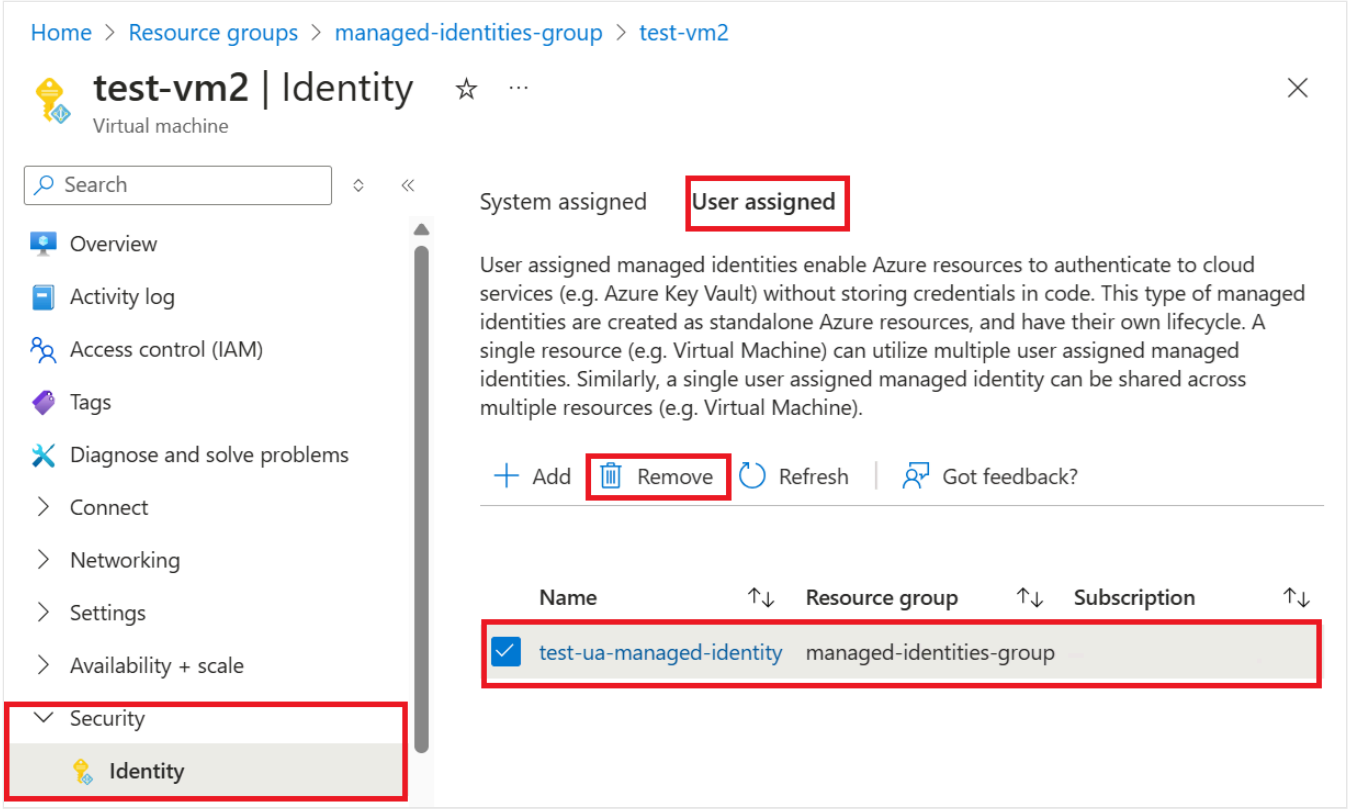


Remove a user-assigned managed identity from a VM

To remove a user-assigned identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

Navigate to the desired VM and select **Security** > **Identity**, **User assigned**, the name of the user-assigned managed identity you want to delete and then click **Remove** (click **Yes** in the confirmation pane).



Next steps

- Using the Azure portal, give an Azure VM's managed identity access to another Azure resource.

Last updated on 01/16/2025

Configure managed identities on Azure virtual machines (VMs)

Managed identities for Azure resources provide Azure services with an automatically managed identity in Microsoft Entra ID. You can use this identity to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

For information about Azure Policy definition and details, see Use Azure Policy to assign managed identities (preview).

In this article, you learn how to enable and disable system and user-assigned managed identities for an Azure Virtual Machine (VM), using the Azure portal.

Prerequisites

- If you're unfamiliar with managed identities for Azure resources, check out the overview section.
- If you don't already have an Azure account, sign up for a free account before continuing.

System-assigned managed identity

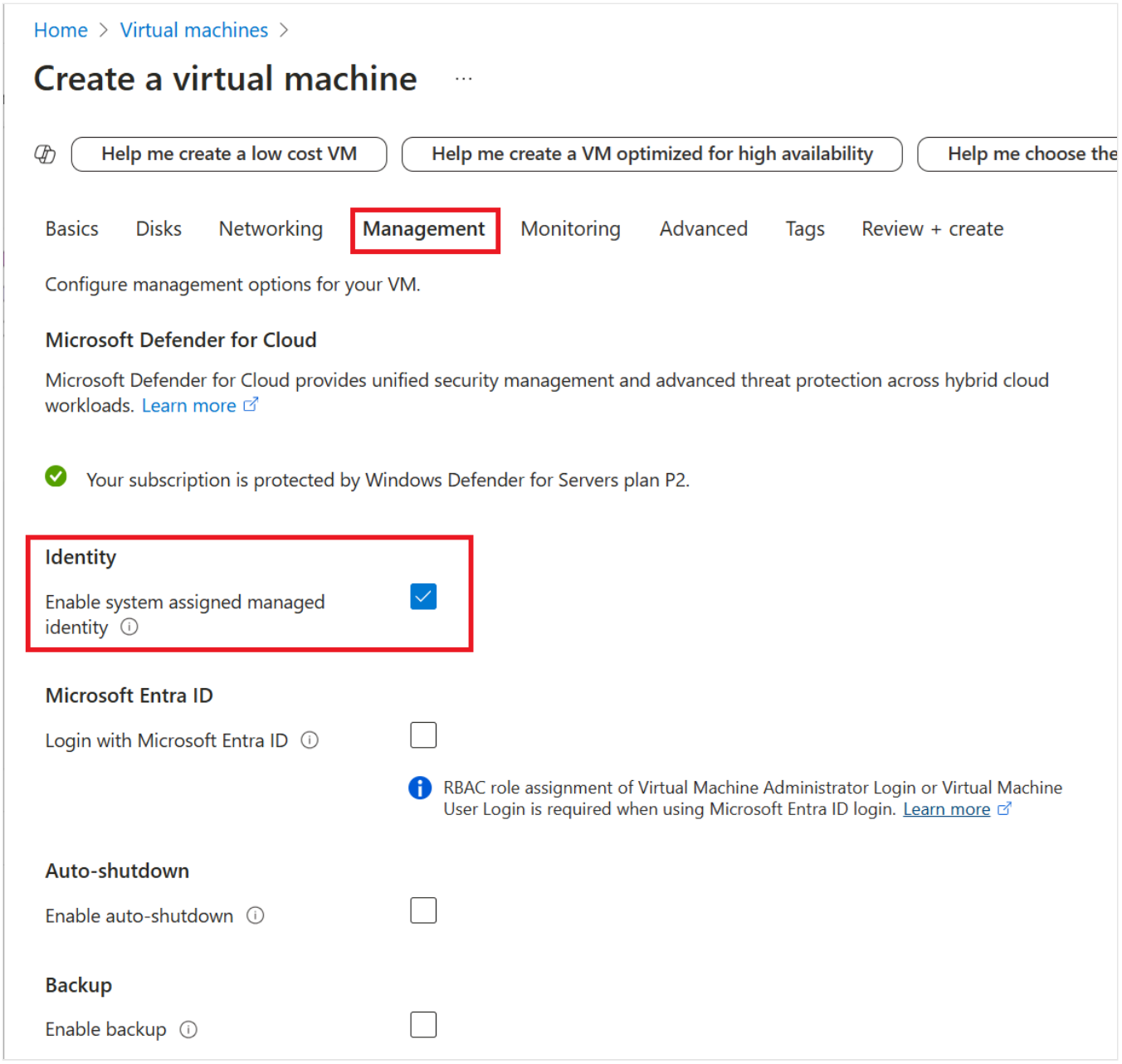
In this section, you learn how to enable and disable the system-assigned managed identity for VM using the Azure portal.

Enable system-assigned managed identity during creation of a VM

To enable system-assigned managed identity on a VM during its creation, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

When creating a Windows virtual machine or Linux virtual machine, select the **Management** tab.

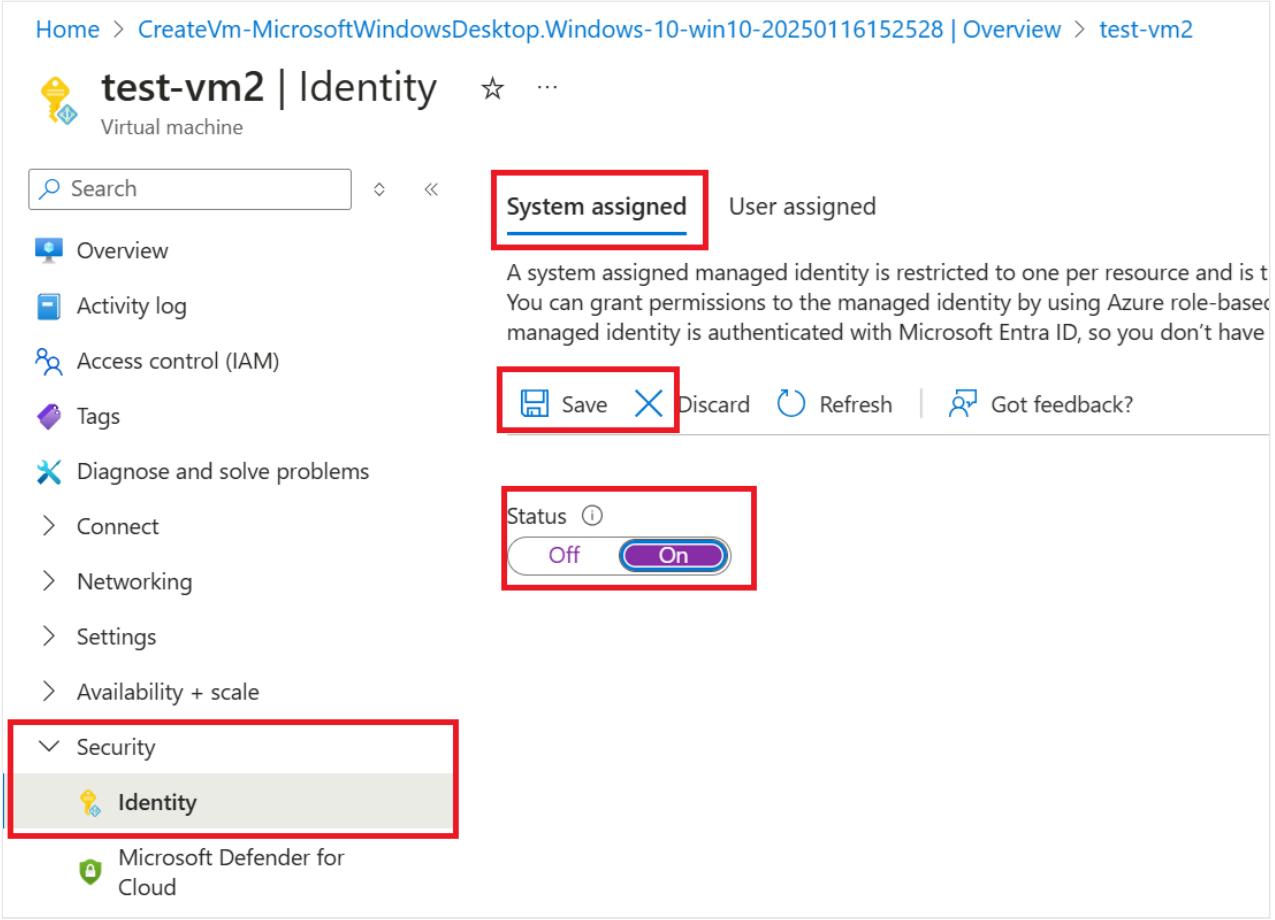
In the **Identity** section, select the **Enable system assigned managed identity** check-box.



Enable system-assigned managed identity on an existing VM

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **On** and then click **Save**:

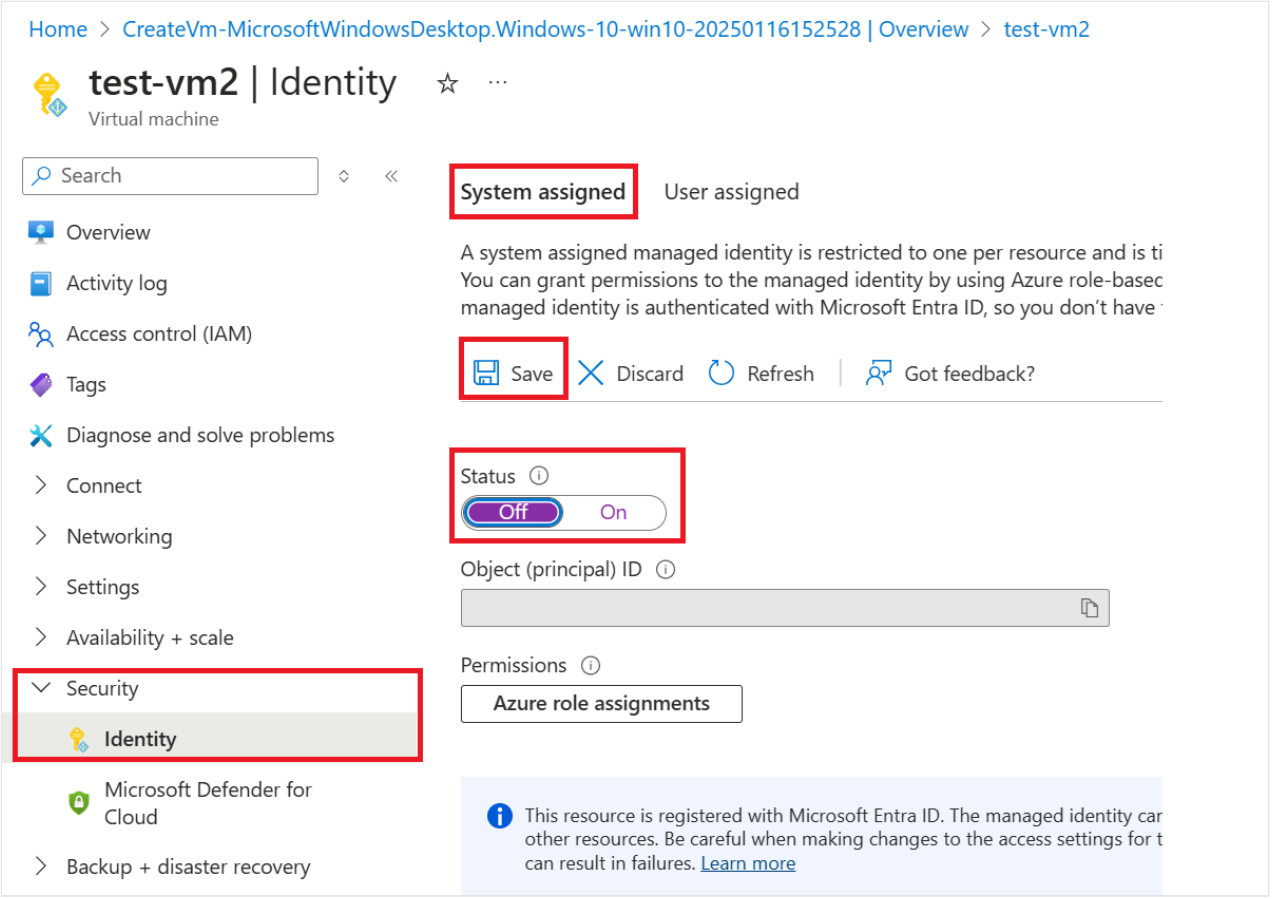


Remove system-assigned managed identity from a VM

To remove system-assigned managed identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

If you have a Virtual Machine that no longer needs system-assigned managed identity:

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **Off** and then click **Save**:



User-assigned managed identity

In this section, you learn how to add and remove a user-assigned managed identity from a VM using the Azure portal.

Assign a user-assigned identity during the creation of a VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

Currently, the Azure portal does not support assigning a user-assigned managed identity during the creation of a VM. First create a Windows virtual machine or Linux virtual machine, then assign a user-assigned managed identity to the VM.

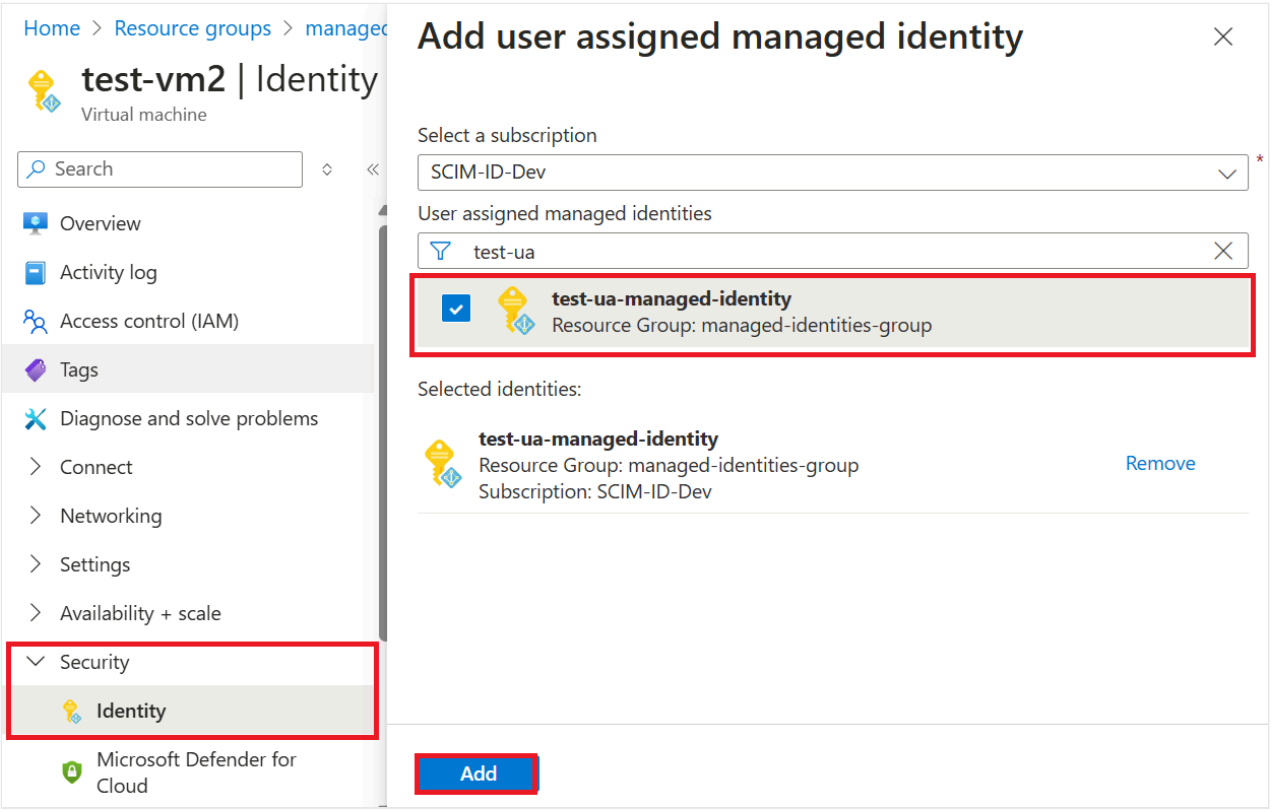
Assign a user-assigned managed identity to an existing VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

2. Navigate to the desired VM and click **Security** > **Identity**, **User assigned** and then **+Add**. Click the user-assigned identity you want to add to the VM and then click **Add**.

3. Select the previously created user assigned managed identity from the list.

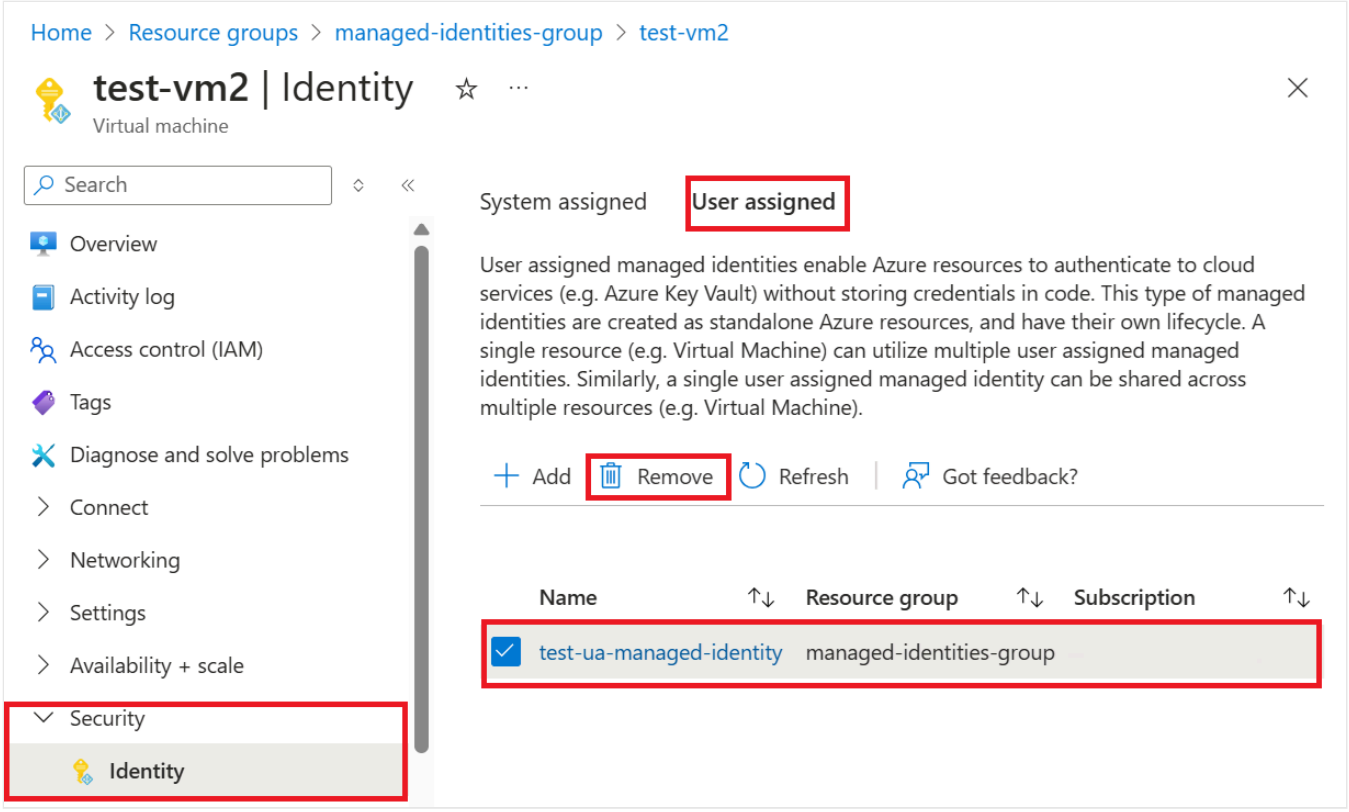


Remove a user-assigned managed identity from a VM

To remove a user-assigned identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

Navigate to the desired VM and select **Security** > **Identity**, **User assigned**, the name of the user-assigned managed identity you want to delete and then click **Remove** (click **Yes** in the confirmation pane).



Next steps

- Using the Azure portal, give an Azure VM's managed identity access to another Azure resource.

Last updated on 01/16/2025

Configure managed identities on Azure virtual machines (VMs)

Managed identities for Azure resources provide Azure services with an automatically managed identity in Microsoft Entra ID. You can use this identity to authenticate to any service that supports Microsoft Entra authentication, without having credentials in your code.

For information about Azure Policy definition and details, see Use Azure Policy to assign managed identities (preview).

In this article, you learn how to enable and disable system and user-assigned managed identities for an Azure Virtual Machine (VM), using the Azure portal.

Prerequisites

- If you're unfamiliar with managed identities for Azure resources, check out the overview section.
- If you don't already have an Azure account, sign up for a free account before continuing.

System-assigned managed identity

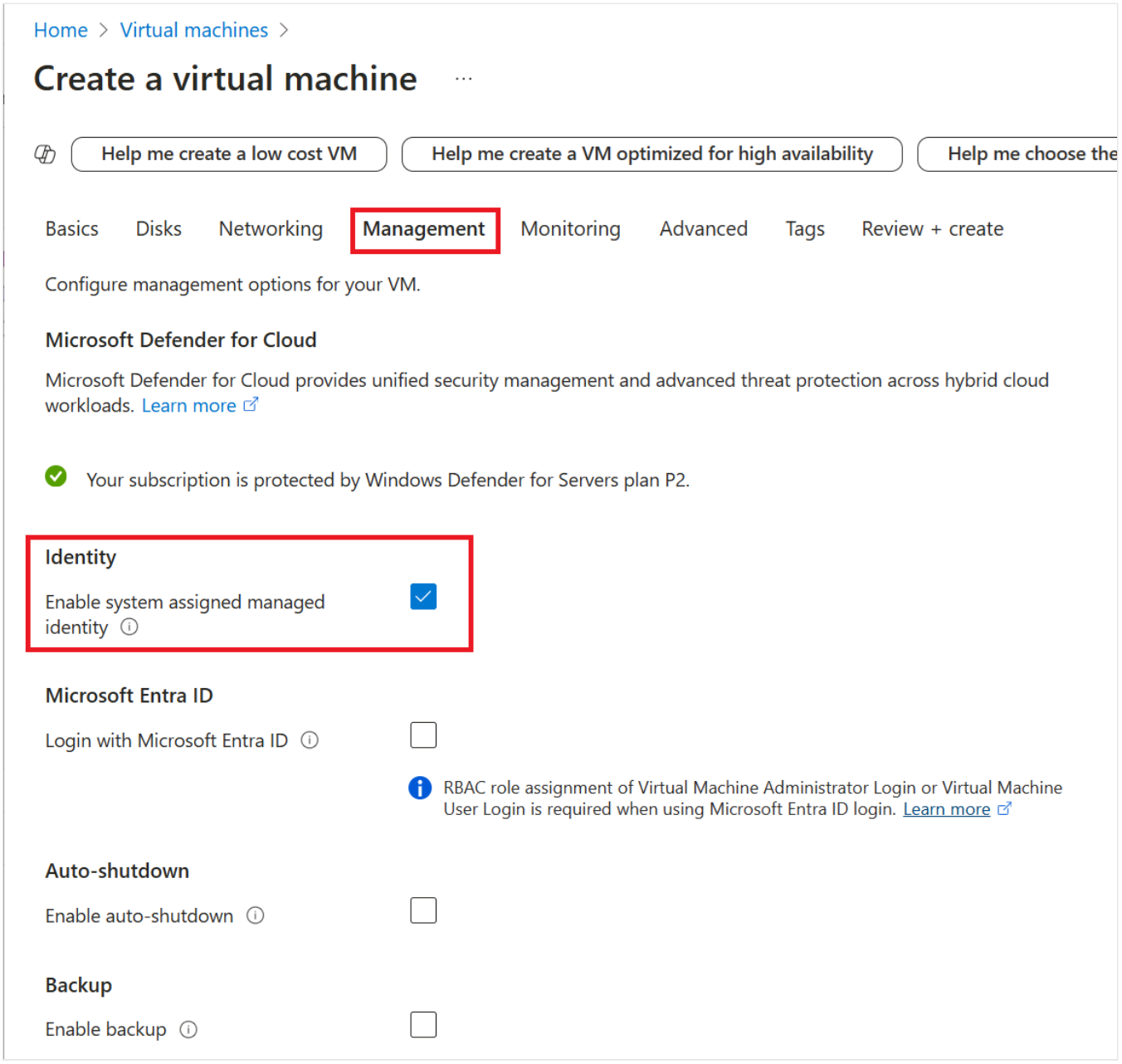
In this section, you learn how to enable and disable the system-assigned managed identity for VM using the Azure portal.

Enable system-assigned managed identity during creation of a VM

To enable system-assigned managed identity on a VM during its creation, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

When creating a Windows virtual machine or Linux virtual machine, select the **Management** tab.

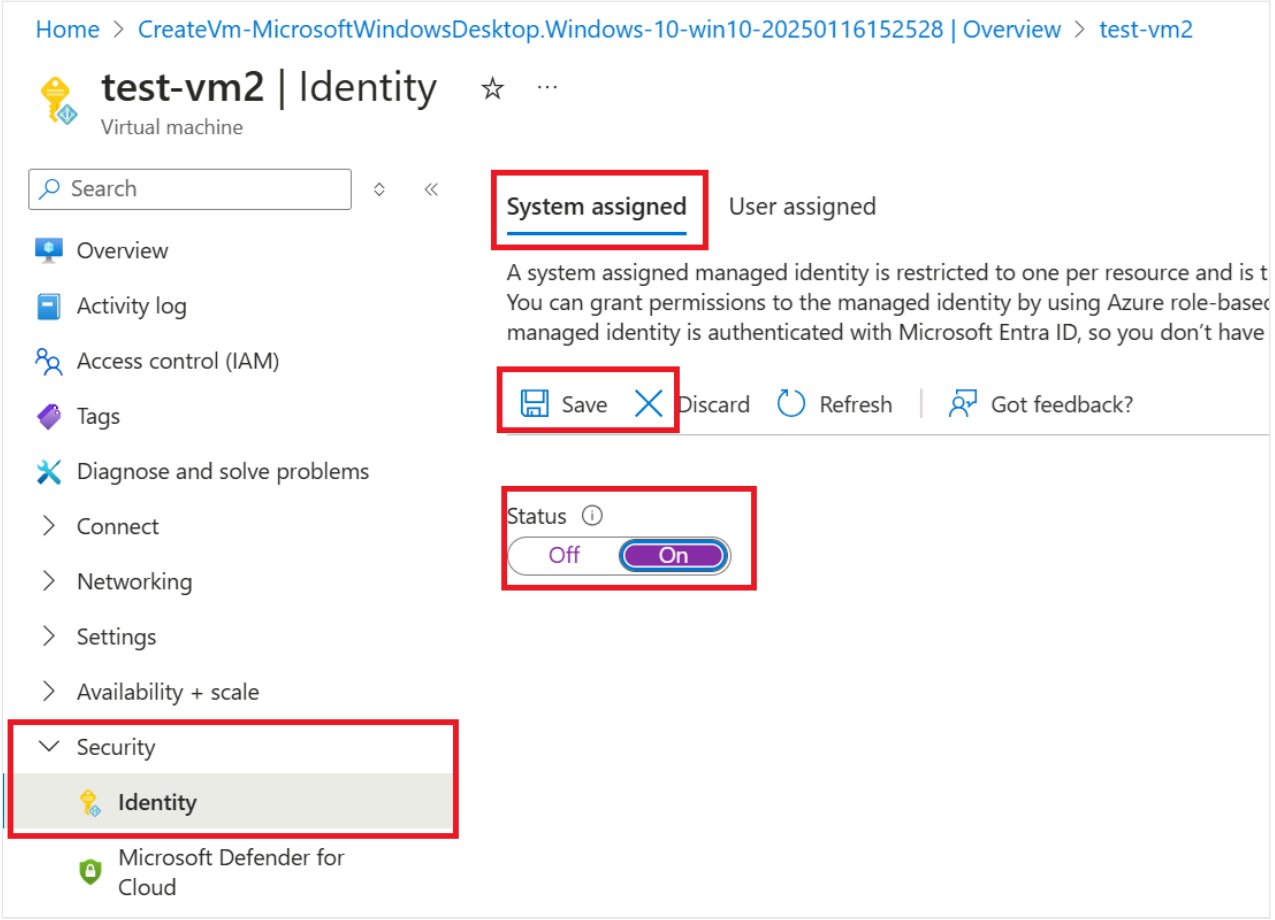
In the **Identity** section, select the **Enable system assigned managed identity** check-box.



Enable system-assigned managed identity on an existing VM

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **On** and then click **Save**:

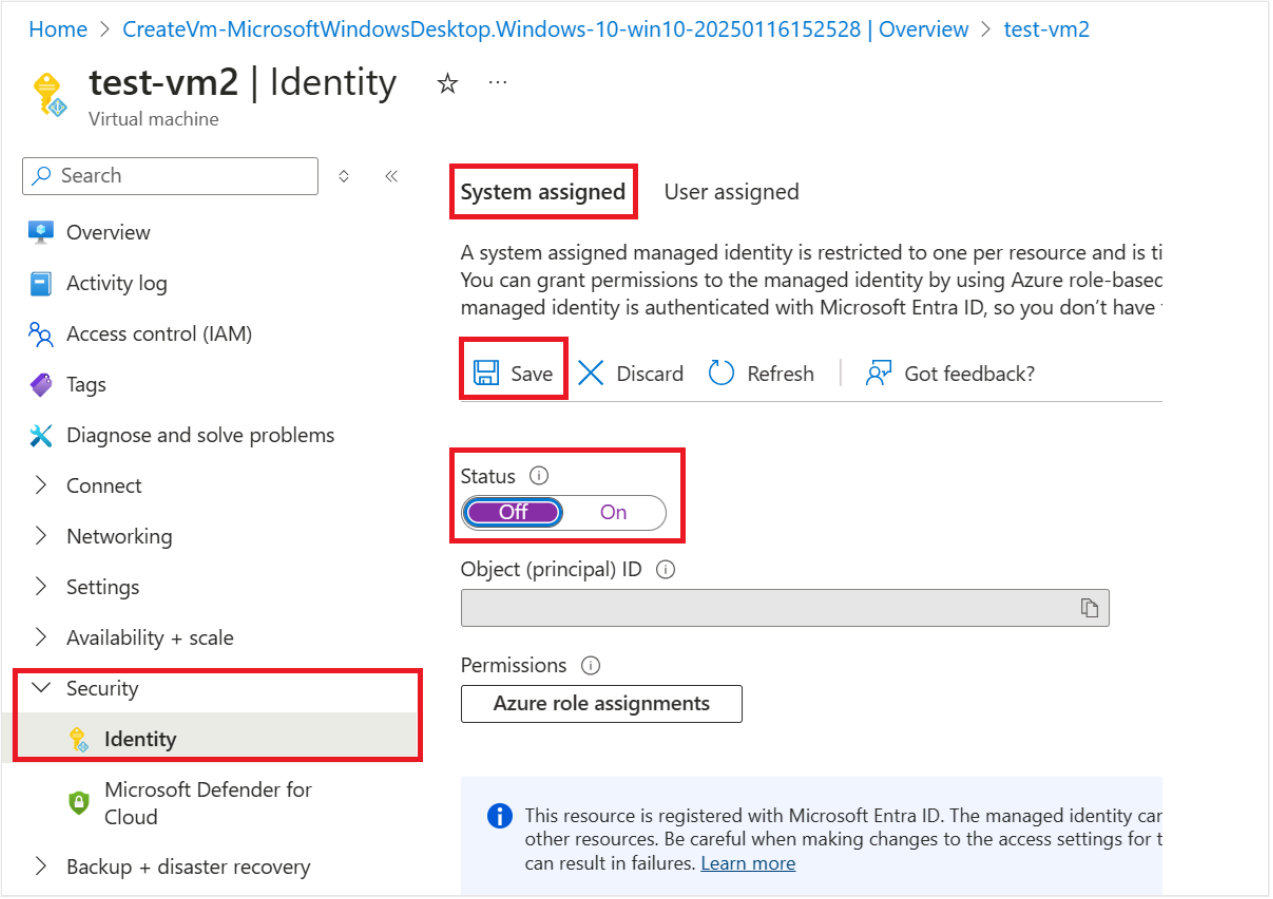


Remove system-assigned managed identity from a VM

To remove system-assigned managed identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

If you have a Virtual Machine that no longer needs system-assigned managed identity:

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and in the **Security** section select **Identity**.
3. Under **System assigned**, **Status**, select **Off** and then click **Save**:



User-assigned managed identity

In this section, you learn how to add and remove a user-assigned managed identity from a VM using the Azure portal.

Assign a user-assigned identity during the creation of a VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

Currently, the Azure portal does not support assigning a user-assigned managed identity during the creation of a VM. First create a Windows virtual machine or Linux virtual machine, then assign a user-assigned managed identity to the VM.

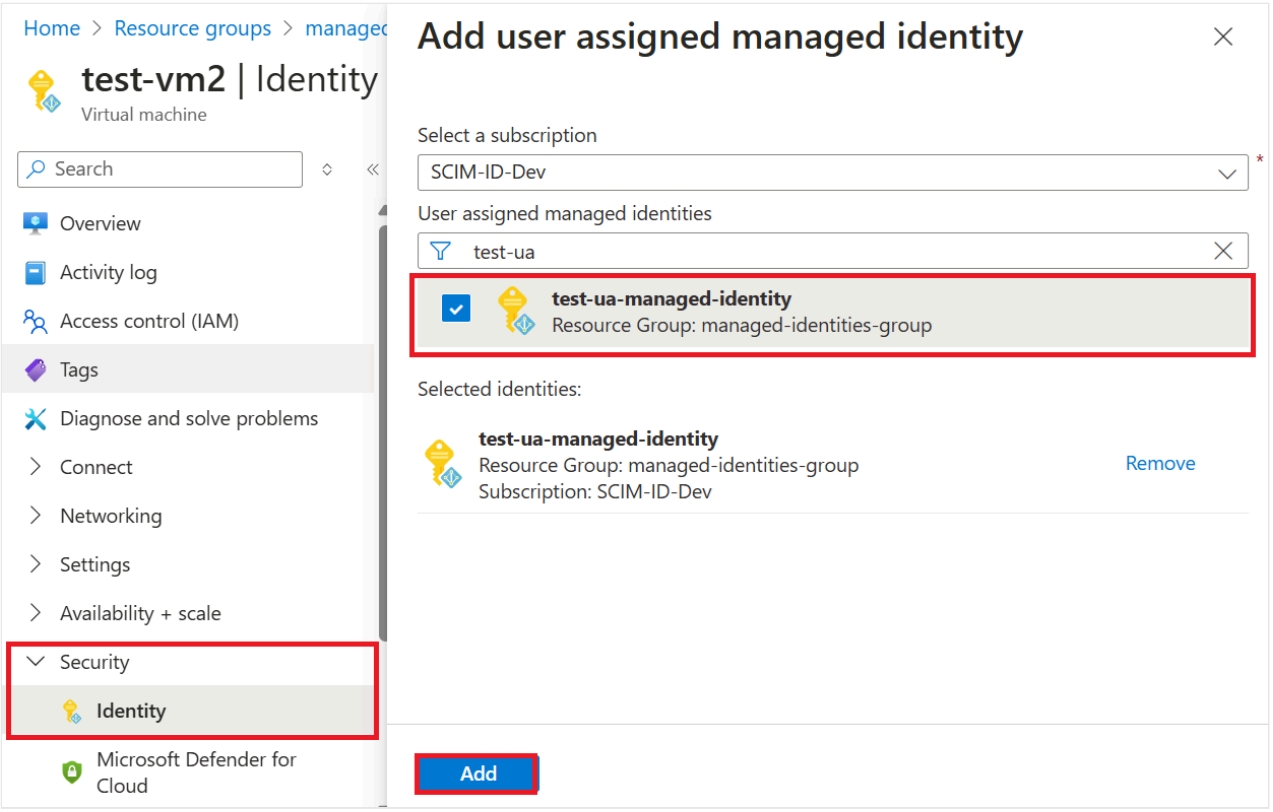
Assign a user-assigned managed identity to an existing VM

To assign a user-assigned identity to a VM, your account needs the Virtual Machine Contributor and Managed Identity Operator role assignments. No other Microsoft Entra directory role assignments are required.

1. Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

2. Navigate to the desired VM and click **Security** > **Identity**, **User assigned** and then **+Add**. Click the user-assigned identity you want to add to the VM and then click **Add**.

3. Select the previously created user assigned managed identity from the list.

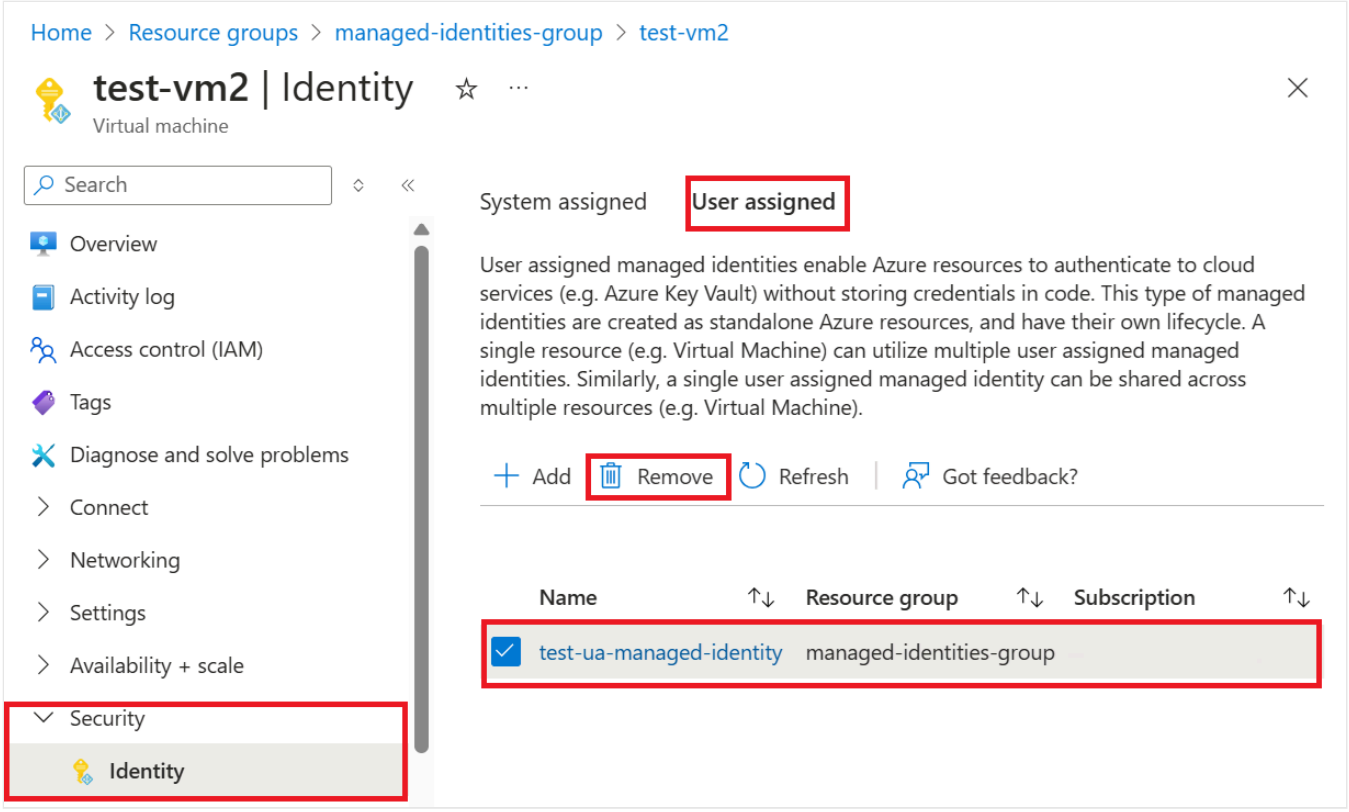


Remove a user-assigned managed identity from a VM

To remove a user-assigned identity from a VM, your account needs the Virtual Machine Contributor role assignment. No other Microsoft Entra directory role assignments are required.

Sign in to the Azure portal using an account associated with the Azure subscription that contains the VM.

Navigate to the desired VM and select **Security** > **Identity**, **User assigned**, the name of the user-assigned managed identity you want to delete and then click **Remove** (click **Yes** in the confirmation pane).



Next steps

- Using the Azure portal, give an Azure VM's managed identity access to another Azure resource.

Last updated on 01/16/2025

Sign in to a Linux virtual machine in Azure by using Microsoft Entra ID and OpenSSH

06/27/2025

To improve the security of Azure Linux virtual machines (VMs) or Azure Arc-enabled Linux servers, you can integrate with Microsoft Entra authentication. You can now use Microsoft Entra ID as a core authentication platform and a certificate authority to SSH into a Linux VM by using Microsoft Entra ID and OpenSSH certificate-based authentication. This functionality allows organizations to manage access to VMs with Azure role-based access control (RBAC) and Conditional Access policies.

This article shows you how to create and configure a Linux VM and log in with Microsoft Entra ID by using OpenSSH certificate-based authentication.

There are many security benefits of using Microsoft Entra ID with OpenSSH certificate-based authentication to sign in to Linux VMs in Azure. They include:

- Use your Microsoft Entra credentials to sign in to Azure Linux VMs.
- Get SSH key-based authentication without needing to distribute SSH keys to users or provision SSH public keys on any Azure Linux VMs that you deploy. This experience is much simpler than having to worry about sprawl of stale SSH public keys that could cause unauthorized access.
- Reduce reliance on local administrator accounts, credential theft, and weak credentials.
- Help secure Linux VMs by configuring password complexity and password lifetime policies for Microsoft Entra ID.
- With RBAC, specify who can sign in to a VM as a regular user or with administrator privileges. When users join your team, you can update the Azure RBAC policy for the VM to grant access as appropriate. When employees leave your organization and their user accounts are disabled or removed from Microsoft Entra ID, they no longer have access to your resources.
- With Conditional Access, configure policies to require multifactor authentication or to require that your client device is managed (for example, compliant or Microsoft Entra hybrid joined) before you can use it SSH into Linux VMs.
- Use Azure deploy and audit policies to require Microsoft Entra login for Linux VMs and flag unapproved local accounts.

Sign in to Linux VMs with Microsoft Entra ID works for customers who use Active Directory Federation Services.

Supported Linux distributions and Azure regions

The following Linux distributions are currently supported for deployments in a supported region:

Expand table

Distribution	Version
AlmaLinux	AlmaLinux 8, AlmaLinux 9
Azure Linux (formerly known as Common Base Linux Mariner)	CBL-Mariner 2.0, Azure Linux 3.0
Debian	Debian 9, Debian 10, Debian 11, Debian 12
openSUSE	openSUSE Leap 42.3, openSUSE Leap 15.1 to 15.5, openSUSE Leap 15.6+
Oracle	Oracle Linux 8, Oracle Linux 9
RedHat Enterprise Linux (RHEL)	RHEL 7.4 to RHEL 7.9, RHEL 8.3+, RHEL 9.0+
Rocky	Rocky 8, Rocky 9
SUSE Linux Enterprise Server (SLES)	SLES 12, SLES 15.1 to 15.5, SLES 15.6+
Ubuntu	Ubuntu 16.04 to Ubuntu 24.04

> **Note**
>
> SUSE made a breaking change with version 15.6 that is incompatible with the older versions. Since the Microsoft Entra login VM extension always installs the latest package, this will not work on older SUSE versions. You can install the aadsshlogin packages from packages.microsoft.com for older SUSE versions. After adding the repo, one can manually install them with this command: `sudo zypper install aadsshlogin=1.0.027980001`.

The following Azure regions are currently supported for this feature:

- Azure Global
- Azure Government
- Microsoft Azure operated by 21Vianet

Use of the SSH extension for the Azure CLI on Azure Kubernetes Service (AKS) clusters is not supported. For more information, see Support policies for AKS.

If you choose to install and use the Azure CLI locally, it must be version 2.22.1 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install the Azure CLI.

> **Note**
>
> This functionality is also available for Azure Arc-enabled servers.

Meet requirements for login with Microsoft Entra ID using OpenSSH certificate-based authentication

To enable Microsoft Entra login through SSH certificate-based authentication for Linux VMs in Azure, be sure to meet the following network, virtual machine, and client (SSH client) requirements.

Network

VM network configuration must permit outbound access to the following endpoints over TCP port 443.

Azure Global:

- `https://packages.microsoft.com`: For package installation and upgrades.
- `http://169.254.169.254`: Azure Instance Metadata Service endpoint.
- `https://login.microsoftonline.com`: For PAM-based (pluggable authentication modules) authentication flows.
- `https://pas.windows.net`: For Azure RBAC flows.

Azure Government:

- `https://packages.microsoft.com`: For package installation and upgrades.
- `http://169.254.169.254`: Azure Instance Metadata Service endpoint.
- `https://login.microsoftonline.us`: For PAM-based authentication flows.
- `https://pasff.usgovcloudapi.net`: For Azure RBAC flows.

Microsoft Azure operated by 21Vianet:

- `https://packages.microsoft.com`: For package installation and upgrades.
- `http://169.254.169.254`: Azure Instance Metadata Service endpoint.
- `https://login.chinacloudapi.cn`: For PAM-based authentication flows.
- `https://pas.chinacloudapi.cn`: For Azure RBAC flows.

Virtual machine

Ensure that your VM is configured with the following functionality:

- System-assigned managed identity. This option is automatically selected when you use the Azure portal to create VMs and select the Microsoft Entra login option. You can also enable system-assigned managed identity on a new or existing VM by using the Azure CLI.
- `aadsshlogin` and `aadsshlogin-selinux` (as appropriate). These packages are installed with the AADSSHLoginForLinux VM extension. The extension is installed when you use the Azure portal or the Azure CLI to create VMs and enable Microsoft Entra login (**Management** tab).

Client

Ensure that your client meets the following requirements:

- SSH client support for OpenSSH-based certificates for authentication. You can use the Azure CLI (2.21.1 or later) with OpenSSH (included in Windows 10 version 1803 or later) or Azure Cloud Shell to meet this requirement.
- SSH extension for the Azure CLI. You can install this extension by using `az extension add --name ssh`. You don't need to install this extension when you're using Azure Cloud Shell, because it comes preinstalled.

  If you're using any SSH client other than the Azure CLI or Azure Cloud Shell that supports OpenSSH certificates, you'll still need to use the Azure CLI with the SSH extension to retrieve ephemeral SSH certificates and optionally a configuration file. You can then use the configuration file with your SSH client.
- TCP connectivity from the client to either the public or private IP address of the VM. (ProxyCommand or SSH forwarding to a machine with connectivity also works.)

> **Important**
>
> SSH clients based on PuTTY now supports OpenSSH certificates and can be used to log in with Microsoft Entra OpenSSH certificate-based authentication.

Enable Microsoft Entra login for a Linux VM in Azure

To use Microsoft Entra login for a Linux VM in Azure, you need to first enable the Microsoft Entra login option for your Linux VM. You then configure Azure role assignments for users who

are authorized to sign in to the VM. Finally, you use the SSH client that supports OpenSSH, such as the Azure CLI or Azure Cloud Shell, to SSH into your Linux VM.

There are two ways to enable Microsoft Entra login for your Linux VM:

- The Azure portal experience when you're creating a Linux VM
- The Azure Cloud Shell experience when you're creating a Linux VM or using an existing one

Azure portal

You can enable Microsoft Entra login for any of the supported Linux distributions by using the Azure portal.

For example, to create an Ubuntu Server 18.04 long-term support (LTS) VM in Azure with Microsoft Entra login:

1. Sign in to the Azure portal by using an account that has access to create VMs, and then select + **Create a resource**.
2. Select **Create** under **Ubuntu Server 18.04 LTS** in the **Popular** view.
3. On the **Management** tab:
   a. Select the **Login with Microsoft Entra ID** checkbox.
   b. Ensure that the **System assigned managed identity** checkbox is selected.
4. Go through the rest of the experience of creating a virtual machine. You'll have to create an administrator account with username and password or SSH public key.

Azure Cloud Shell

Azure Cloud Shell is a free, interactive shell that you can use to run the steps in this article. Common Azure tools are preinstalled and configured in Cloud Shell for you to use with your account. Just select the **Copy** button to copy the code, paste it in Cloud Shell, and then select the Enter key to run it.

There are a few ways to open Cloud Shell:

- Select **Try It** in the upper-right corner of a code block.
- Open Cloud Shell in your browser.
- Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal.

If you choose to install and use the Azure CLI locally, this article requires you to use version 2.22.1 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install the Azure CLI.

1. Create a resource group by running az group create.
2. Create a VM by running az vm create. Use a supported distribution in a supported region.
3. Install the Microsoft Entra login VM extension by using az vm extension set.

The following example deploys a VM and then installs the extension to enable Microsoft Entra login for a Linux VM. VM extensions are small applications that provide post-deployment configuration and automation tasks on Azure Virtual Machines. Customize the example as needed to support your testing requirements.

Azure CLI

```
az group create --name AzureADLinuxVM --location southcentralus
az vm create \
    --resource-group AzureADLinuxVM \
    --name myVM \
    --image Ubuntu2204 \
    --assign-identity \
    --admin-username azureuser \
    --generate-ssh-keys
az vm extension set \
    --publisher Microsoft.Azure.ActiveDirectory \
    --name AADSSHLoginForLinux \
    --resource-group AzureADLinuxVM \
    --vm-name myVM
```

It takes a few minutes to create the VM and supporting resources.

The AADSSHLoginForLinux extension can be installed on an existing (supported distribution) Linux VM with a running VM agent to enable Microsoft Entra authentication. If you're deploying this extension to a previously created VM, the VM must have at least 1 GB of memory allocated or the installation will fail.

The `provisioningState` value of `Succeeded` appears when the extension is successfully installed on the VM. The VM must have a running VM agent to install the extension.

Enable Microsoft Entra login for a Azure Arc-enabled Linux servers

You can find the relevant details on SSH access to Azure Arc-enabled servers

Configure role assignments for the VM

Now that you've created the VM, you need to assign one of the following Azure roles to determine who can sign in to the VM. To assign these roles, you must have the Virtual Machine

Data Access Administrator role, or any role that includes the `Microsoft.Authorization/roleAssignments/write` action such as the Role Based Access Control Administrator role. However, if you use a different role than Virtual Machine Data Access Administrator, we recommend you add a condition to reduce the permission to create role assignments.

- **Virtual Machine Administrator Login:** Users who have this role assigned can sign in to an Azure virtual machine with administrator privileges.
- **Virtual Machine User Login:** Users who have this role assigned can sign in to an Azure virtual machine with regular user privileges.

To allow a user to sign in to a VM over SSH, you must assign the Virtual Machine Administrator Login or Virtual Machine User Login role on the resource group that contains the VM and its associated virtual network, network interface, public IP address, or load balancer resources.

An Azure user who has the Owner or Contributor role assigned for a VM doesn't automatically have privileges to Microsoft Entra sign in to the VM over SSH. There's an intentional (and audited) separation between the set of people who control virtual machines and the set of people who can access virtual machines.

There are two ways to configure role assignments for a VM:

- Azure portal experience
- Azure Cloud Shell experience

> **Note**
>
> The Virtual Machine Administrator Login and Virtual Machine User Login roles use `dataActions` and can be assigned at the management group, subscription, resource group, or resource scope. We recommend that you assign the roles at the management group, subscription, or resource group level and not at the individual VM level. This practice avoids the risk of reaching the Azure role assignments limit per subscription.

Azure portal

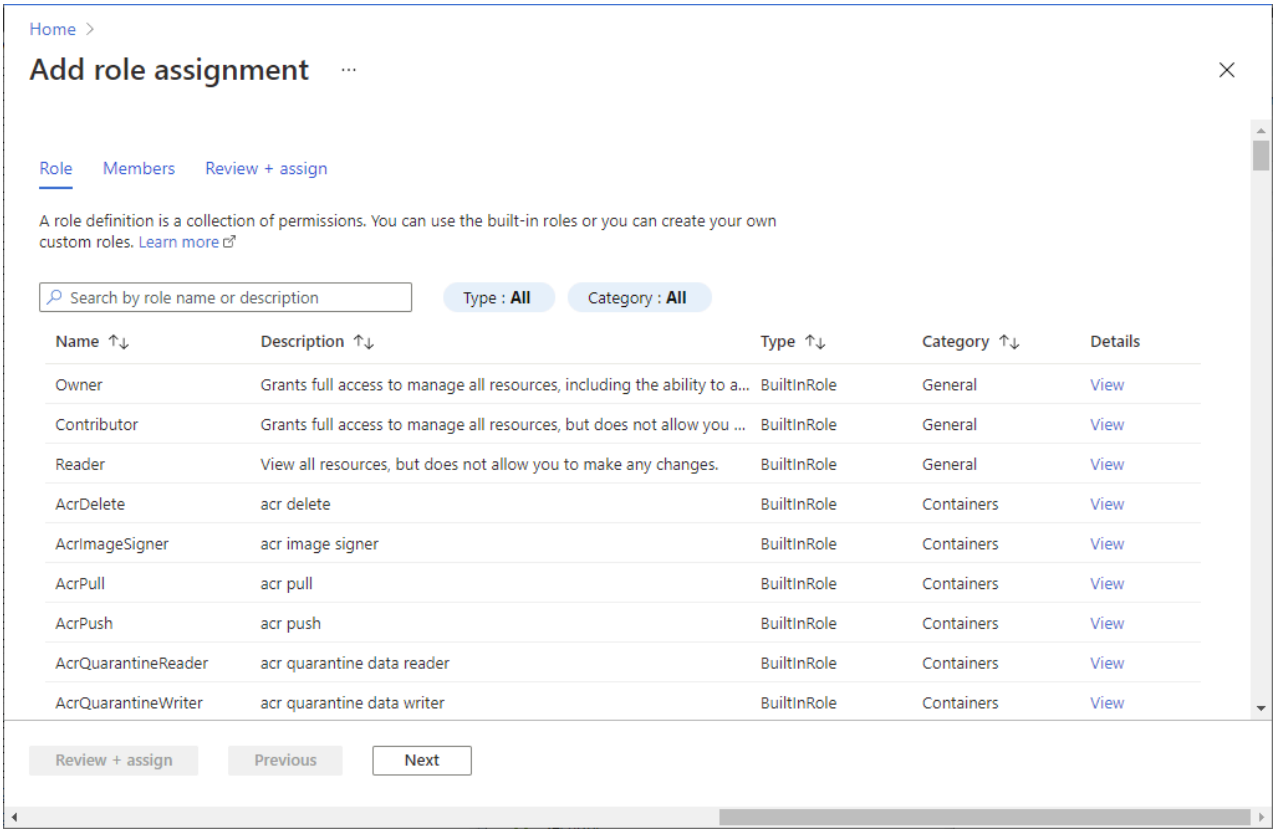
To configure role assignments for your Microsoft Entra ID-enabled Linux VMs:

1. For **Resource Group**, select the resource group that contains the VM and its associated virtual network, network interface, public IP address, or load balancer resource.
2. Select **Access control (IAM)**.
3. Select **Add** > **Add role assignment** to open the **Add role assignment** page.

4. Assign the following role. For detailed steps, see Assign Azure roles by using the Azure portal.

Expand table

Setting	Value
Role	**Virtual Machine Administrator Login** or **Virtual Machine User Login**
Assign access to	User, group, service principal, or managed identity



After a few moments, the security principal is assigned the role at the selected scope.

Azure Cloud Shell

The following example uses az role assignment create to assign the Virtual Machine Administrator Login role to the VM for your current Azure user. You obtain the username of your current Azure account by using az account show, and you set the scope to the VM created in a previous step by using az vm show.

You can also assign the scope at a resource group or subscription level. Normal Azure RBAC inheritance permissions apply.

Azure CLI

```
username=$(az account show --query user.name --output tsv)
rg=$(az group show --resource-group myResourceGroup --query id -o tsv)

az role assignment create \
    --role "Virtual Machine Administrator Login" \
    --assignee $username \
    --scope $rg
```

> **Note**
>
> If your Microsoft Entra domain and login username domain don't match, you must specify the object ID of your user account by using `--assignee-object-id`, not just the username for `--assignee`. You can obtain the object ID for your user account by using **az ad user list**.

For more information on how to use Azure RBAC to manage access to your Azure subscription resources, see Steps to assign an Azure role.

Install the SSH extension for the Azure CLI

If you're using Azure Cloud Shell, no other setup is needed because both the minimum required version of the Azure CLI and the SSH extension for the Azure CLI are already included in the Cloud Shell environment.

Run the following command to add the SSH extension for the Azure CLI:

Azure CLI

```
az extension add --name ssh
```

The minimum version required for the extension is 0.1.4. Check the installed version by using the following command:

Azure CLI

```
az extension show --name ssh
```

Enforce Conditional Access policies

You can enforce Conditional Access policies that are enabled with Microsoft Entra login, such as:

- Requiring multifactor authentication.
- Requiring a compliant or Microsoft Entra hybrid joined device for the device running the SSH client.
- Checking for risks before authorizing access to Linux VMs in Azure.

The application that appears in the Conditional Access policy is called *Azure Linux VM Sign-In*.

> **Note**
>
> Conditional Access policy enforcement that requires device compliance or Microsoft Entra hybrid join on the device that's running the SSH client works only with the Azure CLI that's running on Windows and macOS. It's not supported when you're using the Azure CLI on Linux or Azure Cloud Shell.

Missing application

If the Azure Linux VM Sign-In application is missing from Conditional Access, make sure the application isn't in the tenant:

1. Sign in to the Microsoft Entra admin center as at least a Cloud Application Administrator.
2. Browse to **Entra ID** > **Enterprise apps**.
3. Remove the filters to see all applications, and search for **Virtual Machine**. If you don't see Microsoft Azure Linux Virtual Machine Sign-In as a result, the service principal is missing from the tenant.

Log in by using a Microsoft Entra user account to SSH into the Linux VM

Log in by using the Azure CLI

Enter `az login`. This command opens a browser window, where you can sign in by using your Microsoft Entra account.

Azure CLI

```
az login
```

Then enter `az ssh vm`. The following example automatically resolves the appropriate IP address for the VM.

Azure CLI

```
az ssh vm -n myVM -g AzureADLinuxVM
```

If you're prompted, enter your Microsoft Entra login credentials at the login page, perform multifactor authentication, and/or satisfy device checks. You'll be prompted only if your the Azure CLI session doesn't already meet any required Conditional Access criteria. Close the browser window, return to the SSH prompt, and you'll be automatically connected to the VM.

You're now signed in to the Linux virtual machine with the role permissions as assigned, such as VM User or VM Administrator. If your user account is assigned the Virtual Machine Administrator Login role, you can use sudo to run commands that require root privileges.

Log in by using Azure Cloud Shell

You can use Azure Cloud Shell to connect to VMs without needing to install anything locally to your client machine. Start Cloud Shell by selecting the shell icon in the upper-right corner of the Azure portal.

Cloud Shell automatically connects to a session in the context of the signed-in user. Now run `az login` again and go through the interactive sign-in flow:

Azure CLI

```
az login
```

Then you can use the normal `az ssh vm` commands to connect by using the name and resource group or IP address of the VM:

Azure CLI

```
az ssh vm -n myVM -g AzureADLinuxVM
```

> **Note**
>
> Conditional Access policy enforcement that requires device compliance or Microsoft Entra hybrid join is not supported when you're using Azure Cloud Shell.

Log in by using the Microsoft Entra service principal to SSH into the Linux VM

The Azure CLI supports authenticating with a service principal instead of a user account. Because service principals aren't tied to any particular user, customers can use them to SSH into a VM to support any automation scenarios they might have. The service principal must have VM Administrator or VM User rights assigned. Assign permissions at the subscription or resource group level.

The following example will assign VM Administrator rights to the service principal at the resource group level. Replace the placeholders for service principal object ID, subscription ID, and resource group name.

Azure CLI

```
az role assignment create \
    --role "Virtual Machine Administrator Login" \
    --assignee-object-id <service-principal-objectid> \
    --assignee-principal-type ServicePrincipal \
    --scope "/subscriptions/<subscription-id>/resourceGroups/<resourcegroup-name>"
```

Use the following example to authenticate to the Azure CLI by using the service principal. For more information, see the article Sign in to the Azure CLI with a service principal.

Azure CLI

```
az login --service-principal -u <sp-app-id> -p <password-or-cert> --tenant
<tenant-id>
```

When authentication with a service principal is complete, use the normal the Azure CLI SSH commands to connect to the VM:

Azure CLI

```
az ssh vm -n myVM -g AzureADLinuxVM
```

Export the SSH configuration for use with SSH clients that support OpenSSH

Sign in to Azure Linux VMs with Microsoft Entra ID supports exporting the OpenSSH certificate and configuration. That means you can use any SSH clients that support OpenSSH-based certificates to sign in through Microsoft Entra ID. The following example exports the configuration for all IP addresses assigned to the VM:

Azure CLI

```
az ssh config --file ~/.ssh/config -n myVM -g AzureADLinuxVM
```

Alternatively, you can export the configuration by specifying just the IP address. Replace the IP address in the following example with the public or private IP address for your VM. (You must bring your own connectivity for private IPs.) Enter `az ssh config -h` for help with this command.

Azure CLI

```
az ssh config --file ~/.ssh/config --ip 10.11.123.456
```

You can then connect to the VM through normal OpenSSH usage. Connection can be done through any SSH client that uses OpenSSH.

Run sudo with Microsoft Entra login

After users who are assigned the VM Administrator role successfully SSH into a Linux VM, they'll be able to run sudo with no other interaction or authentication requirement. Users who are assigned the VM User role won't be able to run sudo.

Connect to VMs in virtual machine scale sets

Virtual machine scale sets are supported, but the steps are slightly different for enabling and connecting to VMs in a virtual machine scale set:

1. Create a virtual machine scale set or choose one that already exists. Enable a system-assigned managed identity for your virtual machine scale set:

   Azure CLI

   ```
   az vmss identity assign --name myVMSS --resource-group AzureADLinuxVM
   ```

2. Install the Microsoft Entra extension on your virtual machine scale set:

   Azure CLI

   ```
   az vmss extension set --publisher Microsoft.Azure.ActiveDirectory --name
   AADSSHLoginForLinux --resource-group AzureADLinuxVM --vmss-name myVMSS
   ```

Virtual machine scale sets usually don't have public IP addresses. You must have connectivity to them from another machine that can reach their Azure virtual network. This example shows

how to use the private IP of a VM in a virtual machine scale set to connect from a machine in the same virtual network:

Azure CLI

```
az ssh vm --ip 10.11.123.456
```

> **Note**
>
> You can't automatically determine the virtual machine scale set VM's IP addresses by using the `--resource-group` and `--name` switches.

Migrate from the previous (preview) version

If you're using the previous version of Microsoft Entra login for Linux that was based on device code flow, complete the following steps by using the Azure CLI:

1. Uninstall the AADLoginForLinux extension on the VM:

   Azure CLI

   ```
   az vm extension delete -g MyResourceGroup --vm-name MyVm -n AADLoginForLinux
   ```

   > **Note**
   >
   > Uninstallation of the extension can fail if there are any Microsoft Entra users currently logged in on the VM. Make sure all users are logged out first.

2. Enable system-assigned managed identity on your VM:

   Azure CLI

   ```
   az vm identity assign -g myResourceGroup -n myVm
   ```

3. Install the AADSSHLoginForLinux extension on the VM:

   Azure CLI

   ```
   az vm extension set \
       --publisher Microsoft.Azure.ActiveDirectory \
       --name AADSSHLoginForLinux \
   ```

```
    --resource-group myResourceGroup \
    --vm-name myVM
```

Use Azure Policy to meet standards and assess compliance

Use Azure Policy to:

- Ensure that Microsoft Entra login is enabled for your new and existing Linux virtual machines.
- Assess compliance of your environment at scale on a compliance dashboard.

With this capability, you can use many levels of enforcement. You can flag new and existing Linux VMs within your environment that don't have Microsoft Entra login enabled. You can also use Azure Policy to deploy the Microsoft Entra extension on new Linux VMs that don't have Microsoft Entra login enabled, as well as remediate existing Linux VMs to the same standard.

In addition to these capabilities, you can use Azure Policy to detect and flag Linux VMs that have unapproved local accounts created on their machines. To learn more, review Azure Policy.

Troubleshoot sign-in issues

Use the following sections to correct common errors that can happen when you try to SSH with Microsoft Entra credentials.

Couldn't retrieve token from local cache

If you get a message that says the token couldn't be retrieved from the local cache, you must run `az login` again and go through an interactive sign-in flow. Review the section about logging in by using Azure Cloud Shell.

Access denied: Azure role not assigned

If you see an "Azure role not assigned" error on your SSH prompt, verify that you've configured Azure RBAC policies for the VM that grants the user either the Virtual Machine Administrator Login role or the Virtual Machine User Login role. If you're having problems with Azure role assignments, see the article Troubleshoot Azure RBAC.

Problems deleting the old (AADLoginForLinux) extension

If the uninstallation scripts fail, the extension might get stuck in a transitioning state. When this happens, the extension can leave packages that it's supposed to uninstall during its removal. In such cases, it's better to manually uninstall the old packages and then try to run the `az vm extension delete` command.

To uninstall old packages:

1. Log in as a local user with admin privileges.
2. Make sure there are no logged-in Microsoft Entra users. Call the `who -u` command to see who is logged in. Then use `sudo kill <pid>` for all session processes that the previous command reported.
3. Run `sudo apt remove --purge aadlogin` (Ubuntu/Debian), `sudo yum remove aadlogin` (RHEL), or `sudo zypper remove aadlogin` (openSUSE or SLES).
4. If the command fails, try the low-level tools with scripts disabled:
   a. For Ubuntu/Debian, run `sudo dpkg --purge aadlogin`. If it's still failing because of the script, delete the `/var/lib/dpkg/info/aadlogin.prerm` file and try again.
   b. For everything else, run `rpm -e --noscripts aadogin`.
5. Repeat steps 3-4 for package `aadlogin-selinux`.

Extension installation errors

Installation of the AADSSHLoginForLinux VM extension to existing computers might fail with one of the following known error codes.

Non-zero exit code 22

If you get exit code 22, the status of the AADSSHLoginForLinux VM extension shows as **Transitioning** in the portal.

This failure happens because a system-assigned managed identity is required.

The solution is to:

1. Uninstall the failed extension.
2. Enable a system-assigned managed identity on the Azure VM.
3. Run the extension installation command again.

Non-zero exit code 23

If you get exit code 23, the status of the AADSSHLoginForLinux VM extension shows as **Transitioning** in the portal.

This failure happens when the older AADLoginForLinux VM extension is still installed.

The solution is to uninstall the older AADLoginForLinux VM extension from the VM. The status of the new AADSSHLoginForLinux VM extension will then change to **Provisioning succeeded** in the portal.

Installation failures when using an HTTP proxy

The extension needs an HTTP connection to install packages and check for the existence of a system identity. It runs in the context of `walinuxagent.service` and requires a change to let the agent know about the proxy settings. Open `/lib/systemd/system/walinuxagent.service` file on the target machine and add the following line after `[Service]`:

```
[Service]
Environment="http_proxy=http://proxy.example.com:80/"
Environment="https_proxy=http://proxy.example.com:80/"
Environment="no_proxy=169.254.169.254"
```

Restart the agent (`sudo systemctl restart walinuxagent`). Now try again.

The az ssh vm command fails with KeyError access_token

If the `az ssh vm` command fails, you're using an outdated version of the Azure CLI client.

The solution is to upgrade the Azure CLI client to version 2.21.0 or later.

SSH connection is closed

After a user successfully signs in by using `az login`, connection to the VM through `az ssh vm -ip <address>` or `az ssh vm --name <vm_name> -g <resource_group>` might fail with "Connection closed by <ip_address> port 22."

One cause for this error is that the user isn't assigned to the Virtual Machine Administrator Login or Virtual Machine User Login role within the scope of this VM. In that case, the solution is to add the user to one of those Azure RBAC roles within the scope of this VM.

This error can also happen if the user is in a required Azure RBAC role, but the system-assigned managed identity has been disabled on the VM. In that case, perform these actions:

1. Enable the system-assigned managed identity on the VM.

2. Allow several minutes to pass before the user tries to connect by using `az ssh vm --ip <ip_address>`.

Connection problems with virtual machine scale sets

VM connections with virtual machine scale sets can fail if the scale set instances are running an old model.

Upgrading scale set instances to the latest model might resolve the problem, especially if an upgrade hasn't been done since the Microsoft Entra Login extension was installed. Upgrading an instance applies a standard scale set configuration to the individual instance.

AllowGroups or DenyGroups statements in sshd_config cause the first sign in to fail for Microsoft Entra users

If *sshd_config* contains either `AllowGroups` or `DenyGroups` statements, the first login fails for Microsoft Entra users. If the statement was added after users have already had a successful login, they can log in.

One solution is to remove `AllowGroups` and `DenyGroups` statements from *sshd_config*.

Another solution is to move `AllowGroups` and `DenyGroups` to a `match user` section in *sshd_config*. Make sure the match template excludes Microsoft Entra users.

Getting Permission Denied when trying to connect from Azure Shell to Linux Red Hat/Oracle 7.X VM.

The OpenSSH server version in the target VM 7.4 is too old. Version incompatible with OpenSSH client version 8.8. Refer to RSA SHA256 certificates no longer work for more information.

Workaround:

- Adding option `"PubkeyAcceptedKeyTypes= +ssh-rsa-cert-v01@openssh.com"` in the `az ssh vm` command.

Azure CLI

```
az ssh vm -n myVM -g MyResourceGroup -- -A -o "PubkeyAcceptedKeyTypes= +ssh-rsa-cert-v01@openssh.com"
```

- Adding the option `"PubkeyAcceptedKeyTypes= +ssh-rsa-cert-v01@openssh.com"` in the `/home/<user>/.ssh/config file`.

Add the `"PubkeyAcceptedKeyTypes +ssh-rsa-cert-v01@openssh.com"` into the client config file.

config

```
Host *
PubkeyAcceptedKeyTypes +ssh-rsa-cert-v01@openssh.com
```

Next steps

- What is a device identity?
- Common Conditional Access policies

Sign in to Windows virtual machine in Azure using Microsoft Entra ID and Azure Roles Based Access Control

Organizations can improve the security of Windows Server virtual machines in Azure by integrating with Microsoft Entra authentication. You can now use Microsoft Entra ID as a core authentication platform to Remote Desktop Protocol (RDP) into supported versions of Windows Server. You can then centrally control and enforce Azure role-based access control (RBAC) and Conditional Access policies that allow or deny access to the virtual machines.

This article shows you how to create and configure a Windows Server virtual machine and sign in by using Microsoft Entra ID-based authentication.

There are many security benefits of using Microsoft Entra ID-based authentication to sign in to Windows Server virtual machines in Azure. They include:

- Use Microsoft Entra authentication including passwordless to sign in to Windows Server virtual machines. Reduce reliance on local administrator accounts.
- Use Password complexity and password lifetime policies that you configure for Microsoft Entra ID also help secure Windows Server virtual machines.
- Use Azure role-based access control:
  - Specify who can sign in as a regular user or with administrator privileges.
  - When users join or leave your team, you can update Azure role-based access control policy to grant access as appropriate.
  - When employees leave your organization and their user accounts are disabled or removed from Microsoft Entra ID, they no longer have access to your resources.
- Use Conditional Access policy "phishing resistant MFA" and other signals such as user sign-in risk.
- Use Azure Policy to deploy and audit policies to require Microsoft Entra sign in for Windows Server virtual machines and to flag the use of unapproved local accounts on the virtual machines.
- Use Intune to automate and scale Microsoft Entra join with mobile device management (MDM) autoenrollment of Azure Windows VMs that are part of your virtual desktop infrastructure (VDI) deployments. MDM autoenrollment requires Microsoft Entra ID P1 licenses.

MDM autoenrollment requires Microsoft Entra ID P1 licenses. Windows Server VMs don't support MDM enrollment.

> **Important**
>
> After you enable this capability, your Azure virtual machine will be Microsoft Entra joined. You can't join them to another domain, like on-premises Active Directory or Microsoft Entra Domain Services. If you need to do so, disconnect the device from Microsoft Entra by uninstalling the extension. In addition, if you deploy a supported golden image, you can enable Microsoft Entra ID authentication by installing the extension. Conditional Access isn't supported with Windows Server with Microsoft Entra join extension in Azure Windows virtual machines.

Requirements

Supported Azure regions and Windows distributions

This feature currently supports the following Windows Server distributions:

- Windows 11 24H2 or later installed.
- Windows Server 2025 or later installed with Desktop Experience.

This feature is now available in the following Azure clouds:

- Azure Global
- Azure Government
- Microsoft Azure operated by 21Vianet

> **Note**
>
> CIS hardened images support Microsoft Entra ID authentication for Microsoft Windows Enterprise and Microsoft Windows Server offerings. For more information, see: CIS Hardened Images on Microsoft Windows Enterprise.

Network requirements

To enable Microsoft Entra authentication to virtual machines in Azure, you need to ensure that your network configuration permits outbound access to the following endpoints over TCP port

443.

Azure Global:

- `https://enterpriseregistration.windows.net`: Device registration.
- `http://169.254.169.254`: Azure Instance Metadata Service endpoint.
- `https://login.microsoftonline.com`: Authentication flows.
- `https://pas.windows.net`: Azure role-based access control flows.

Azure Government:

- `https://enterpriseregistration.microsoftonline.us`: Device registration.
- `http://169.254.169.254`: Azure Instance Metadata Service endpoint.
- `https://login.microsoftonline.us`: Authentication flows.
- `https://pasff.usgovcloudapi.net`: Azure role-based access control flows.

Microsoft Azure operated by 21Vianet:

- `https://enterpriseregistration.partner.microsoftonline.cn`: Device registration.
- `http://169.254.169.254`: Azure Instance Metadata Service endpoint.
- `https://login.chinacloudapi.cn`: Authentication flows.
- `https://pas.chinacloudapi.cn`: Azure role-based access control flows.

Authentication requirements

- **User account type**: Only standard Microsoft Entra user accounts from the same tenant as the azure windows virtual machine are supported. Guest accounts (B2B users) cannot be used for authentication.
- **Managed identities**: You must enable system-assigned managed identity on your Azure virtual machine before installing the Microsoft Entra sign in virtual machine extension. Managed Identities are stored in a single Microsoft Entra tenant and currently don't support cross directory scenarios.
- **Role assignments**: Users must be assigned either the **Virtual Machine Administrator Login** or **Virtual Machine User Login** role to sign in to the VM. Having the Owner or

Contributor role alone doesn't grant sign-in privileges.

- **Virtual Machine Administrator Login:** Users who have this role assigned can sign in to an Azure virtual machine with administrator privileges.
- **Virtual Machine User Login:** Users who have this role assigned can sign in to an Azure virtual machine with regular user privileges.
- **Authentication methods**:
  - Passwordless authentication is recommended and supported via Microsoft Entra credentials
  - Password-based authentication is supported with proper role assignments
  - Windows Hello for Business is supported using the certificate trust model only (not key trust model)
  - Local administrator accounts created manually by adding users to local groups aren't supported

Enable Microsoft Entra sign in for a Windows virtual machine in Azure

To use Microsoft Entra sign in for a Windows virtual machine, you must:

1. Enable the Microsoft Entra sign in extension for the device.
2. Configure Azure role assignments for users.

Enable Microsoft Entra sign in extension

- Azure virtual machine running Windows

You must enable system-assigned managed identity on your Azure virtual machine before you install the Microsoft Entra sign in virtual machine extension. Managed Identities are stored in a single Microsoft Entra tenant and currently don't support cross directory scenarios.

The following sample demonstrates Azure templates for Azure Virtual Machine extensions.

JSON

```
{
  "$schema": "http://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
```

```
    "vmName": {
      "type": "string"
    },
    "location": {
      "type": "string"
    }
  },
  "resources": [
    {
      "name": "[concat(parameters('vmName'),'/AADLogin')]",
      "type": "Microsoft.Compute/virtualMachines/extensions",
      "location": "[parameters('location')]",
      "apiVersion": "2015-06-15",
      "properties": {
        "publisher": "Microsoft.Azure.ActiveDirectory",
        "type": "AADLoginForWindows",
        "typeHandlerVersion": "1.0",
        "autoUpgradeMinorVersion": true
      }
    }
  ]
}
```

Configure role assignments

To assign user roles, you must have the Virtual Machine Data Access Administrator role, or any role that includes the `Microsoft.Authorization/roleAssignments/write` action such as the Role Based Access Control Administrator role. However, if you use a different role than Virtual Machine Data Access Administrator, we recommend you add a condition to reduce the permission to create role assignments.

- **Virtual Machine Administrator Login:** Users who have this role assigned can sign in to an Azure virtual machine with administrator privileges.
- **Virtual Machine User Login:** Users who have this role assigned can sign in to an Azure virtual machine with regular user privileges.

> ⓘ Note
>
> Manually elevating a user to become a local administrator on the device by adding the user to a member of the local administrators' group or by running `net localgroup administrators /add "AzureAD\UserUpn"` command isn't supported. You need to use roles in Azure to authorize sign in.

> ⓘ Note

An Azure user who has the Owner or Contributor role assigned doesn't automatically have privileges to sign in to devices. The reason is to provide audited separation between the set of people who control virtual machines and the set of people who can access virtual machines.

The following documentation provides step-by-step details to add user accounts to role assignments in Azure:

- Assign Azure roles by using the Azure portal
- Assign Azure roles by using the Azure CLI
- Assign Azure roles by using Azure PowerShell

Sign in by using Microsoft Entra credentials to a Windows VM

You can sign in over RDP using one of two methods:

- Passwordless using any of the supported Microsoft Entra credentials (recommended)
- Password/passwordless using Windows Hello for Business deployed using certificate trust model

Sign in using passwordless authentication with Microsoft Entra ID

To use passwordless authentication for your Windows Server VMs in Azure, the session host (VM) must be running:

- Windows 11 with 2022-10 Cumulative Updates for Windows 11 (KB5018418) or later installed.
- Windows 10, version 20H2 or later with 2022-10 Cumulative Updates for Windows 10 (KB5018410) or later installed.
- Windows Server 2022 with 2022-10 Cumulative Update for Microsoft server operating system (KB5018421) or later installed.
- Windows Server 2025 or later installed.

Note

When using the **web account to sign in to the remote computer** option, there's no requirement for the local device to be joined to a domain or Microsoft Entra ID.

To connect to the remote computer:

- Launch **Remote Desktop Connection** from Windows Search, or by running `mstsc.exe`.
- Select **Use a web account to sign in to the remote computer** option in the **Advanced** tab. This option is equivalent to the `enablerdsaadauth` RDP property. For more information, see Supported RDP properties with Remote Desktop Services.
- Specify the name of the remote computer and select **Connect**.

> **Important**
>
> IP address can't be used with **Use a web account to sign in to the remote computer** option. The name must match the hostname of the remote device in Microsoft Entra ID and be network addressable, resolving to the IP address of the remote device.

- When prompted for credentials, specify your user name in `user@domain.com` format.
- You're then prompted to allow the remote desktop connection when connecting to a new PC. Microsoft Entra remembers up to 15 hosts for 30 days before prompting again. If you see this dialogue, select **Yes** to connect.

> **Important**
>
> If your organization is using Microsoft Entra Conditional Access, your device must satisfy the Conditional Access requirements to allow connection to the remote computer. Conditional Access policies might be applied to the application **Microsoft Remote Desktop (a4a365df-50f1-4397-bc59-1a1564b8bb9c)** for controlled access.

> **Note**
>
> The Windows lock screen in the remote session doesn't support Microsoft Entra authentication tokens or passwordless authentication methods like FIDO keys. The lack of support for these authentication methods means that users can't unlock their screens in a remote session. When you try to lock a remote session, either through user action or system policy, the session is instead disconnected and the service sends a message to the user. Disconnecting the session also ensures that when the connection is relaunched after

a period of inactivity, Microsoft Entra ID reevaluates the applicable Conditional Access policies.

Sign in using password/passwordless authentication with Microsoft Entra ID

Both Password-based authentication, and Passwordless authentication are supported to sign in to Windows virtual machines.

> **Important**
>
> Remote connection to VMs that are joined to Microsoft Entra ID is allowed only from Windows 10 or later PCs that are either Microsoft Entra registered(minimum required build is 20H1) or Microsoft Entra joined or Microsoft Entra hybrid joined to the *same* directory as the VM. Additionally, to RDP by using Microsoft Entra credentials, users must belong to one of the two Azure roles, Virtual Machine Administrator Login or Virtual Machine User Login.
>
> If you're using a Microsoft Entra registered windows 10 or later PC, you must enter credentials in the `AzureAD\UPN` format (for example, `AzureAD\john@contoso.com`). At this time, you can use Azure Bastion to sign in with Microsoft Entra authentication via the Azure CLI and the native RDP client mstsc.

To sign in to your Windows Server 2019 virtual machine by using Microsoft Entra ID:

1. Go to the overview page of the virtual machine enabled with Microsoft Entra sign in.
2. Select **Connect** to open the **Connect to virtual machine** pane.
3. Select **Download RDP File**.
4. Select **Open** to open the Remote Desktop Connection client.
5. Select **Connect** to open the Windows sign in dialog.
6. Sign in by using your Microsoft Entra credentials.

You're now signed in to the Windows Server 2019 Azure virtual machine with the role permissions as assigned, such as VM User or VM Administrator.

> **Note**

You can save the `.RDP` file locally on your computer to start future remote desktop connections to your virtual machine, instead of going to the virtual machine overview page in the Azure portal and using the connect option.

Enforce Conditional Access policies

You can enforce Conditional Access policies, such as "phishing resistant MFA" or user sign-in risk check, before users can access Windows Server virtual machines in Azure. To apply a Conditional Access policy, you must select the **Microsoft Azure Windows Virtual Machine Sign-in** app from the cloud apps or actions assignment option.

Conditional Access policies that restrict sign in using device configuration rules aren't supported when connecting from a Windows Server device. For a guide on setting conditional access policies, see: Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication.

> **Note**
>
> If you require MFA as a control, then you must supply an MFA claim as part of the client that initiates the RDP session to the target Windows Server virtual machine. The Remote Desktop application supports Conditional Access policies, however if you're using web sign in, you must use a Windows Hello for Business PIN or biometric authentication. Support for biometric authentication was added to the RDP client in Windows 10 version 1809. Remote desktop using Windows Hello for Business authentication is available only for deployments that use a certificate trust model and isn't available for a key trust model.

Use Azure Policy to meet standards and assess compliance

Use Azure Policy to:

- Ensure that Microsoft Entra sign in is enabled for your new and existing Windows Server virtual machines.
- Assess compliance of your environment at scale on a compliance dashboard.

With this capability, you can use many levels of enforcement. You can flag new and existing Windows Server virtual machines within your environment that don't have Microsoft Entra sign

in enabled. You can also use Azure Policy to deploy the Microsoft Entra extension to Windows virtual machines in Azure.

In addition to these capabilities, you can use Azure Policy to detect and flag Windows machines that have unapproved local accounts created on their devices. To learn more, review Azure Policy.

Troubleshoot deployment problems

The AADLoginForWindows extension must be installed successfully for the device to complete the Microsoft Entra join process. If the extension fails to be installed correctly, perform the following steps:

1. Connect to the device and examine the *CommandExecution.log* file under *C:\WindowsAzure\Logs\Plugins\Microsoft.Azure.ActiveDirectory.AADLoginForWindows\1.0.0.1*.

   If the extension restarts after the initial failure, the log with the deployment error will be saved as *CommandExecution_YYYYMMDDHHMMSSSSS.log*.

2. Open a PowerShell window on the device. Verify that the following queries against the Azure Instance Metadata Service endpoint running on the host return the expected output:

   For Azure virtual machines:

Expand table

Command to run	Expected output
`curl.exe -H Metadata:true "http://169.254.169.254/metadata/instance?api-version=2017-08-01"`	Correct information about the Azure virtual machine
`curl.exe -H Metadata:true "http://169.254.169.254/metadata/identity/info?api-version=2018-02-01"`	Valid tenant ID associated with the Azure subscription
`curl.exe -H Metadata:true "http://169.254.169.254/metadata/identity/oauth2/token?resource=urn:ms-drs:enterpriseregistration.windows.net&api-version=2018-02-01"`	Valid access token issued by Microsoft Entra ID for the managed identity that is assigned to this virtual machine

3. Ensure that the required endpoints are accessible from the device via PowerShell:

   - `curl.exe https://login.microsoftonline.com/ -D -`
   - `curl.exe https://login.microsoftonline.com/<TenantID>/ -D -`
   - `curl.exe https://enterpriseregistration.windows.net/ -D -`
   - `curl.exe https://device.login.microsoftonline.com/ -D -`
   - `curl.exe https://pas.windows.net/ -D -`

   Replace `<TenantID>` with the Microsoft Entra tenant ID associated with the Azure subscription. `login.microsoftonline.com/<TenantID>`, `enterpriseregistration.windows.net`, and `pas.windows.net` should return 404 Not Found, which is expected behavior.

4. View the device state by running `dsregcmd /status`. The goal is for the device state to show as `AzureAdJoined : YES`.

   Microsoft Entra join activity is captured in Event Viewer under the *User Device Registration\Admin* log at *Event Viewer (local)\Applications* and *Services Logs\Microsoft\Windows\User Device Registration\Admin*.

If the AADLoginForWindows extension fails with an error code, you can perform the following steps.

Device name already exists

If a device object with the same displayName as the hostname of the Azure virtual machine exists, the device fails to join Microsoft Entra with a hostname duplication error. Avoid duplication by modifying the hostname.

Terminal error code 1007 and exit code -2145648574

Terminal error code 1007 and exit code -2145648574 translate to `DSREG_E_MSI_TENANTID_UNAVAILABLE`. The extension can't query the Microsoft Entra tenant information.

Connect to the device as a local administrator and verify that the endpoint returns a valid tenant ID from Azure Instance Metadata Service. Run the following command from an elevated PowerShell window on the device:

```
curl -H Metadata:true http://169.254.169.254/metadata/identity/info?api-version=2018-02-01
```

This problem can also happen when the admin attempts to install the AADLoginForWindows extension, but the device doesn't have a system-assigned managed identity. In that case, go to the **Identity** pane of the device. On the **System assigned** tab, verify that the **Status** toggle is set to **On**.

Exit code -2145648607

Exit code -2145648607 translates to `DSREG_AUTOJOIN_DISC_FAILED`. The extension can't reach the `https://enterpriseregistration.windows.net` endpoint.

1. Verify that the required endpoints are accessible from the device via PowerShell:

   - `curl https://login.microsoftonline.com/ -D -`
   - `curl https://login.microsoftonline.com/<TenantID>/ -D -`
   - `curl https://enterpriseregistration.windows.net/ -D -`
   - `curl https://device.login.microsoftonline.com/ -D -`
   - `curl https://pas.windows.net/ -D -`

   Replace `<TenantID>` with the Microsoft Entra tenant ID of the Azure subscription. If you need to find the tenant ID, you can hover over your account name or select **Entra ID** > **Overview** > **Properties** > **Tenant ID**.

   Attempts to connect to `enterpriseregistration.windows.net` might return 404 Not Found, which is expected behavior. Attempts to connect to `pas.windows.net` might prompt for PIN credentials or might return 404 Not Found. (You don't need to enter the PIN.) Either one is sufficient to verify that the URL is reachable.

2. If any of the commands fails with "Couldn't resolve host `<URL>`," try running this command to determine which DNS server Windows is using:

   `nslookup <URL>`

   Replace `<URL>` with the fully qualified domain names that the endpoints use, such as `login.microsoftonline.com`.

3. See whether specifying a public DNS server allows the command to succeed:

   `nslookup <URL> 208.67.222.222`

4. If necessary, change the DNS server assigned to the network security group that the device belongs to.

Exit code 51

Exit code 51 translates to "This extension isn't supported on this operating system."

The AADLoginForWindows extension is intended to be installed only on Azure virtual machines with Windows Server 2022 or Windows 10 1809 or later. Ensure that your version of Windows Server is supported. If it isn't supported, uninstall the extension.

Troubleshoot sign-in problems

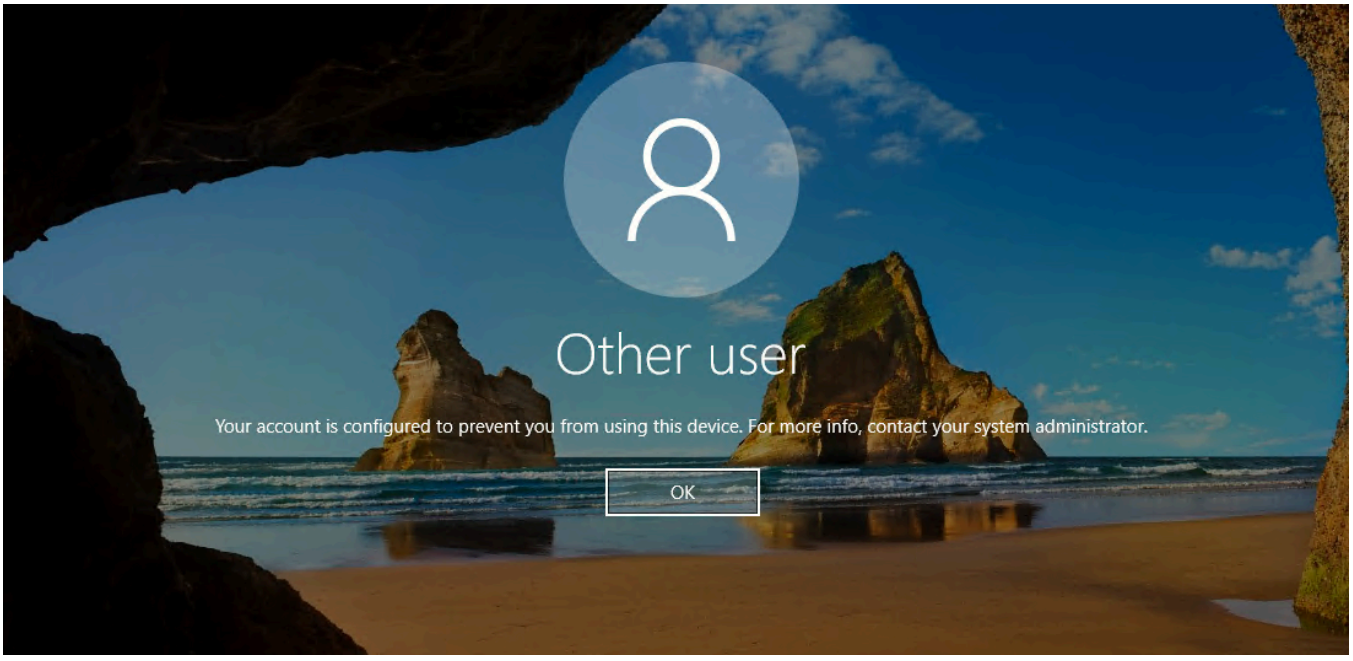
Use the following information to correct sign-in problems.

You can view the device and single sign-on (SSO) state by running `dsregcmd /status`. The goal is for the device state to show as `AzureAdJoined : YES` and for the SSO state to show `AzureAdPrt : YES`.

RDP sign-in via Microsoft Entra accounts is captured in Event Viewer under the *Applications and Services Logs\Microsoft\Windows\AAD\Operational* event logs.

Azure role not assigned

You might get the following error message when you initiate a remote desktop connection to your device: "Your account is configured to prevent you from using this device. For more info, contact your system administrator."

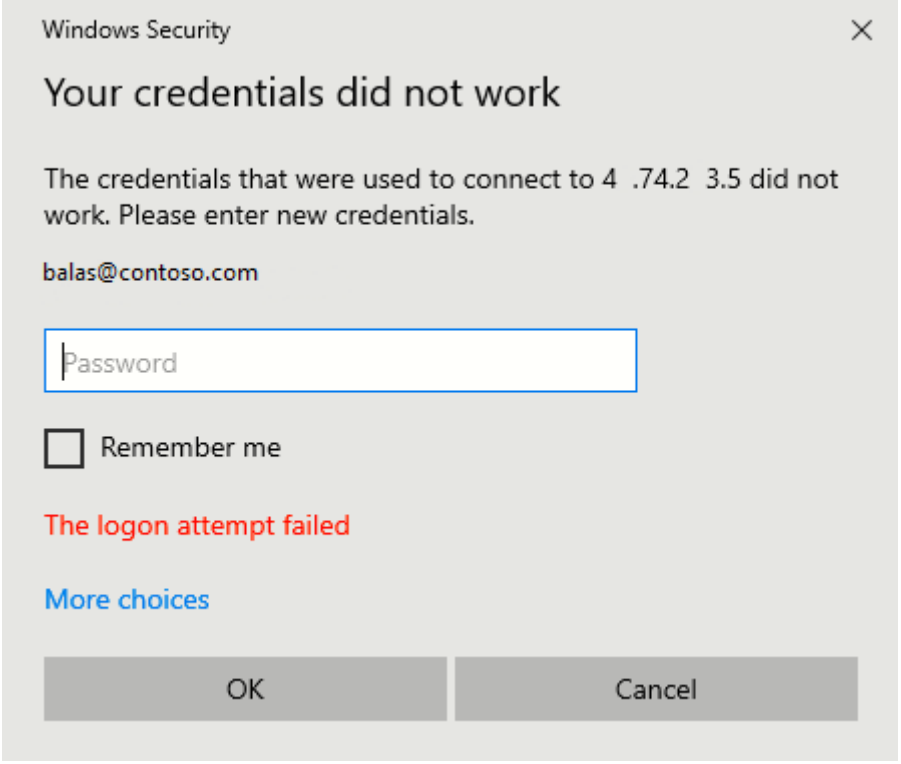


Verify the Azure role-based access control policies that grant the user the Virtual Machine Administrator Login or Virtual Machine User Login role.

If you're having problems with Azure role assignments, see Troubleshoot Azure role-based access control.

Unauthorized client or password change required

You might get the following error message when you initiate a remote desktop connection to your device: "Your credentials didn't work."



Try these solutions:

- The Windows 10 or later PC that you're using to initiate the remote desktop connection must be Microsoft Entra joined, or Microsoft Entra hybrid joined to the same Microsoft Entra directory. For more information about device identity, see the article What is a device identity?.

  Windows 10 Build 20H1 added support for a Microsoft Entra registered PC to initiate an RDP connection to your virtual machine. When you're using a client device that's Microsoft Entra registered (not Microsoft Entra joined or Microsoft Entra hybrid joined) as the RDP client to initiate connections to your virtual machine, you must enter credentials in the format `AzureAD\UPN` (for example, `AzureAD\john@contoso.com`).

  Verify that the AADLoginForWindows extension wasn't uninstalled after the Microsoft Entra join finished.

Also, make sure that the security policy **Network security: Allow PKU2U authentication requests to this computer to use online identities** is enabled on both the server *and* the client.

- Verify that the user doesn't have a temporary password. Temporary passwords can't be used to sign in to a remote desktop connection.

  Sign in with the user account in a web browser. For instance, sign in to the Azure portal in a private browsing window. If you're prompted to change the password, set a new password. Then try connecting again.

AADSTS293004: The target-device identifier in the request xxx wasn't found in the tenant xxx

Cause:

The computer name entered in mstsc doesn't match with any one of the "hostnames" attributes for target AADJ device. For example, AADJ device host name is the short name like device_1, but the computer name entered in mstsc is the FQDN like device_1.contoso.com.

Try these solutions:

There are multiple ways to resolve the issue:

1. Modify the HOSTS entry on client machine, add an A DNS record that points the correct device name (confirm from AAD Device record) to the IP of target machine. Use that device name in mstsc.
2. Check if the target machine is managed and if the hostname is set through Group Policy or MDM via the DNS Client PrimaryDnsSuffix value ADMX_DnsClient Policy CSP | Microsoft Learn. If this is set and is incorrect, it needs to be removed or set correctly.
3. When customer requires to use FQDN to connect but AAD Device name is a short name, login to the target machine via local admin, add a "**Primary DNS Suffix**" for their domain suffix. Detailed instructions:

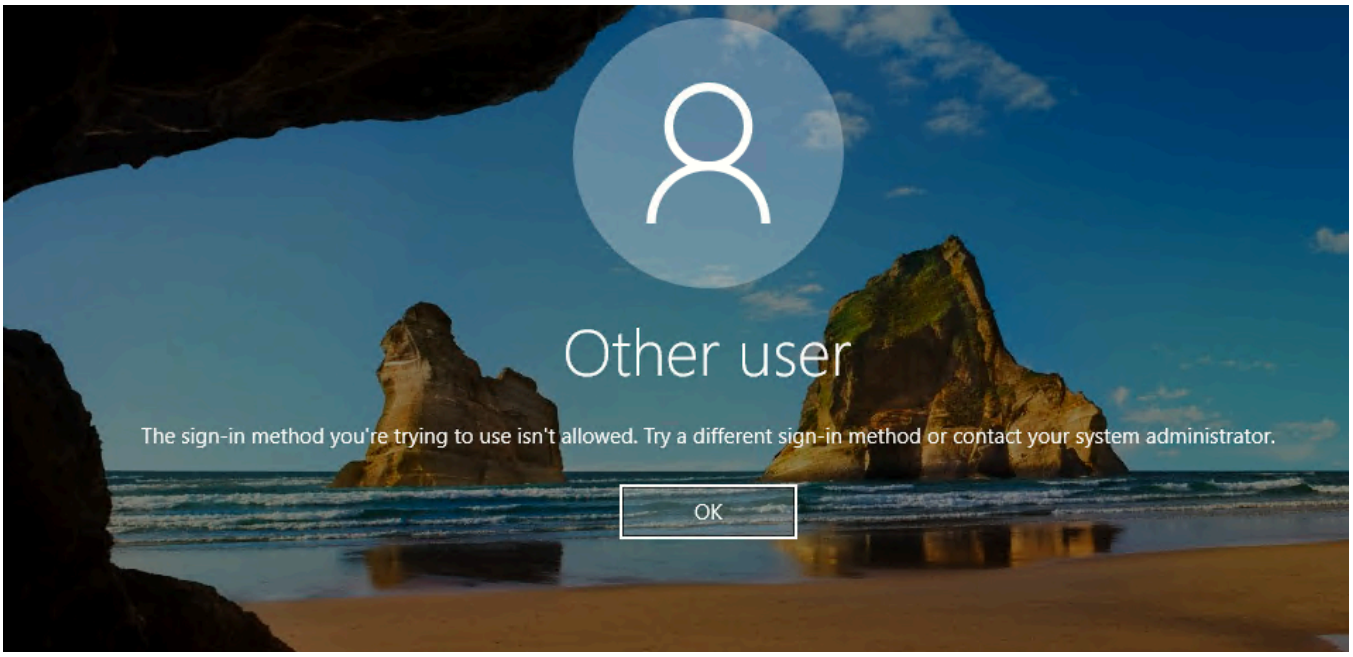
- Navigate to **Advanced System Settings/System Properties** ->** Computer Name** tab -> select the "**Change**" button to rename the computer -> select "**More**..." under the existing computer name -> Type in your domain name and select **OK** -> Save and reboot.
- Once it's done, we can RDP with FQDN directly and no need to modify the HOSTS entry.

> ⓘ **Note**

In such case, when Primary DNS Suffix is added, the Device-Sync scheduled task is triggered when adding the FQDN into AAD device "hostnames" attributes. This is why it resolves the issue.

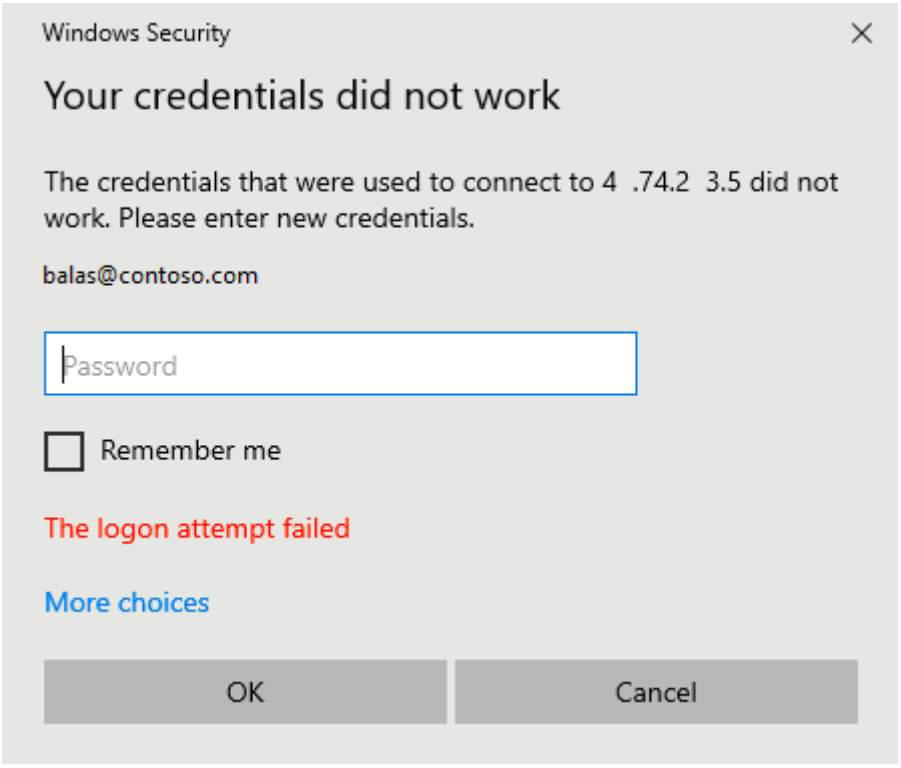
MFA sign-in method required

You might see the following error message when you initiate a remote desktop connection to your device: "The sign-in method you're trying to use isn't allowed. Try a different sign-in method or contact your system administrator."



If you configure a Conditional Access policy that requires MFA, you need to ensure that the device initiating the connection uses strong authentication such as Windows Hello.

Another MFA-related error message is the one described previously: "Your credentials didn't work."



If you configure a legacy per-user **Enabled/Enforced Microsoft Entra multifactor authentication** setting and you see the error, you can resolve the problem by removing the per-user MFA setting. For more information, see the article Enable per-user Microsoft Entra multifactor authentication to secure sign-in events.

If Windows Hello for Business isn't an option, configure a Conditional Access policy that excludes the Microsoft Azure Windows Virtual Machine Sign-in app. To learn more about Windows Hello for Business, see Windows Hello for Business overview.

Support for biometric authentication with RDP was added in Windows 10 version 1809. Using Windows Hello for Business authentication during RDP is available for deployments that use a certificate trust model or key trust model.

Share your feedback about this feature or report problems with using it on the Microsoft Entra feedback forum.

Missing application

If the Microsoft Azure Windows Virtual Machine Sign-in application is missing from Conditional Access, make sure that the application is in the tenant:

1. Sign in to the Microsoft Entra admin center as at least a Cloud Application Administrator.
2. Browse to **Entra ID** > **Enterprise apps**.

3. Remove the filters to see all applications, and search for **VM**. If you don't see **Microsoft Azure Windows Virtual Machine Sign-in** as a result, the service principal is missing from the tenant.

Another way to verify it is via Graph PowerShell:

1. Install the Graph PowerShell SDK.
2. Run `Connect-MgGraph -Scopes "ServicePrincipalEndpoint.ReadWrite.All"`, followed by `"Application.ReadWrite.All"`.
3. Sign in with a Global Administrator account.
4. Consent to the permission prompt.
5. Run `Get-MgServicePrincipal -ConsistencyLevel eventual -Search '"DisplayName:Azure Windows VM Sign-In"'`.

    - If this command results in no output and returns you to the PowerShell prompt, you can create the service principal with the following Graph PowerShell command:

      ```
      New-MgServicePrincipal -AppId 372140e0-b3b7-4226-8ef9-d57986796201
      ```

    - Successful output shows that the Microsoft Azure Windows Virtual Machine Sign-in app and its ID were created.

6. Sign out of Graph PowerShell by using the `Disconnect-MgGraph` command.

Some tenants might see the application named Azure Windows VM Sign-in instead of Microsoft Azure Windows Virtual Machine Sign-in. The application has the same Application ID of 372140e0-b3b7-4226-8ef9-d57986796201.

Unable to use this capability when require compliant device Conditional Access policy is enforced on Azure Windows VM Sign-in resource and you're connecting from a Windows Server device

Windows Server device compliance configuration in Conditional Access policy isn't supported.

Next steps

For more information about Microsoft Entra ID, see What is Microsoft Entra ID?.

Last updated on 04/02/2026

Guest updates and host maintenance overview

Article • 09/27/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

This article provides an overview of guest update and host maintenance options for Azure virtual machines (VMs).

Azure periodically updates its infrastructure to improve reliability, performance, and security or to launch new features. Most updates are transparent to you. To incorporate these updates, Azure uses a robust infrastructure that includes region pairs and availability zones combined with multiple tools and features.

Azure also offers you the ability to control updates on Azure machines like virtual machine scale sets, host machines, guest VMs, and extensions attached to VMs. This control is possible through maintenance configurations, which you can use to set up recurring schedules for times when you want available platform updates to occur.

With Azure infrastructure updates, you can:

- Upgrade network components.
- Decommission hardware on networks.
- Patch software components in hosting environments.
- Update guest operating system (OS) software on VMs.

To perform these updates, use the tools that are available in Azure.

The maintenance platform aims to provide you with a *unified maintenance experience* for all Azure resources that are affected during maintenance. The maintenance experience is available for host (Azure Dedicated Host and isolated VMs) and guest (VMs and Azure Arc VMs) resources, Azure Kubernetes Service, Salesforce Marketing Cloud (SFMC), and network gateways (Azure VPN Gateway, Azure ExpressRoute, and virtual network gateways in Azure). To deploy, use either the Azure portal, PowerShell, or the Azure CLI.

Maintenance control provides you with an option to skip or defer certain updates and schedule them only during your preferred maintenance window. In exceptional circumstances where a high-severity security issue arises that could potentially compromise customer services, Azure reserves the right to bypass these schedules to implement urgent changes. This measure is taken solely to help ensure the safety and

integrity of customer services. It's employed only when the updates have no effect on customer resources. Such instances are infrequent and are invoked only as a last resort to safeguard resources.

Host maintenance

Host maintenance is performed on the physical hosts where VMs are located and is usually transparent to you. Some updates might have an effect on hosts that you can tolerate. During these updates, the VMs that are allocated on the hosts might freeze (*non-rebootful updates*), reboot (*rebootful updates*), or migrate live to another updated host. Azure chooses the update mechanism that affects your VMs the least.

Dedicated hosts, isolated VMs, and shared hosts

The host maintenance experience is available for dedicated hosts, isolated VMs, and shared hosts. Dedicated hosts are hosts in which one customer owns all VMs. Shared hosts are hosts in which VMs from multiple customers reside together. Isolated VMs are large machines that are isolated to a specific hardware type and dedicated to a single customer.

On dedicated hosts, you have the host maintenance experience available for all updates. You can opt into maintenance control and schedule a maintenance window based on your needs within 35 days from the last maintenance date. Isolated VMs have the maintenance control experience available like dedicated hosts.

You can use maintenance control to:

- Apply all updates together.
- Wait up to 35 days to apply updates for host machines.
- Set up a maintenance schedule or use Azure Functions to automate platform updates.
- Configure maintenance across subscriptions and resource groups.

On shared hosts, the maintenance experience is available for rebootful updates or high-impact updates. Currently, the maintenance control experience isn't available for updates that take less than 30 seconds.

Maintenance notifications

Azure provides notifications before, during, and after maintenance operations. Scheduled events provide notifications before an event starts and while it's in progress so that your application can react automatically. Flash Health events provide information

that you can consume to analyze alerts and trends in VM availability for reporting and root cause analysis.

Scheduled events

Scheduled events provide advance notification of upcoming availability impacts so that you can prepare your application ahead of time. They're delivered directly to the affected VM and to all VMs in the same placement group for automated resilience. For information on scheduled events, see Scheduled events for Windows VMs and Scheduled events for Linux.

Flash Health events

Flash Health events provide near real-time information about past availability impacts so that you can react to events and easily mitigate incidents. Flash information is available in Azure Monitor, Azure Resource Graph, or Azure Event Grid to integrate with your systems and processes.

Guest updates

This section explains guest update options.

OS image upgrade

Automatic OS upgrades are available for virtual machine scale sets. An upgrade works by replacing the OS disk of a VM with a new disk that was created by using the latest image version. Any configured extensions and custom data scripts are run on the OS disk, while data disks are retained. To minimize the application downtime, upgrades take place in batches. No more than 20% of the scale set upgrades at any time.

Maintenance control is also available for OS image upgrades. You can opt in to this experience by using maintenance configurations to schedule when to apply these image upgrades. To use this experience, scale sets need to have automatic OS upgrades enabled. You can schedule recurrence for up to a week (seven days). A minimum of five hours is required for the maintenance window.

Guest VM patching

Automatic VM guest patching is integrated with Azure Update Manager. You can save recurring deployment schedules to install updates for your Windows Server and Linux

machines in Azure, in on-premises environments, and in other cloud environments that are connected by using Azure Arc-enabled servers.

Guest extension upgrades

Automatic Extension Upgrade is available for Azure Virtual Machines and Azure Virtual Machine Scale Sets. When Automatic Extension Upgrade is enabled on a VM or scale set, the extension is automatically upgraded whenever the extension publisher releases a new version for that extension. The extension upgrade process replaces the existing extension version on a VM with the new version of the same extension.

The health of the VM is monitored after the new extension is installed. If the VM isn't in a healthy state within five minutes of the upgrade completion, the extension version rolls back to the previous version. Maintenance control on extensions is currently available only via the CLI and PowerShell. You can schedule recurrence for up to a week (seven days). A minimum of five hours is required for the maintenance window.

Hotpatch

Hotpatching is a new way to install updates on new Windows Server Azure Edition VMs that doesn't require a reboot after installation. Hotpatching for Windows Server Azure Edition VMs has the following benefits:

- Lower workload effect with fewer reboots.
- Faster deployment of updates because the packages are smaller, install faster, and have easier patch orchestration with Update Manager.
- Better protection because the hotpatch update packages are scoped to Windows security updates that install faster without rebooting.

Azure update management

You can use Update Manager in Azure Automation to manage system updates for your Windows and Linux VMs in Azure, in on-premises environments, and in other cloud environments. You can quickly assess the status of available updates on all agent machines and manage the process of installing required updates for servers.

Update Manager

Update Manager is a unified service in Azure that helps you manage and govern updates (Windows and Linux), both on-premises and with other cloud platforms, across hybrid environments from a single dashboard. The new functionality provides a native

and an out-of-the-box experience with granular access controls. You have the flexibility to create schedules or act now and the ability to check updates automatically. The enhanced functionality ensures that administrators have visibility into the health of all systems in the environment. For more information, see Key benefits.

Related content

For more ways to increase the uptime of your applications and services, see Availability options for Azure virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Virtual Machine Scale Set automatic OS image upgrades

Article • 01/17/2025

> **Note**
>
> Many of the steps listed in this document apply to Virtual Machine Scale Sets using Uniform Orchestration mode. We recommend using Flexible Orchestration for new workloads. For more information, see Orchestration modes for Virtual Machine Scale Sets in Azure.

Enabling automatic OS image upgrades on your scale set helps ease update management by safely and automatically upgrading the OS disk for all instances in the scale set.

Automatic OS upgrade has the following characteristics:

- Once configured, the latest OS image published by image publishers is automatically applied to the scale set without user intervention.
- Upgrades batches of instances in a rolling manner each time a new image is published by the publisher.
- Integrates with application health probes and Application Health extension.
- Works for all virtual machine sizes, and for both Windows and Linux images including custom images through Azure Compute Gallery.
- You can opt out of automatic upgrades at any time (OS Upgrades can be initiated manually as well).
- The OS Disk of a virtual machine is replaced with the new OS Disk created with latest image version. Configured extensions and custom data scripts are run, while persisted data disks are retained.
- Extension sequencing is supported.
- Can be enabled on a scale set of any size.

> **Note**
>
> Before enabling automatic OS image upgrades, check requirements section of this documentation.

How does automatic OS image upgrade work?

An upgrade works by replacing the OS disk of a virtual machine with a new disk created using the image version. Any configured extensions and custom data scripts are run on the OS disk, while data disks are retained. To minimize the application downtime, upgrades take place in batches, with no more than 20% of the scale set upgrading at any time.

You must integrate an Azure Load Balancer application health probe or Application Health extension to track the health of the application after an upgrade. This allows the platform to validate the virtual machine health to ensure updates are applied in a safe manner. We recommend incorporating an application heartbeat to validate upgrade success.

Availability-first Updates

The availability-first model for platform orchestrated updates described below ensures that availability configurations in Azure are respected across multiple availability levels.

Across regions:

- An update moves across Azure globally in a phased manner to prevent Azure-wide deployment failures.
- A 'phase' can have one or more regions, and an update moves across phases only if eligible virtual machines in the previous phase update successfully.
- Geo-paired regions won't be updated concurrently and cannot be in the same regional phase.
- The success of an update is measured by tracking the health of a virtual machine post update.

Within a region:

- Virtual machines in different Availability Zones are not updated concurrently with the same update.

Within a 'set':

- All virtual machines in a common scale set are not updated concurrently.
- Virtual machines in a common Virtual Machine Scale Set are grouped in batches and updated within Update Domain boundaries as described below.

The platform orchestrated updates process is followed for rolling out supported OS platform image upgrades every month. For custom images through Azure Compute Gallery, an image upgrade is only kicked off for a particular Azure region when the new image is published and replicated to the region of that scale set.

Upgrading virtual machines in a scale set

The region of a scale set becomes eligible to get image upgrades either through the availability-first process for platform images or replicating new custom image versions for Share Image Gallery. The image upgrade is then applied to an individual scale set in a batched manner as follows:

1. Before you begin the upgrade process, the orchestrator ensures that no more than 20% of instances in the entire scale set are unhealthy (for any reason).
2. The upgrade orchestrator identifies the batch of virtual machines to upgrade, with any one batch having a maximum of 20% of the total instance count, subject to a minimum batch size of one virtual machine. There is no minimum scale set size requirement and scale sets with 5 or fewer instances have 1 virtual machine per upgrade batch (minimum batch size).
3. The OS disk of every virtual machine in the selected upgrade batch is replaced with a new OS disk created from the image. All specified extensions and configurations in the scale set model are applied to the upgraded instance.
4. For scale sets with configured application health probes or Application Health extension, the upgrade waits up to 5 minutes for the instance to become healthy, before moving on to upgrade the next batch. If an instance does not recover its health in 5 minutes after an upgrade, then by default the previous OS disk for the instance is restored.
5. The upgrade orchestrator also tracks the percentage of instances that become unhealthy post an upgrade. The upgrade stops if more than 20% of upgraded instances become unhealthy during the upgrade process.
6. The above process continues until all instances in the scale set have been upgraded.

The scale set OS upgrade orchestrator checks for the overall scale set health before upgrading every batch. While you're upgrading a batch, there could be other concurrent planned or unplanned maintenance activities that could impact the health of your scale set instances. In such cases if more than 20% of the scale set's instances become unhealthy, then the scale set upgrade stops at the end of current batch.

To modify the default settings associated with Rolling Upgrades, review Azure's Rolling Upgrade Policy.

> ⓘ **Note**
>
> Automatic OS upgrade does not upgrade the reference image Sku on the scale set. To change the Sku (such as Ubuntu 18.04-LTS to 20.04-LTS), you must update the

scale set model directly with the desired image Sku. Image publisher and offer can't be changed for an existing scale set.

OS image upgrade versus reimage

Both **OS Image Upgrade** and **Reimage** are methods used to update virtual machines within a scale set, but they serve different purposes and have distinct impacts.

OS image upgrade involves updating the underlying operating system image that is used to create new instances in a scale set. When you perform an OS image upgrade, Azure creates new virtual machines with the updated OS image and gradually replace the old virtual machines in the scale set with the new ones. This process is typically performed in stages to ensure high availability. OS image upgrades are a non-disruptive way to apply updates or changes to the underlying OS of the virtual machines in a scale set. Existing virtual machines are not affected until they are replaced with the new instances.

Reimaging a virtual machine in a scale set is a more immediate and disruptive action. When you choose to reimage a virtual machine , Azure stops the selected virtual machine, perform the reimage operation, and then restart the virtual machine using the same OS image. This effectively reinstalls the OS on that specific virtual machine. Reimaging is typically used when you need to troubleshoot or reset a specific virtual machine due to issues with that instance.

Key differences:

- OS Image Upgrade is a gradual and non-disruptive process that updates the OS image for the entire Virtual Machine Scale Set over time, ensuring minimal impact on running workloads.
- Reimage is a more immediate and disruptive action that affects only the selected virtual machine, stopping it temporarily and reinstalling the OS.

When to use each method:

- Use OS Image Upgrade when you want to update the OS image for the entire scale set while maintaining high availability.
- Use Reimage when you need to troubleshoot or reset a specific virtual machine within the virtual Machine Scale Set.

It's essential to carefully plan and choose the appropriate method based on your specific requirements to minimize any disruption to your applications and services running in a Virtual Machine Scale Set.

Supported OS images

Only certain OS platform images are currently supported. Custom images are supported if the scale set uses custom images through Azure Compute Gallery.

The following platform SKUs are currently supported (and more are added periodically):

Expand table

Publisher	OS Offer	Sku
Canonical	UbuntuServer	18.04-LTS
Canonical	UbuntuServer	18_04-LTS-Gen2
Canonical	0001-com-ubuntu-server-focal	20_04-LTS
Canonical	0001-com-ubuntu-server-focal	20_04-LTS-Gen2
Canonical	0001-com-ubuntu-server-jammy	22_04-LTS
Canonical	0001-com-ubuntu-server-jammy	22_04-LTS-Gen2
MicrosoftCblMariner	Cbl-Mariner	cbl-mariner-1
MicrosoftCblMariner	Cbl-Mariner	1-Gen2
MicrosoftCblMariner	Cbl-Mariner	cbl-mariner-2
MicrosoftCblMariner	Cbl-Mariner	cbl-mariner-2-Gen2
MicrosoftSqlServer	Sql2017-ws2019	enterprise
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter
MicrosoftWindowsServer	WindowsServer	2016-Datacenter
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-gensecond
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-gs
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-with-Containers
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-with-containers-gs

Publisher	OS Offer	Sku
MicrosoftWindowsServer	WindowsServer	2019-Datacenter
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-Core
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-Core-with-Containers
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-gensecond
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-gs
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-with-Containers
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-with-Containers-gs
MicrosoftWindowsServer	WindowsServer	2022-Datacenter
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-smalldisk-g2
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-core
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-core-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-g2
MicrosoftWindowsServer	WindowsServer	Datacenter-core-20h2-with-containers-smalldisk-gs
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-azure-edition
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-azure-edition-smalldisk
Mirantis	Windows_with_Mirantis_Container_Runtime_2019	win_2019_mcr_23_0
Mirantis	Windows_with_Mirantis_Container_Runtime_2019	win_2019_mcr_23_0_gen2

Requirements for configuring automatic OS image upgrade

- The *version* property of the image must be set to latest.
- Must use application health probes or Application Health extension for non-Service Fabric scale sets. For Service Fabric requirements, see Service Fabric requirement.
- Use Compute API version 2018-10-01 or higher.
- Ensure that external resources specified in the scale set model are available and updated. Examples include SAS URI for bootstrapping payload in virtual machine extension properties, payload in storage account, reference to secrets in the model, and more.
- For scale sets using Windows virtual machines, starting with Compute API version 2019-03-01, the *virtualMachineProfile.osProfile.windowsConfiguration.enableAutomaticUpdates* property must set to *false* in the scale set model definition. The *enableAutomaticUpdates* property enables in-VM patching where "Windows Update" applies operating system patches without replacing the OS disk. With automatic OS image upgrades enabled on your scale set, which can be done by setting the *automaticOSUpgradePolicy.enableAutomaticOSUpgrade* to *true*, an extra patching process through Windows Update is not required.
- The patch orchestration mode must *not* be set to `AutomaticByPlatform` in the scale set model definition. With automatic OS image upgrades enabled on your scale set, a platform orchestration patching process is not required.

> **Note**
>
> After an OS disk is replaced through reimage or upgrade, the attached data disks may have their drive letters reassigned. To retain the same drive letters for attached disks, it is suggested to use a custom boot script.

Service Fabric requirements

If you are using Service Fabric, ensure the following conditions are met:

- Service Fabric durability level is Silver or Gold. If Service Fabric durability is Bronze, only Stateless-only node types support automatic OS image upgrades).
- The Service Fabric extension on the scale set model definition must have TypeHandlerVersion 1.1 or above.

- Durability level should be the same at the Service Fabric cluster and Service Fabric extension on the scale set model definition.
- More health probes or use of application health extension is not required for Silver or Gold durability. Bronze durability with Stateless-only node types requires an additional health probe.
- The property *virtualMachineProfile.osProfile.windowsConfiguration.enableAutomaticUpdates* property must set to *false* in the scale set model definition. The *enableAutomaticUpdates* property enables in-VM patching using "Windows Update" and is not supported on Service Fabric scale sets. You should use the *automaticOSUpgradePolicy.enableAutomaticOSUpgrade* property instead.

Ensure that durability settings are not mismatched on the Service Fabric cluster and Service Fabric extension, as a mismatch results in upgrade errors. Durability levels can be modified per the guidelines outlined on this page.

Automatic OS image upgrade for custom images

Automatic OS image upgrade is supported for custom images deployed through Azure Compute Gallery. Other custom images are not supported for automatic OS image upgrades.

Additional requirements for custom images

- The setup and configuration process for automatic OS image upgrade is the same for all scale sets as detailed in the configuration section of this page.
- Scale sets instances configured for automatic OS image upgrades are upgraded to the version of the Azure Compute Gallery image when a new version of the image is published and replicated to the region of that scale set. If the new image is not replicated to the region where the scale is deployed, the scale set instances are not upgraded to the version. Regional image replication allows you to control the rollout of the new image for your scale sets.
- The new image version should not be excluded from the version for that gallery image. Image versions excluded from the gallery image's version are not rolled out to the scale set through automatic OS image upgrade.

> **Note**

It can take up to 3 hours for a scale set to trigger the first image upgrade rollout after the scale set is first configured for automatic OS upgrades due to certain factors such as Maintenance Windows or other restrictions. Customers on the latest image may not get an upgrade until a new image is available.

Configure automatic OS image upgrade

To configure automatic OS image upgrade, ensure that the *automaticOSUpgradePolicy.enableAutomaticOSUpgrade* property is set to *true* in the scale set model definition.

> ⓘ Note
>
> **Upgrade Policy mode** and **Automatic OS Upgrade Policy** are separate settings and control different aspects of the scale set. When there are changes in the scale set template, the Upgrade Policy `mode` determines what happens to existing instances in the scale set. However, Automatic OS Upgrade Policy `enableAutomaticOSUpgrade` is specific to the OS image and tracks changes the image publisher has made and determines what happens when there is an update to the image.

> ⓘ Note
>
> If `enableAutomaticOSUpgrade` is set to *true*, `enableAutomaticUpdates` is automatically set to *false* and cannot be set to *true*.

REST API

The following example describes how to set automatic OS upgrades on a scale set model:

```
PUT or PATCH on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Mic
rosoft.Compute/virtualMachineScaleSets/myScaleSet?api-version=2021-03-01`
```

JSON

```
{
  "properties": {
```

```
    "upgradePolicy": {
      "automaticOSUpgradePolicy": {
        "enableAutomaticOSUpgrade":  true
      }
    }
  }
}
```

Azure PowerShell

Use the New-AzVmss cmdlet to configure automatic OS image upgrades for your scale set during provisioning. The following example configures automatic upgrades for the scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure PowerShell

```
New-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "myScaleSet"
-AutomaticOSUpgrade $true
```

Use the Update-AzVmss cmdlet to configure automatic OS image upgrades for your existing scale set. The following example configures automatic upgrades for the scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure PowerShell

```
Update-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName
"myScaleSet" -AutomaticOSUpgrade $true
```

Azure CLI 2.0

Use az vmss create to configure automatic OS image upgrades for your scale set during provisioning. Use Azure CLI 2.0.47 or above. The following example configures automatic upgrades for the scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure CLI

```
az vmss create --name myScaleSet --resource-group myResourceGroup --enable-
auto-os-upgrade true --upgrade-policy-mode Rolling
```

Use az vmss update to configure automatic OS image upgrades for your existing scale set. Use Azure CLI 2.0.47 or above. The following example configures automatic

upgrades for the scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure CLI

```
az vmss update --name myScaleSet --resource-group myResourceGroup --enable-auto-os-upgrade true --upgrade-policy-mode Rolling
```

> **Note**
>
> After configuring automatic OS image upgrades for your scale set, you must also bring the scale set virtual machines to the latest scale set model if your scale set uses the 'Manual' upgrade policy.

ARM templates

The following example describes how to set automatic OS upgrades on a scale set model via Azure Resource Manager templates (ARM templates):

JSON

```
"properties": {
   "upgradePolicy": {
     "mode": "Automatic",
     "RollingUpgradePolicy": {
         "BatchInstancePercent": 20,
         "MaxUnhealthyInstancePercent": 25,
         "MaxUnhealthyUpgradedInstancePercent": 25,
         "PauseTimeBetweenBatches": "PT0S"
     },
    "automaticOSUpgradePolicy": {
      "enableAutomaticOSUpgrade": true,
        "useRollingUpgradePolicy": true,
        "disableAutomaticRollback": false
    }
  },
  },
"imagePublisher": {
   "type": "string",
   "defaultValue": "MicrosoftWindowsServer"
 },
 "imageOffer": {
   "type": "string",
   "defaultValue": "WindowsServer"
 },
 "imageSku": {
   "type": "string",
   "defaultValue": "2022-datacenter"
```

```
  },
  "imageOSVersion": {
    "type": "string",
    "defaultValue": "latest"
  }
```

Bicep

The following example describes how to set automatic OS upgrades on a scale set model via Bicep:

JSON

```
properties: {
    overprovision: overProvision
    upgradePolicy: {
      mode: 'Automatic'
      automaticOSUpgradePolicy: {
        enableAutomaticOSUpgrade: true
      }
    }
}
```

Using Application Health Extension

During an OS Upgrade, virtual machines in a scale set are upgraded one batch at a time. The upgrade should continue only if the customer application is healthy on the upgraded virtual machines. We recommend that the application provides health signals to the scale set OS Upgrade engine. By default, during OS Upgrades the platform considers virtual machine power state and extension provisioning state to determine if a virtual machine is healthy after an upgrade. During the OS Upgrade of a virtual machine, the OS disk on a virtual machine is replaced with a new disk based on latest image version. After the OS Upgrade has completed, the configured extensions are run on these virtual machines. The application is considered healthy only when all the extensions on the instance are successfully provisioned.

A scale set can optionally be configured with Application Health Probes to provide the platform with accurate information on the ongoing state of the application. Application Health Probes are Custom Load Balancer Probes that are used as a health signal. The application running on a scale set virtual machine can respond to external HTTP or TCP requests indicating whether it's healthy. For more information on how Custom Load Balancer Probes work, see to Understand load balancer probes. Application Health

Probes are not supported for Service Fabric scale sets. Non-Service Fabric scale sets require either Load Balancer application health probes or Application Health extension.

If the scale set is configured to use multiple placement groups, probes using a Standard Load Balancer need to be used.

> ⓘ Note
>
> Only one source of health monitoring can be used for a Virtual Machine Scale Set, either an Application Health Extension or a Health Probe. If you have both options enabled, you need to remove one before using orchestration services like Instance Repairs or Automatic OS Upgrades.

Configuring a Custom Load Balancer Probe as Application Health Probe on a scale set

As a best practice, create a load balancer probe explicitly for scale set health. The same endpoint for an existing HTTP probe or TCP probe can be used, but a health probe could require different behavior from a traditional load-balancer probe. For example, a traditional load balancer probe could return unhealthy if the load on the instance is too high, but that would not be appropriate for determining the instance health during an automatic OS upgrade. Configure the probe to have a high probing rate of less than two minutes.

The load-balancer probe can be referenced in the *networkProfile* of the scale set and can be associated with either an internal or public facing load-balancer as follows:

JSON

```
"networkProfile": {
  "healthProbe" : {
    "id": "[concat(variables('lbId'), '/probes/',
variables('sshProbeName'))]"
  },
  "networkInterfaceConfigurations":
  ...
}
```

> ⓘ Note
>
> When using Automatic OS Upgrades with Service Fabric, the new OS image is rolled out Update Domain by Update Domain to maintain high availability of the services

running in Service Fabric. To utilize Automatic OS Upgrades in Service Fabric your cluster node type must be configured to use the Silver Durability Tier or higher. For Bronze Durability tier, automatic OS image upgrade is only supported for Stateless node types. For more information on the durability characteristics of Service Fabric clusters, see **this documentation**.

Using Application Health extension

The Application Health extension is deployed inside a Virtual Machine Scale Set instance and reports on virtual machine health from inside the scale set instance. You can configure the extension to probe on an application endpoint and update the status of the application on that instance. This instance status is checked by Azure to determine whether an instance is eligible for upgrade operations.

As the extension reports health from within a virtual machine, the extension can be used in situations where external probes such as Application Health Probes (that utilize custom Azure Load Balancer probes) can't be used.

There are multiple ways of deploying the Application Health extension to your scale sets as detailed in the examples in this article.

> ⓘ Note
>
> Only one source of health monitoring can be used for a Virtual Machine Scale Set, either an Application Health Extension or a Health Probe. If you have both options enabled, you need to remove one before using orchestration services like Instance Repairs or Automatic OS Upgrades.

Configure custom metrics for rolling upgrades on Virtual Machine Scale Sets (Preview)

> ⓘ Note
>
> **Custom metrics for rolling upgrades on Virtual Machine Scale Sets is currently in preview.** Previews are made available to you on the condition that you agree to the supplemental terms of use. Some aspects of these features may change prior to general availability (GA).

Custom metrics for rolling upgrades enables you to utilize the application health extension to emit custom metrics to your Virtual Machine Scale Set. These custom metrics can be used to tell the scale set the order in which virtual machines should be updated when a rolling upgrade is triggered. The custom metrics can also inform your scale set when an upgrade should be skipped on a specific instance. This allows you to have more control over the ordering and the update process itself.

Custom metrics can be used in combination with other rolling upgrade functionality such as automatic OS upgrades, automatic extension upgrades and MaxSurge rolling upgrades.

For more information, see custom metrics for rolling upgrades on Virtual Machine Scale Sets

Get the history of automatic OS image upgrades

You can check the history of the most recent OS upgrade performed on your scale set with Azure PowerShell, Azure CLI 2.0, or the REST APIs. You can get history for the last five OS upgrade attempts within the past two months.

Keep credentials up to date

If your scale set uses any credentials to access external resources, such as a virtual machine extension configured to use a SAS token for storage account, then ensure that the credentials are updated. If any credentials including certificates and tokens have expired, the upgrade fails and the first batch of virtual machines are left in a failed state.

The recommended steps to recover virtual machines and re-enable automatic OS upgrade if there's a resource authentication failure are:

- Regenerate the token (or any other credentials) passed into your extensions.
- Ensure that any credential used from inside the virtual machine to talk to external entities is up to date.
- Update extensions in the scale set model with any new tokens.
- Deploy the updated scale set, which updates all virtual machines including the failed ones.

REST API

The following example uses REST API to check the status for the scale set named *myScaleSet* in the resource group named *myResourceGroup*:

```
GET on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Mic
rosoft.Compute/virtualMachineScaleSets/myScaleSet/osUpgradeHistory?api-
version=2021-03-01`
```

The GET call returns properties similar to the following example output:

JSON

```
{
    "value": [
        {
            "properties": {
        "runningStatus": {
          "code": "RollingForward",
          "startTime": "2018-07-24T17:46:06.1248429+00:00",
          "completedTime": "2018-04-21T12:29:25.0511245+00:00"
        },
        "progress": {
          "successfulInstanceCount": 16,
          "failedInstanceCount": 0,
          "inProgressInstanceCount": 4,
          "pendingInstanceCount": 0
        },
        "startedBy": "Platform",
        "targetImageReference": {
          "publisher": "MicrosoftWindowsServer",
          "offer": "WindowsServer",
          "sku": "2016-Datacenter",
          "version": "2016.127.20180613"
        },
        "rollbackInfo": {
          "successfullyRolledbackInstanceCount": 0,
          "failedRolledbackInstanceCount": 0
        }
      },
      "type": "Microsoft.Compute/virtualMachineScaleSets/rollingUpgrades",
      "location": "westeurope"
    }
  ]
}
```

Azure PowerShell

Use the Get-AzVmss cmdlet to check OS upgrade history for your scale set. The following example details how you review the OS upgrade status for a scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure PowerShell

```
Get-AzVmss -ResourceGroupName "myResourceGroup" -VMScaleSetName "myScaleSet"
-OSUpgradeHistory
```

Azure CLI 2.0

Use az vmss get-os-upgrade-history to check the OS upgrade history for your scale set. Use Azure CLI 2.0.47 or above. The following example details how you review the OS upgrade status for a scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure CLI

```
az vmss get-os-upgrade-history --resource-group myResourceGroup --name
myScaleSet
```

How to get the latest version of a platform OS image?

You can get the available image versions for automatic OS upgrade supported SKUs using the below examples:

REST API

```
GET on
`/subscriptions/subscription_id/providers/Microsoft.Compute/locations/{locat
ion}/publishers/{publisherName}/artifacttypes/vmimage/offers/{offer}/skus/{s
kus}/versions?api-version=2021-03-01`
```

Azure PowerShell

Azure PowerShell

```
Get-AzVmImage -Location "westus" -PublisherName "Canonical" -offer "0001-com-ubuntu-server-jammy" -sku "22_04-lts"
```

Azure CLI 2.0

Azure CLI

```
az vm image list --location "westus" --publisher "Canonical" --offer "0001-com-ubuntu-server-jammy" --sku "22_04-lts" --all
```

Manually trigger OS image upgrades

With automatic OS image upgrade enabled on your scale set, you don't need to manually trigger image updates on your scale set. The OS upgrade orchestrator automatically applies the latest available image version to your scale set instances without any manual intervention.

For specific cases where you don't want to wait for the orchestrator to apply the latest image, you can trigger an OS image upgrade manually using the below examples.

> ⓘ **Note**
>
> Manual trigger of OS image upgrades does not provide automatic rollback capabilities. If an instance does not recover its health after an upgrade operation, its previous OS disk can't be restored.

REST API

Use the Start OS Upgrade API call to start a rolling upgrade to move all Virtual Machine Scale Set instances to the latest available image OS version. Instances that are already running the latest available OS version are not affected. The following example details how you can start a rolling OS upgrade on a scale set named *myScaleSet* in the resource group named *myResourceGroup*:

```
POST on `/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachineScaleSets/myScaleSet/osRollingUpgrade?api-version=2021-03-01`
```

Azure PowerShell

Use the Start-AzVmssRollingOSUpgrade cmdlet to check OS upgrade history for your scale set. The following example details how you can start a rolling OS upgrade on a scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure PowerShell

```
Start-AzVmssRollingOSUpgrade -ResourceGroupName "myResourceGroup" -
VMScaleSetName "myScaleSet"
```

Azure CLI 2.0

Use az vmss rolling-upgrade start to check the OS upgrade history for your scale set. Use Azure CLI 2.0.47 or above. The following example details how you can start a rolling OS upgrade on a scale set named *myScaleSet* in the resource group named *myResourceGroup*:

Azure CLI

```
az vmss rolling-upgrade start --resource-group "myResourceGroup" --name
"myScaleSet" --subscription "subscriptionId"
```

Leverage Activity Logs for Upgrade Notifications and Insights

Activity Log is a subscription log that provides insight into subscription-level events that have occurred in Azure. Customers are able to:

- See events related to operations performed on their resources in Azure portal
- Create action groups to tune notification methods like email, sms, webhooks, or ITSM
- Set up suitable alerts using different criteria using Portal, ARM resource template, PowerShell or CLI to be sent to action groups

Customers receive three types of notifications related to Automatic OS Upgrade operation:

- Submission of upgrade request for a particular resource
- Outcome of submission request along with any error details
- Outcome of upgrade completion along with any error details

Setting up Action Groups for Activity log alerts

An action group is a collection of notification preferences defined by the owner of an Azure subscription. Azure Monitor and Service Health alerts use action groups to notify users that an alert has been triggered.

Action groups can be created and managed using:

- ARM Resource Manager
- Portal
- PowerShell:
  - New-AzActionGroup
  - Get-AzActionGroup
  - Remove-AzActionGroup
- CLI

Customers can set up the following using action groups:

- SMS and/or Email notifications
- Webhooks - Customers can attach webhooks to their automation runbooks and configure their action groups to trigger the runbooks. You can start a runbook from a webhook
- ITSM Connections

Investigate and Resolve Auto Upgrade Errors

The platform can return errors on virtual machines while performing Automatic Image Upgrade with Rolling Upgrade policy. The Get Instance View of a virtual machine contains the detailed error message to investigate and resolve an error. The Rolling Upgrades - Get Latest can provide more details on rolling upgrade configuration and status. The Get OS Upgrade History provides details on the last image upgrade operation on the scale set. Below are the topmost errors that can result in Rolling Upgrades.

RollingUpgradeInProgressWithFailedUpgradedVMs

- Error is triggered for a virtual machine failure.
- The detailed error message mentions whether the rollout continues/pauses based on the configured threshold.

MaxUnhealthyUpgradedInstancePercentExceededInRollingUpgrade

- Error is triggered when the percent of upgraded virtual machines exceed the max threshold allowed for unhealthy virtual machines.
- The detailed error message aggregates the most common error contributing to the unhealthy virtual machines. See MaxUnhealthyUpgradedInstancePercent.

MaxUnhealthyInstancePercentExceededInRollingUpgrade

- Error is triggered when the percent of unhealthy virtual machines exceed the max threshold allowed for unhealthy virtual machines during an upgrade.
- The detailed error message displays the current unhealthy percent and the configured allowable unhealthy virtual machine percentage. See maxUnhealthyInstancePercent.

MaxUnhealthyInstancePercentExceededBeforeRollingUpgrade

- Error is triggered when the percent of unhealthy virtual machines exceed the max threshold allowed for unhealthy virtual machines before an upgrade takes place.
- The detailed error message displays the current unhealthy percent and the configured allowable unhealthy virtual machine percentage. See maxUnhealthyInstancePercent.

InternalExecutionError

- Error is triggered when an unhandled, unformatted or unexpected occurs during execution.
- The detailed error message displays the cause of the error.

RollingUpgradeTimeoutError

- Error is triggered when the rolling upgrade process has timed out.
- The detailed error message displays the length of time the system timed out after attempting to update.

Next steps

Learn about the Application Health Extension

Feedback

Was this page helpful? Yes No

In-place upgrade for VMs running Windows Server in Azure

An in-place upgrade allows you to go from an older operating system to a newer one while keeping your settings, server roles, and data intact. This article teaches you how to move your Azure VMs to a later version of Windows Server using an in-place upgrade. Currently, upgrading to Windows Server 2012, Windows Server 2016, Windows Server 2019, Windows Server 2022, and Windows Server 2025 are supported.

> **⊗ Caution**
>
> Following the process in this article causes a disconnection between the data plane and the control plane of the virtual machine (VM). Azure capabilities such as Auto guest patching, Auto OS image upgrades, Hotpatching, and Azure Update Manager won't be available. This means that the source image information in the VM properties, including the publisher, offer, and plan, remains unchanged. The image used to deploy the VM remains the same, and only the OS is upgraded. To utilize these features, create a new VM using your preferred operating system instead of performing an in-place upgrade.

> **ⓘ Important**
>
> **Virtual Machine (VM)** running in Azure and experiencing Windows Update issues? Trying to diagnose and resolve Windows Update or Windows OS Upgrade issues for your Azure Virtual Machine? Try one of the following tools:
>
> - Azure VM Windows Update Error Detection Tool to diagnose specific Windows Update errors
> - Azure VM Windows Servicing Stack Reset Tool to reset the Windows Servicing stack.
> - Azure Virtual Machine (VM) Windows OS Upgrade Assessment Tool to validate the OS upgrade path and any known issues
>
> *If you're experiencing Windows Update or Windows OS Upgrade issues with virtual machines, before contacting support, run these tools.*

Prerequisites

Before you begin an in-place upgrade:

- Review the upgrade requirements for the target operating system:
  - Upgrade options for Windows Server 2012 from Windows Server 2008 (64-bit) or Windows Server 2008 R2
  - Upgrade options for Windows Server 2016 from Windows Server 2012 or Windows Server 2012 R2
  - Upgrade options for Windows Server 2019 from Windows Server 2012 R2 or Windows Server 2016
  - Upgrade options for Windows Server 2022 from Windows Server 2016 or Windows Server 2019
  - Upgrade options for Windows Server 2025 from Windows Server 2022, Windows Server 2019, Windows Server 2016, or Windows Server 2012 R2
- Run the Azure Virtual Machine (VM) Windows OS Upgrade Assessment Tool to validate the OS upgrade path and any known issues.
- Verify the operating system disk has enough free space to perform the in-place upgrade. If more space is needed follow these steps to expand the operating system disk attached to the VM.
- Disable antivirus and anti-spyware software and firewalls. These types of software can conflict with the upgrade process. Re-enable antivirus and anti-spyware software and firewalls after the upgrade is completed.

Upgrade VM to volume license (KMS server activation)

The upgrade media provided by Azure requires the VM to be configured for Windows Server volume licensing. This is the default behavior for any Windows Server VM that was installed from a generalized image in Azure. If the VM was imported into Azure, then it might need to be converted to volume licensing to use the upgrade media provided by Azure. To confirm the VM is configured for volume license activation follow these steps to configure the appropriate KMS client setup key. If the activation configuration was changed, then follow these steps to verify connectivity to Azure KMS service.

Upgrade to Managed Disks

The in-place upgrade process requires the use of Managed Disks on the VM to be upgraded. Most VMs in Azure are using Managed Disks, and retirement for unmanaged disks support was announced in November of 2022. If the VM is currently using unmanaged disks, then follow these steps to migrate to Managed Disks.

Create snapshot of the operating system disk

We recommend that you create a snapshot of your operating system disk and any data disks before starting the in-place upgrade process. This enables you to revert to the previous state of the VM if anything fails during the in-place upgrade process. To create a snapshot on each disk, follow these steps to create a snapshot of a disk.

> **Note**
>
> During the upgrade process, the upgrade media disk is only created in the `en-US` language. Currently, no other languages or editions are supported. To avoid errors caused by previously using a different language ISO to install the OS, you can set the system language to `en-US` or change the system locale to `English (United States)` in Control Panel.

Create upgrade media disk

To start an in-place upgrade the upgrade media must be attached to the VM as a Managed Disk. To create the upgrade media, modify the variables in the following PowerShell script for Windows Server 2025. The upgrade media disk can be used to upgrade multiple VMs, but it can only be used to upgrade a single VM at a time. To upgrade multiple VMs simultaneously multiple upgrade disks must be created for each simultaneous upgrade.

Expand table

Parameter	Definition
resourceGroup	Name of the resource group where the upgrade media Managed Disk will be created. The named resource group is created if it doesn't exist.
location	Azure region where the upgrade media Managed Disk is created. This must be the same region as the VM to be upgraded.
zone	Azure zone in the selected region where the upgrade media Managed Disk will be created. This must be the same zone as the VM to be upgraded. For regional VMs (nonzonal) the zone parameter should be "".

Parameter	Definition
diskName	Name of the Managed Disk that will contain the upgrade media
sku	Windows Server upgrade media version. This must be either: `server2025Upgrade` or `server2022Upgrade` or `server2019Upgrade` or `server2016Upgrade` or `server2012Upgrade`. The upgrade media disk is created using the latest version of the specified SKU.

If you have more than one subscription, you should run `Set-AzContext -Subscription '<subscription name or id>` to specify which subscription to use.

PowerShell script

Azure PowerShell

```
#
# Customer specific parameters


# Resource group of the source VM
$resourceGroup = "WindowsServerUpgrades"

# Location of the source VM
$location = "WestUS2"

# Zone of the source VM, if any
$zone = ""

# Disk name for the that will be created
$diskName = "WindowsServer2025UpgradeDisk"

# Target version for the upgrade - must be one of these five strings:
server2025Upgrade, server2022Upgrade, server2019Upgrade, server2016Upgrade or
server2012Upgrade
$sku = "server2025Upgrade"


# Common parameters

$publisher = "MicrosoftWindowsServer"
$offer = "WindowsServerUpgrade"
$managedDiskSKU = "Standard_LRS"

#
# Get the latest version of the special (hidden) VM Image from the Azure Marketplace

$versions = Get-AzVMImage -PublisherName $publisher -Location $location -Offer
$offer -Skus $sku | sort-object -Descending {[version] $_.Version    }
$latestString = $versions[0].Version
```

```
# Get the special (hidden) VM Image from the Azure Marketplace by version - the
image is used to create a disk to upgrade to the new version


$image = Get-AzVMImage -Location $location `
                       -PublisherName $publisher `
                       -Offer $offer `
                       -Skus $sku `
                       -Version $latestString

#
# Create Resource Group if it doesn't exist
#

if (-not (Get-AzResourceGroup -Name $resourceGroup -ErrorAction SilentlyContinue)) {
    New-AzResourceGroup -Name $resourceGroup -Location $location
}

#
# Create Managed Disk from LUN 0
#

if ($zone){
    $diskConfig = New-AzDiskConfig -SkuName $managedDiskSKU `
                                   -CreateOption FromImage `
                                   -Zone $zone `
                                   -Location $location
} else {
    $diskConfig = New-AzDiskConfig -SkuName $managedDiskSKU `
                                   -CreateOption FromImage `
                                   -Location $location
}

Set-AzDiskImageReference -Disk $diskConfig -Id $image.Id -Lun 0

New-AzDisk -ResourceGroupName $resourceGroup `
           -DiskName $diskName `
           -Disk $diskConfig
```

Attach upgrade media to the VM

Attach the upgrade media for the target Windows Server version to the VM which will be upgraded. This can be done while the VM is in the running or stopped state.

Portal instructions

1. Sign in to the Azure portal.

2. Search for and select **Virtual machines**.

3. Select a virtual machine to perform the in-place upgrade from the list.
4. On the **Virtual machine** page, select **Disks**.
5. On the **Disks** page, select **Attach existing disks**.
6. In the drop-down for **Disk name**, select the name of the upgrade disk created in the previous step.
7. Select **Save** to attach the upgrade disk to the VM.

Perform in-place upgrade to Windows Server 2016, 2019, 2022, or 2025

To initiate the in-place upgrade the VM must be in the `Running` state. Once the VM is in a running state use the following steps to perform the upgrade.

1. Connect to the VM using RDP or RDP-Bastion.
2. Determine the drive letter for the upgrade disk (typically E: or F: if there are no other data disks).
3. Start Windows PowerShell.
4. Change directory to the only directory on the upgrade disk.
5. Execute the following command to start the upgrade:

   PowerShell

   ```
   .\setup.exe /auto upgrade /dynamicupdate disable
   ```

   You can use /eula accept switch in the Windows Server upgrade command to automatically accept the Microsoft Software License Terms (End User License Agreement or EULA) during the upgrade process. Using the /eula accept switch can help avoid issues where the upgrade process stalls because the EULA wasn't accepted manually. This switch ensures that the upgrade process can proceed smoothly without requiring user interaction to accept the license terms.

   PowerShell

   ```
   .\setup.exe /auto upgrade /dynamicupdate disable /eula accept
   ```

6. Select the correct "Upgrade to" image based on the current version and configuration of the VM using the Windows Server upgrade matrix.

During the upgrade process, the VM will automatically disconnect from the RDP session. After the VM is disconnected from the RDP session the progress of the upgrade can be monitored through the screenshot functionality available in the Azure portal.

Perform in-place upgrade to Windows Server 2012 only

To initiate the in-place upgrade the VM must be in the `Running` state. Once the VM is in a running state use the following steps to perform the upgrade.

1. Connect to the VM using RDP or RDP-Bastion.
2. Determine the drive letter for the upgrade disk (typically E: or F: if there are no other data disks).
3. Start Windows PowerShell.
4. Change directory to the only directory on the upgrade disk.
5. Execute the following command to start the upgrade:

   PowerShell

   ```
   .\setup.exe
   ```

6. When Windows Setup launches, select **Install now**.
7. For **Get important updates for Windows Setup**, select **No thanks**.
8. Select the correct Windows Server 2012 "Upgrade to" image based on the current version and configuration of the VM using the Windows Server upgrade matrix.
9. On the **License terms** page, select **I accept the license terms** and then select **Next**.
10. For **What type of installation do you want?" select **Upgrade: Install Windows and keep files, settings, and applications**.
11. Setup will product a **Compatibility report**, you can ignore any warnings and select **Next**.
12. When complete, the machine reboots, automatically disconnecting you from the RDP session. After the VM is disconnected from the RDP session the progress of the upgrade can be monitored through the screenshot functionality available in the Azure portal.

Post upgrade steps

Once the upgrade process has completed successfully the following steps should be taken to clean up any artifacts which were created during the upgrade process:

- Delete the snapshots of the OS disk and data disk(s) if they were created.
- Delete the upgrade media Managed Disk.
- Enable any antivirus, anti-spyware, or firewall software that may have been disabled at the start of the upgrade process.

> **Important**
>
> The image plan information will not change after the upgrade process.

Recover from failure

If the in-place upgrade process failed to complete successfully you can return to the previous version of the VM if snapshots of the operating system disk and data disk(s) were created. To revert the VM to the previous state using snapshots complete the following steps:

1. Create a new Managed Disk from the OS disk snapshot and each data disk snapshot following the steps in Create a disk from a snapshot making sure to create the disks in the same Availability Zone as the VM if the VM is in a zone.
2. Stop the VM.
3. Swap the OS disk of the VM.
4. Detach any data disks from the VM.
5. Attach data disks created from the snapshots in step 1.
6. Restart the VM.

Next steps

- For more information, see Perform an in-place upgrade of Windows Server
- For information about using Azure Migrate to upgrade, see Azure Migrate Windows Server upgrade

Last updated on 12/08/2025

Azure Virtual Machine (VM) Windows OS Upgrade assessment tool

Applies to: ✔ Windows VMs

Summary

This article provides information about the Azure Virtual Machine (VM) Windows OS Upgrade Assessment tool, a PowerShell script designed to assess the readiness of a Windows machine (desktop or server) for an **in-place OS upgrade**, with special considerations for **Azure VMs**. The tool evaluates OS version, supported upgrade paths, system disk space, and Azure security features like **Trusted Launch**, **Secure Boot**, and **virtual trusted platform modules (vTPMs)**.

Features

- **OS version detection**
  - Identifies whether the system is Windows 10, 11, or a Windows Server edition.
- **Server upgrade path check**
  - Matches current server version to supported upgrade targets.
- **Hardware validation**
  - Disk space (greater than or equal to 64 GB).
  - Physical memory (greater than or equal to 4 GB).
- **Azure VM security feature verification**
  - Trusted Launch.
  - Secure Boot.
  - vTPM.
- **Azure Virtual Desktop detection**
  - Flags unsupported pooled host pool configurations.
- **Upgrade recommendations**
  - Outputs supported upgrade paths or relevant upgrade guidance.

For more information, see Azure VM Windows OS Upgrade Assessment Script.

Output

```
Windows Version: Windows 10 Pro

[Passed] Disk Space (Free: 101.46 GB)
[Passed] Physical Memory (Total: 16 GB)
[Failed] VM Generation: Gen2 required for upgrade
[Failed] Trusted Launch
[Failed] Secure Boot
[Failed] TPM Enabled

FAILED: The VM is running Windows 10 Gen1. Upgrade to Windows 11 is only supported for Gen2 VMs
```



How to run the tool

You can run the tool in any of the following manners.

1. Download from GitHub and run within the VM

Download the scripts from GitHub and then run them manually. To access the scripts, follow the resource links in the previous sections.

2. Use prepackaged run command scripts

For more information, see Run scripts in your Windows VM by using action run commands.

Recommended workflow

1. Run `Windows_OSUpgrade_Assessment_Validation` to verify the OS can be upgraded.
2. Apply the suggested fixes or refer to the official documentation.

Additional resources

- In-place upgrade (server) for VMs running Windows Server in Azure
- In-place upgrade (client) for VMs running Windows in Azure

Third-party contact disclaimer

Microsoft provides third-party contact information to help you find additional information about this topic. This contact information may change without notice. Microsoft does not guarantee the accuracy of third-party contact information.

Last updated on 04/16/2026

In-place upgrade for supported VMs running Windows in Azure (Windows client)

Applies to: ✔ Windows VMs

Original product version: Windows 10, version 1803, all editions, Windows 10, version 1709, all editions, Virtual Machine running Windows, Windows 10, Windows 8.1, Windows 7 Enterprise
Original KB number: 4014997

Summary

This article describes how to do an in-place system upgrade of supported Windows 10-based and Windows-11-based Azure Virtual Machines (VMs). It also describes workarounds for Azure VMs that aren't supported for in-place system upgrades.

> ⓘ Note
>
> **Looking for Windows Server?**
> For Azure VMs running Windows Server, see In-place upgrade for supported Windows Server VMs.

> ⊗ Caution
>
> Following the process in this article causes a disconnection between the data plane and the control plane of the VM. Azure capabilities like Auto guest patching, Auto OS image upgrades, Hotpatching, and Azure Update Manager aren't available. To use these features, create a new VM by using your preferred operating system instead of performing an in-place upgrade.

> ⓘ Important
>
> **Virtual machine (VM)** running in Azure and experiencing Windows Update issues?

> Trying to diagnose and resolve Windows Update or Windows OS upgrade issues for your Azure VM? Try one of the following tools:
>
> - Azure VM Windows Update Error Detection Tool to diagnose specific Windows Update errors.
> - Azure VM Windows Windows Update Reset Tool to reset the Windows servicing stack.
> - Azure VM Windows OS Upgrade Assessment Tool to validate the OS upgrade path and any known issues.
>
> If you're experiencing performance problems with VMs, run these tools first before contacting support.

Symptoms

Consider the following scenario:

- You have a VM that's running Windows in a Microsoft Azure environment.
- You run an in-place upgrade of the VM to a newer version of the operating system.

In this scenario, the upgrade may fail or become blocked and require direct console access.

Cause

In-place system upgrades are supported for specific versions of Azure Windows VMs. We're working to broaden support of in-place system upgrade (as a Feature Update) to other versions soon.

Windows versions supported for in-place system upgrades

- Windows 10 single-session, all editions, all versions

  > **Note**
  > - You can't do an in-place upgrade from a single-session SKU of Windows to a multi-session SKU. For more information, see Can I upgrade a Windows VM to Windows Enterprise multi-session? .
  > - When migrating from Windows 10 to Windows 11, follow best practices by deploying new VMs. This approach avoids potential compatibility issues and

ensures an optimized configuration. The VM must meet the hardware requirements for Windows 11.

Windows versions that don't support in-place system upgrades (consider using a workaround)

- Windows 10 and 11 multi-session (all versions) — commonly used as Azure Virtual Desktop session hosts
- Windows 10 and 11 single-session up to Windows 10 and 11 multisesion (cross-SKU)
- Windows 8.1
- Windows 7 Enterprise

> **Important**
>
> **Azure Virtual Desktop session hosts:** In-place upgrade **isn't supported** for pooled or multisesion Azure Virtual Desktop session hosts. These VMs use Windows 10/11 Enterprise multisesion, which doesn't support in-place feature updates. Attempting an in-place upgrade (IPU) on an Azure Virtual Desktop session host can result in a broken session host, failed user connections, and loss of session host availability. For Azure Virtual Desktop environments, use image-based replacement instead. See Azure Virtual Desktop session host upgrade guidance later in this article.

In-place system upgrade process for a Windows 10 VM

This process takes 45-60 minutes to complete and restarts the VM. To perform the in-place system upgrade, follow these steps:

1. Run the Azure Virtual Machine (VM) Windows OS Upgrade Assessment Tool to validate the OS upgrade path and check for any known issues.
2. Verify that the Windows 10 VM doesn't use Ephemeral OS Disk. This feature isn't currently supported.
3. Verify that the Windows 10 VM has at least 2 GB of RAM, and 12 GB of free disk space on the system disk.

4. To prevent data loss, back up the Windows 10 VM by using Azure Backup. You can also use a third-party backup solution from Azure Marketplace Backup & Recovery.

5. To check whether the backup was successful, turn off the original Windows 10 VM and verify that you can restore a new VM from the backup and that all applications are running successfully.

   > **Note**
   >
   > You can use either the original Windows 10 VM or the restored VM as a source for an in-place system upgrade. The VMs can't run simultaneously unless you change one VM's system name and IP address to avoid conflicts.

6. Connect to the Windows 10 VM, and then go to **Settings** > **Updates & Security** > **Windows Update**.

7. In Windows Update, select **Check for updates**.

8. When the Feature Update item appears, select **Download and install now**.

The update downloads and installs. User settings and data are preserved, and the VM restarts automatically.

If you have general questions about this procedure, post them to Microsoft Q&A. Add the `azure-virtual-machines` tag to your questions.

Workaround

To work around this issue, create an Azure VM that's running a supported version. Download and upgrade the VHD of the VM.

To prevent data loss, back up the Windows 10 VM by using Azure Backup. Or, use a third-party backup solution from Azure Marketplace Backup & Recovery.

Download and upgrade the VHD

Step 1: Do an in-place upgrade in a local Hyper-V VM

1. Download the VHD of the VM.
2. Attach the VHD to a local Hyper-V VM.
3. Start the VM.

4. Run the in-place upgrade.

Step 2: Upload the VHD to Azure

Follow the steps in the following article to upload the VHD to Azure and to deploy the VM.

Upload a generalized VHD and use it to create new VMs in Azure

> **Note**
>
> When you perform an in-place upgrade on Azure Windows VMs, the VM properties on the Azure portal aren't updated. The changes are reflected only within the OS. This means that the source image information in the VM properties (including the publisher, offer, and plan) remains unchanged. The image that's used to deploy the VM remains the same. Only the OS is upgraded.

Azure Virtual Desktop session host upgrade guidance

Azure Virtual Desktop session hosts that use Windows 10 or Windows 11 Enterprise multi-session don't support IPUs. Attempting an IPU on these VMs can cause:

- Broken session host registration with the Azure Virtual Desktop host pool.
- User connection failures.
- Loss of FSLogix profile container connectivity.
- Session host unavailability requiring manual recovery.

Recommended approach: Image-based replacement

Instead of upgrading the OS on an existing session host, replace session hosts with new VMs built from an updated image:

1. **Create a new golden image** — Build a new VM image with the target Windows version, install required applications, and configure FSLogix and other Azure Virtual Desktop components.
2. **Add new session hosts** — Deploy new session hosts from the updated image into the existing host pool.
3. **Drain existing session hosts** — Set existing (old-version) session hosts to drain mode so no new user sessions connect to them.

4. **Remove old session hosts** — After users move to the new session hosts, remove the old VMs from the host pool and deallocate them.

For more information, see:

- Create a golden image in Azure
- Host pool management approaches for Azure Virtual Desktop

Azure Virtual Desktop personal desktop exception

For Azure Virtual Desktop **personal desktop** session hosts that use Windows 10 or Windows 11 **single-session** (not multisesion), in-place upgrade follows the same process as standard Azure VMs. See In-place system upgrade process for a Windows 10 VM earlier in this article.

> **Note**
>
> Personal desktops that use single-session Windows are eligible for in-place upgrade, but personal desktops that use multisesion Windows aren't. Verify the Windows SKU before attempting an upgrade. Run `winver` or check `SystemInfo` on the session host to confirm whether it's single-session or multisesion.

References

Microsoft server software support for Microsoft Azure virtual machines.

Third-party contact disclaimer

Microsoft provides third-party contact information to help you find additional information about this topic. This contact information may change without notice. Microsoft does not guarantee the accuracy of third-party contact information.

Last updated on 04/17/2026

Azure Virtual Machine (VM) Windows OS Upgrade assessment tool

Applies to: ✔ Windows VMs

Summary

This article provides information about the Azure Virtual Machine (VM) Windows OS Upgrade Assessment tool, a PowerShell script designed to assess the readiness of a Windows machine (desktop or server) for an **in-place OS upgrade**, with special considerations for **Azure VMs**. The tool evaluates OS version, supported upgrade paths, system disk space, and Azure security features like **Trusted Launch**, **Secure Boot**, and **virtual trusted platform modules (vTPMs)**.

Features

- **OS version detection**
  - Identifies whether the system is Windows 10, 11, or a Windows Server edition.
- **Server upgrade path check**
  - Matches current server version to supported upgrade targets.
- **Hardware validation**
  - Disk space (greater than or equal to 64 GB).
  - Physical memory (greater than or equal to 4 GB).
- **Azure VM security feature verification**
  - Trusted Launch.
  - Secure Boot.
  - vTPM.
- **Azure Virtual Desktop detection**
  - Flags unsupported pooled host pool configurations.
- **Upgrade recommendations**
  - Outputs supported upgrade paths or relevant upgrade guidance.

For more information, see Azure VM Windows OS Upgrade Assessment Script.

Output

```
Windows Version: Windows 10 Pro

[Passed] Disk Space (Free: 101.46 GB)
[Passed] Physical Memory (Total: 16 GB)
[Failed] VM Generation: Gen2 required for upgrade
[Failed] Trusted Launch
[Failed] Secure Boot
[Failed] TPM Enabled

FAILED: The VM is running Windows 10 Gen1. Upgrade to Windows 11 is only supported for Gen2 VMs
```



How to run the tool

You can run the tool in any of the following manners.

1. Download from GitHub and run within the VM

Download the scripts from GitHub and then run them manually. To access the scripts, follow the resource links in the previous sections.

2. Use prepackaged run command scripts

For more information, see Run scripts in your Windows VM by using action run commands.

Recommended workflow

1. Run `Windows_OSUpgrade_Assessment_Validation` to verify the OS can be upgraded.
2. Apply the suggested fixes or refer to the official documentation.

Additional resources

- In-place upgrade (server) for VMs running Windows Server in Azure
- In-place upgrade (client) for VMs running Windows in Azure

Third-party contact disclaimer

Microsoft provides third-party contact information to help you find additional information about this topic. This contact information may change without notice. Microsoft does not guarantee the accuracy of third-party contact information.

Last updated on 04/16/2026

Automatic Guest Patching for Azure Virtual Machines and Scale Sets

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

Enabling automatic guest patching for your Azure Virtual Machines (VMs) and Scale Sets (VMSS) helps ease update management by safely and automatically patching virtual machines to maintain security compliance, while limiting the blast radius of VMs.

Automatic VM guest patching has the following characteristics:

- Patches classified as *Critical* or *Security* are automatically downloaded and applied on the VM.
- Patches are applied during off-peak hours for IaaS VMs in the VM's time zone.
- Patches are applied during all hours for VMSS Flex.
- Azure manages the patch orchestration and follows availability-first principles.
- Virtual machine health, as determined through platform health signals, is monitored to detect patching failures.
- Application health can be monitored through the Application Health extension.
- Works for all VM sizes.

How does automatic VM guest patching work?

If automatic VM guest patching is enabled on a VM, then the available *Critical* and *Security* patches are downloaded and applied automatically on the VM. This process kicks off automatically every month when new patches are released. Patch assessment and installation are automatic, and the process includes rebooting the VM as configured. The rebootSetting parameter on the VM Model takes precedence over settings in another system, such as Maintenance Configuration.

The VM is assessed periodically every few days and multiple times within any 30-day period to determine the applicable patches for that VM. The patches can be installed any day on the VM during off-peak hours for the VM. This automatic assessment ensures that any missing patches are discovered at the earliest possible opportunity.

Patches are installed within 30 days of the monthly patch releases, following availability-first orchestration. Patches are installed only during off-peak hours for the VM, depending on the time zone of the VM. The VM must be running during the off-peak hours for patches to be automatically installed. If a VM is powered off during a periodic assessment, the platform automatically assess and apply patches (if necessary) during the next periodic assessment (usually within a few days) when the VM is powered on.

Definition updates and other patches not classified as *Critical* or *Security* won't be installed through automatic VM guest patching. To install patches with other patch classifications or schedule patch installation within your own custom maintenance window, you can use Azure Update Manager.

Enabling Automatic Guest Patching on single-instance VMs or Virtual Machine Scale Sets in Flexible orchestration mode allows the Azure platform to update your fleet in phases. Phased deployment follows Azure's Safe Deployment Practices and reduces the impact radius if any issues are identified with the latest update. Health monitoring is recommended for single instance VMs and required for Virtual Machine Scale Sets in Flexible orchestration mode to detect issues any issues with the update.

Availability-first Updates

Azure orchestrates the patch installation process across all public and private clouds for VMs that have enabled Automatic Guest Patching. The orchestration follows availability-first principles across different levels of availability provided by Azure.

For a group of virtual machines undergoing an update, the Azure platform orchestrates updates:

Across regions:

- A monthly update is orchestrated across Azure globally in a phased manner to prevent global deployment failures.
- A phase can have one or more regions, and an update moves to the next phases only if eligible VMs in a phase update successfully.
- Geo-paired regions aren't updated concurrently and can't be in the same regional phase.
- The success of an update is measured by tracking the VM's health post update. VM Health is tracked through platform health indicators for the VM.

Within a region:

- VMs in different Availability Zones aren't updated concurrently with the same update.
- VMs that aren't part of an availability set are batched on a best effort basis to avoid concurrent updates for all VMs in a subscription.

Within an availability set:

- All VMs in a common availability set aren't updated concurrently.
- VMs in a common availability set are updated within Update Domain boundaries and VMs across multiple Update Domains aren't updated concurrently.

- In an Update Domain, no more than 20% of the VMs within an availability set are be updated at a time. For availability sets with fewer than 10 VMs, VMs update one at a time within an Update Domain.

Restricting the number of concurrently patched VMs across regions, within a region, or within an availability set limits the impact of a faulty patch on a given set of VMs. With health monitoring, any potential issues are flagged before they impact the entire workload.

The patch installation date for a given VM may vary month-to-month, as a specific VM may be picked up in a different batch between monthly patching cycles.

Which patches are installed?

The patches installed depend on the rollout stage for the VM. Every month, a new global rollout is started where all security and critical patches assessed for an individual VM are installed for that VM. The rollout is orchestrated across all Azure regions in batches.

The exact set of patches to be installed vary based on the VM configuration, including OS type, and assessment timing. It's possible for two identical VMs in different regions to get different patches installed if there are more or less patches available when the patch orchestration reaches different regions at different times. Similarly, but less frequently, VMs within the same region but assessed at different times (due to different Availability Zone or Availability Set batches) might get different patches.

As the Automatic VM Guest Patching doesn't configure the patch source, two similar VMs configured to different patch sources, such as public repository vs private repository, may also see a difference in the exact set of patches installed.

For OS types that release patches on a fixed cadence, VMs configured to the public repository for the OS can expect to receive the same set of patches across the different rollout phases in a month. For example, Windows VMs configured to the public Windows Update repository.

As a new rollout is triggered every month, a VM receives at least one patch rollout every month if the VM is powered on during off-peak hours. This process ensures that the VM is patched with the latest available security and critical patches on a monthly basis. To ensure consistency in the set of patches installed, you can configure your VMs to assess and download patches from your own private repositories.

Supported OS images

> **Important**

Automatic VM guest patching, on-demand patch assessment and on-demand patch installation are supported only on VMs created from images with the exact combination of publisher, offer and sku from the below supported OS images list. Custom images or any other publisher, offer, sku combinations aren't supported. More images are added periodically. Don't see your SKU in the list? Request support by filing out Image Support Request.

Supported Windows Images (Hotpatchable)

Expand table

Publisher	OS Offer	Sku
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition-core
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition-smalldisk
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition-core-smalldisk
MicrosoftWindowsServer	MicrosoftServerOperatingSystems-previews	windows-server-2025-azure-edition-hotpatch
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-core-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-core
MicrosoftWindowsServer	MicrosoftServerOperatingSystems-previews	windows-server-2022-azure-edition-core-preview
MicrosoftWindowsServer	WindowsServerHotpatch-previews	windows-server-2022-azure-edition-hotpatch
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-hotpatch
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-hotpatch-smalldisk

Supported Windows Images (non-Hotpatchable)

Expand table

Publisher	OS Offer	Sku
Center-for-Internet-Security-Inc	cis-windows-server-2019-v1-0-0-l2	cis-ws2019-l2
Center-for-Internet-Security-Inc	cis-windows-server	cis-windows-server2019-l1-gen1
Center-for-Internet-Security-Inc	cis-windows-server	cis-windows-server2022-l1-gen1
Center-for-Internet-Security-Inc	cis-windows-server	cis-windows-server2022-l1-gen2
Center-for-Internet-Security-Inc	cis-windows-server	cis-windows-server2022-l2-gen2
Center-for-Internet-Security-Inc	cis-windows-server-2022-l2	cis-windows-server-2022-l2-gen2
MicrosoftWindowsServer	WindowsServer	2008-R2-SP1
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter-gensecond
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter-smalldisk-g2
MicrosoftWindowsServer	WindowsServer	2016-Datacenter
MicrosoftWindowsServer	WindowsServer	2016-datacenter-gensecond
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-Server-Core
MicrosoftWindowsServer	WindowsServer	2016-datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2016-datacenter-with-containers
MicrosoftWindowsServer	WindowsServer	2019-Datacenter
MicrosoftWindowsServer	WindowsServer	2019-Datacenter-Core
MicrosoftWindowsServer	WindowsServer	2019-datacenter-gensecond
MicrosoftWindowsServer	WindowsServer	2019-datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2019-datacenter-smalldisk-g2

Publisher	OS Offer	Sku
MicrosoftWindowsServer	WindowsServer	2019-datacenter-with-containers
MicrosoftWindowsServer	WindowsServer	2022-datacenter
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-core
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-core-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-hotpatch
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-hotpatch-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-datacenter-azure-edition-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-datacenter-core
MicrosoftWindowsServer	WindowsServer	2022-datacenter-core-g2
MicrosoftWindowsServer	WindowsServer	2022-datacenter-g2
MicrosoftWindowsServer	WindowsServer	2022-datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-datacenter-smalldisk-g2
MicrosoftWindowsServer	WindowsServerHotpatch-previews	windows-server-2022-azure-edition-hotpatch
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition-core
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition-core-smalldisk
MicrosoftWindowsServer	WindowsServer	2025-datacenter-azure-edition-smalldisk
MicrosoftWindowsServer	WindowsServer	2025-datacenter
MicrosoftWindowsServer	WindowsServer	2025-datacenter-core
MicrosoftWindowsServer	WindowsServer	2025-datacenter-core-smalldisk

Publisher	OS Offer	Sku
MicrosoftWindowsServer	WindowsServer	2025-datacenter-smalldisk
MicrosoftWindowsServer	MicrosoftServerOperatingSystems-previews	windows-server-2025-azure-edition-hotpatch

Supported Linux Images

Expand table

Publisher	OS Offer	Sku
AlmaLinux	almalinux-hpc	8-hpc-gen2
AlmaLinux	almalinux-hpc	8_5-hpc
AlmaLinux	almalinux-hpc	8_5-hpc-gen2
AlmaLinux	almalinux-hpc	8_6-hpc
AlmaLinux	almalinux-hpc	8_6-hpc-gen2
AlmaLinux	almalinux-hpc	8_7-hpc-gen1
AlmaLinux	almalinux-hpc	8_7-hpc-gen2
AlmaLinux	almalinux-hpc	8_10-hpc-gen1
AlmaLinux	almalinux-hpc	8_10-hpc-gen2
Canonical	UbuntuServer	16.04-LTS
Canonical	UbuntuServer	16.04.0-LTS
Canonical	UbuntuServer	18.04-LTS
Canonical	UbuntuServer	18_04-LTS-arm64
Canonical	UbuntuServer	18_04-LTS-gen2
Canonical	ubuntu-24_04-lts	cvm
Canonical	ubuntu-24_04-lts	server
Canonical	ubuntu-24_04-lts	server-arm64
Canonical	ubuntu-24_04-lts	server-gen1
Canonical	ubuntu-24_04-lts	ubuntu-pro

Publisher	OS Offer	Sku
Canonical	ubuntu-24_04-lts	ubuntu-pro-cvm
Canonical	0001-com-ubuntu-confidential-vm-focal	20_04-lts-cvm
Canonical	0001-com-ubuntu-confidential-vm-jammy	22_04-lts-cvm
Canonical	0001-com-ubuntu-pro-bionic	pro-18_04-lts
Canonical	0001-com-ubuntu-pro-focal	pro-20_04-lts
Canonical	0001-com-ubuntu-pro-focal	pro-20_04-lts-gen2
Canonical	0001-com-ubuntu-pro-jammy	pro-22_04-lts-gen2
Canonical	0001-com-ubuntu-server-focal	20_04-lts
Canonical	0001-com-ubuntu-server-focal	20_04-lts-arm64
Canonical	0001-com-ubuntu-server-focal	20_04-lts-gen2
Canonical	0001-com-ubuntu-server-jammy	22_04-lts
Canonical	0001-com-ubuntu-server-jammy	22_04-lts-arm64
Canonical	0001-com-ubuntu-server-jammy	22_04-lts-gen2
Center-for-Internet-Security-Inc	cis-rhel-7-v2-2-0-l1	cis-redhat7-l1
Center-for-Internet-Security-Inc	cis-rhel	cis-redhat7-l2-gen1
Center-for-Internet-Security-Inc	cis-rhel	cis-redhat8-l1-gen1
Center-for-Internet-Security-Inc	cis-rhel	cis-redhat8-l1-gen2
Center-for-Internet-Security-Inc	cis-rhel	cis-redhat8-l2-gen1
Center-for-Internet-Security-Inc	cis-rhel	cis-redhat9-l1-gen2
Center-for-Internet-Security-Inc	cis-rhel	cis-redhat9-l2-gen2
Center-for-Internet-Security-Inc	cis-ubuntu-linux-2004-l1	cis-ubuntu-linux-2004-l1

Publisher	OS Offer	Sku
Center-for-Internet-Security-Inc	cis-ubuntu-linux-2204-l1	cis-ubuntu-linux-2204-l1
Center-for-Internet-Security-Inc	cis-ubuntu	cis-ubuntulinux2004-l1-gen1
Center-for-Internet-Security-Inc	cis-ubuntu	cis-ubuntulinux2204-l1-gen2
Center-for-Internet-Security-Inc	cis-ubuntu	cis-ubuntulinux2404-l1-gen1
Center-for-Internet-Security-Inc	cis-ubuntu	cis-ubuntulinux2404-l1-gen2
MicrosoftCblMariner	azure-linux-3	azure-linux-3-gen2
MicrosoftCblMariner	cbl-mariner	cbl-mariner-1
MicrosoftCblMariner	cbl-mariner	1-gen2
MicrosoftCblMariner	cbl-mariner	cbl-mariner-2
MicrosoftCblMariner	cbl-mariner	cbl-mariner-2-arm64
MicrosoftCblMariner	cbl-mariner	cbl-mariner-2-gen2
MicrosoftCblMariner	cbl-mariner	cbl-mariner-2-gen2-fips
Microsoft-AKS	aks	aks-engine-ubuntu-1804-202112
Microsoft-DSVM	aml-workstation	ubuntu-20
Microsoft-DSVM	aml-workstation	ubuntu-20-gen2
Microsoft-DSVM	ubuntu-hpc	1804
Microsoft-DSVM	ubuntu-hpc	1804-ncv4
Microsoft-DSVM	ubuntu-hpc	2004
Microsoft-DSVM	ubuntu-hpc	2004-preview-ndv5
Microsoft-DSVM	ubuntu-hpc	2204
Microsoft-DSVM	ubuntu-hpc	2204-preview-ndv5
Redhat	RHEL	7.2
Redhat	RHEL	7.3

Publisher	OS Offer	Sku
Redhat	RHEL	7.4
Redhat	RHEL	7.5
Redhat	RHEL	7.6
Redhat	RHEL	7.7
Redhat	RHEL	7.8
Redhat	RHEL	7_9
Redhat	RHEL	7-lvm
Redhat	RHEL	7-raw
Redhat	RHEL	8
Redhat	RHEL	8.1
Redhat	RHEL	81gen2
Redhat	RHEL	8.2
Redhat	RHEL	82gen2
Redhat	RHEL	8_3
Redhat	RHEL	83-gen2
Redhat	RHEL	8_4
Redhat	RHEL	84-gen2
Redhat	RHEL	8_5
Redhat	RHEL	85-gen2
Redhat	RHEL	8_6
Redhat	RHEL	86-gen2
Redhat	RHEL	8_7
Redhat	RHEL	8_8
Redhat	RHEL	8_10
Redhat	RHEL	8-lvm
Redhat	RHEL	8-lvm-gen2

Publisher	OS Offer	Sku
Redhat	RHEL-RAW	8-raw
Redhat	RHEL-RAW	8-raw-gen2
Redhat	RHEL	9_0
Redhat	RHEL	9_1
Redhat	RHEL	9_2
Redhat	RHEL	9_3
Redhat	RHEL	9_4
Redhat	RHEL	9-lvm
Redhat	RHEL	9-lvm-gen2
Redhat	RHEL-ARM64	8_6-arm64
Redhat	RHEL-ARM64	9_0-arm64
Redhat	RHEL-ARM64	9_1-arm64
Redhat	RHEL-SAP-APPS	86sapapps-gen2
Redhat	RHEL-SAP-APPS	88sapapps-gen2
Redhat	RHEL-SAP-HA	86sapha-gen2
Redhat	RHEL-SAP-HA	88sapha-gen2
SUSE	sles-12-sp5	gen1
SUSE	sles-12-sp5	gen2
SUSE	sles-15-sp2	gen1
SUSE	sles-15-sp2	gen2
SUSE	sles-sap-15-sp4	gen2
SUSE	sles-sap-15-sp5	gen2

Patch orchestration modes

VMs on Azure now support the following patch orchestration modes:

AutomaticByPlatform (Azure-orchestrated patching):

- This mode is supported for both Linux and Windows VMs.
- This mode enables automatic VM guest patching for the virtual machine and subsequent patch installation is orchestrated by Azure.
- During the installation process, this mode assesses the VM for available patches and save the details in Azure Resource Graph.
- This mode is required for availability-first patching.
- This mode is only supported for VMs that are created using the supported OS platform images above.
- For Windows VMs, setting this mode also disables the native Automatic Updates on the Windows virtual machine to avoid duplication.
- To use this mode on Linux VMs, set the property `osProfile.linuxConfiguration.patchSettings.patchMode=AutomaticByPlatform` in the VM template.
- To use this mode on Windows VMs, set the property `osProfile.windowsConfiguration.patchSettings.patchMode=AutomaticByPlatform` in the VM template.
- Enabling this mode sets the Registry Key SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU\NoAutoUpdate to 1

AutomaticByOS:

- This mode is supported only for Windows VMs.
- This mode enables Automatic Updates on the Windows virtual machine, and patches are installed on the VM through Automatic Updates.
- This mode doesn't support availability-first patching.
- This mode is set by default if no other patch mode is specified for a Windows VM.
- To use this mode on Windows VMs, set the property `osProfile.windowsConfiguration.enableAutomaticUpdates=true`, and set the property `osProfile.windowsConfiguration.patchSettings.patchMode=AutomaticByOS` in the VM template.
- Enabling this mode sets the Registry Key SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU\NoAutoUpdate to 0

Manual:

- This mode is supported only for Windows VMs.
- This mode disables Automatic Updates on the Windows virtual machine. When deploying a VM using CLI or PowerShell, setting `--enable-auto-updates` to `false` also sets `patchMode` to `manual` and disables Automatic Updates.
- This mode doesn't support availability-first patching.
- This mode should be set when using custom patching solutions.

- To use this mode on Windows VMs, set the property `osProfile.windowsConfiguration.enableAutomaticUpdates=false`, and set the property `osProfile.windowsConfiguration.patchSettings.patchMode=Manual` in the VM template.
- Enabling this mode sets the Registry Key SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU\NoAutoUpdate to 1

ImageDefault:

- This mode is supported only for Linux VMs.
- This mode doesn't support availability-first patching.
- This mode honors the default patching configuration in the image used to create the VM.
- This mode is set by default if no other patch mode is specified for a Linux VM.
- To use this mode on Linux VMs, set the property `osProfile.linuxConfiguration.patchSettings.patchMode=ImageDefault` in the VM template.

> **Note**
>
> For Windows VMs, the property `osProfile.windowsConfiguration.enableAutomaticUpdates` can only be set when the VM is first created. This impacts certain patch mode transitions. Switching between AutomaticByPlatform and Manual modes is supported on VMs that have `osProfile.windowsConfiguration.enableAutomaticUpdates=false`. Similarly switching between AutomaticByPlatform and AutomaticByOS modes is supported on VMs that have `osProfile.windowsConfiguration.enableAutomaticUpdates=true`. Switching between AutomaticByOS and Manual modes isn't supported. Azure recommends that Assessment Mode be enabled on a VM even if Azure Orchestration isn't enabled for patching. This allows the platform to assess the VM every 24 hours for any pending updates, and save the details in Azure Resource Graph. The platform performs assessment to report consolidated results when the machine's desired patch configuration state is applied or confirmed. This is reported as a 'Platform'-initated assessment.

Requirements for enabling automatic VM guest patching

- The virtual machine must have the Azure VM Agent for Windows or Linux installed.
- For Linux VMs, the Azure Linux agent must be version 2.2.53.1 or higher. Update the Linux agent if the current version is lower than the required version.
- For Windows VMs, the Windows Update service must be running on the virtual machine.

- The virtual machine must be able to access the configured update endpoints. If your virtual machine is configured to use private repositories for Linux or Windows Server Update Services (WSUS) for Windows VMs, the relevant update endpoints must be accessible.
- Use Compute API version 2021-03-01 or higher to access all functionality including on-demand assessment and on-demand patching.
- Custom images aren't currently supported.
- VMSS Flexible Orchestration requires the installation of Application Health extension. This is optional for IaaS VMs.

Enable automatic VM guest patching

Automatic VM guest patching can be enabled on any Windows or Linux VM that is created from a supported platform image.

REST API for Linux VMs

The following example describes how to enable automatic VM guest patching:

```
PUT on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVirtualMachine?api-version=2020-12-01`
```

JSON

```
{
  "location": "<location>",
  "properties": {
    "osProfile": {
      "linuxConfiguration": {
        "provisionVMAgent": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform"
        }
      }
    }
  }
}
```

REST API for Windows VMs

The following example describes how to enable automatic VM guest patching:

```
PUT on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVirtualMachine?api-version=2020-12-01`
```

JSON

```
{
  "location": "<location>",
  "properties": {
    "osProfile": {
      "windowsConfiguration": {
        "provisionVMAgent": true,
        "enableAutomaticUpdates": true,
        "patchSettings": {
          "patchMode": "AutomaticByPlatform"
        }
      }
    }
  }
}
```

Azure PowerShell when creating a Windows VM

Use the Set-AzVMOperatingSystem cmdlet to enable automatic VM guest patching when creating a VM.

Azure PowerShell

```
Set-AzVMOperatingSystem -VM $VirtualMachine -Windows -ComputerName $ComputerName -Credential $Credential -ProvisionVMAgent -EnableAutoUpdate -PatchMode "AutomaticByPlatform"
```

Azure PowerShell when updating a Windows VM

Use the Set-AzVMOperatingSystem and Update-AzVM cmdlet to enable automatic VM guest patching on an existing VM.

Azure PowerShell

```
$VirtualMachine = Get-AzVM -ResourceGroupName "myResourceGroup" -Name "myVM"
Set-AzVMOperatingSystem -VM $VirtualMachine -PatchMode "AutomaticByPlatform"
Update-AzVM -ResourceGroupName "myResourceGroup" -VM $VirtualMachine
```

Azure CLI for Windows VMs

Use az vm create to enable automatic VM guest patching when creating a new VM. The following example configures automatic VM guest patching for a VM named *myVM* in the resource group named *myResourceGroup*:

Azure CLI

```
az vm create --resource-group myResourceGroup --name myVM --image Win2019Datacenter
--enable-agent --enable-auto-update --patch-mode AutomaticByPlatform
```

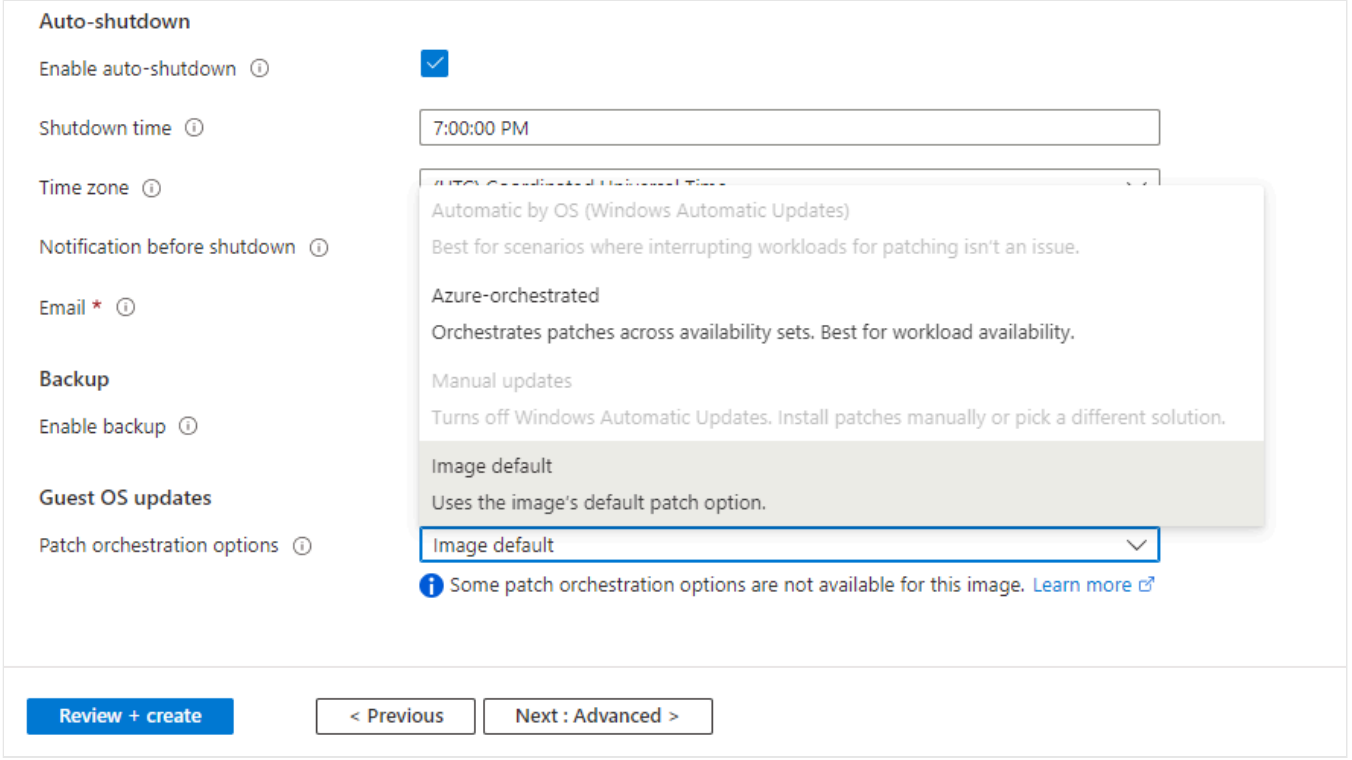
To modify an existing VM, use az vm update

Azure CLI

```
az vm update --resource-group myResourceGroup --name myVM --set
osProfile.windowsConfiguration.enableAutomaticUpdates=true
osProfile.windowsConfiguration.patchSettings.patchMode=AutomaticByPlatform
```

Azure portal

When creating a VM using the Azure portal, patch orchestration modes can be set under the **Management** tab for both Linux and Windows.



Enablement and assessment

> **Note**

It can take more than three hours to enable automatic VM guest updates on a VM, as the enablement is completed during the VM's off-peak hours. As assessment and patch installation occur only during off-peak hours, your VM must be also be running during off-peak hours to apply patches.

When automatic VM guest patching is enabled for a VM, a VM extension of type `Microsoft.CPlat.Core.LinuxPatchExtension` is installed on a Linux VM or a VM extension of type `Microsoft.CPlat.Core.WindowsPatchExtension` is installed on a Windows VM. This extension doesn't need to be manually installed or updated, as this extension is managed by the Azure platform as part of the automatic VM guest patching process.

It can take more than three hours to enable automatic VM guest updates on a VM, as the enablement is completed during the VM's off-peak hours. The extension is also installed and updated during off-peak hours for the VM. If the VM's off-peak hours end before enablement can be completed, the enablement process resumes during the next available off-peak time.

The platform makes periodic patching configuration calls to ensure alignment when model changes are detected on IaaS VMs or scale sets in Flexible orchestration. Certain model changes such as, but not limited to, updating assessment mode, patch mode, and extension update may trigger a patching configuration call.

Automatic updates are disabled in most scenarios, and patch installation is done through the extension going forward. The following conditions apply.

- If a Windows VM previously had Automatic Windows Update turned on through the AutomaticByOS patch mode, then Automatic Windows Update is turned off for the VM when the extension is installed.
- For Ubuntu VMs, the default automatic updates are disabled automatically when Automatic VM Guest Patching completes enablement.
- For RHEL, automatic updates need to be manually disabled. Execute:

Bash

```
sudo systemctl stop packagekit
```

Bash

```
sudo systemctl mask packagekit
```

To verify whether automatic VM guest patching has completed and the patching extension is installed on the VM, you can review the VM's instance view. If the enablement process is complete, the extension is installed and the assessment results for the VM are available under

`patchStatus`. The VM's instance view can be accessed through multiple ways as described below.

REST API

```
GET on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVirtualMachine/instanceView?api-version=2020-12-01`
```

Azure PowerShell

Use the Get-AzVM cmdlet with the `-Status` parameter to access the instance view for your VM.

Azure PowerShell

```
Get-AzVM -ResourceGroupName "myResourceGroup" -Name "myVM" -Status
```

PowerShell currently only provides information on the patch extension. Information about `patchStatus` will also be available soon through PowerShell.

Azure CLI

Use az vm get-instance-view to access the instance view for your VM.

Azure CLI

```
az vm get-instance-view --resource-group myResourceGroup --name myVM
```

Understanding the patch status for your VM

The `patchStatus` section of the instance view response provides details on the latest assessment and the last patch installation for your VM.

The assessment results for your VM can be reviewed under the `availablePatchSummary` section. An assessment is periodically conducted for a VM that has automatic VM guest patching enabled. The count of available patches after an assessment is provided under `criticalAndSecurityPatchCount` and `otherPatchCount` results. Automatic VM guest patching installs all patches assessed under the *Critical* and *Security* patch classifications. Any other assessed patch is skipped.

The patch installation results for your VM can be reviewed under the `lastPatchInstallationSummary` section. This section provides details on the last patch installation attempt on the VM, including the number of patches that were installed, pending, failed, or skipped. Patches are installed only during the off-peak hours maintenance window for the VM. Pending and failed patches are automatically retried during the next off-peak hours maintenance window.

Disable automatic VM guest patching

Automatic VM guest patching can be disabled by changing the patch orchestration mode for the VM.

To disable automatic VM guest patching on a Linux VM, change the patch mode to `ImageDefault`.

To enable automatic VM guest patching on a Windows VM, the property `osProfile.windowsConfiguration.enableAutomaticUpdates` determines which patch modes can be set on the VM and this property can only be set when the VM is first created. This impacts certain patch mode transitions:

- For VMs that have `osProfile.windowsConfiguration.enableAutomaticUpdates=false`, disable automatic VM guest patching by changing the patch mode to `Manual`.
- For VMs that have `osProfile.windowsConfiguration.enableAutomaticUpdates=true`, disable automatic VM guest patching by changing the patch mode to `AutomaticByOS`.
- Switching between AutomaticByOS and Manual modes isn't supported.

Use the examples from the enablement section above in this article for API, PowerShell and CLI usage examples to set the required patch mode.

On-demand patch assessment

If automatic VM guest patching is already enabled for your VM, a periodic patch assessment is performed on the VM during the VM's off-peak hours. This process is automatic and the results of the latest assessment can be reviewed through the VM's instance view as described earlier in this document. You can also trigger an on-demand patch assessment for your VM at any time. Patch assessment can take a few minutes to complete and the status of the latest assessment is updated on the VM's instance view.

> ⓘ **Note**

On-demand patch assessment doesn't automatically trigger patch installation. If you have enabled automatic VM guest patching, then the assessed and applicable patches for the VM are installed during the VM's off-peak hours, following the availability-first patching process described earlier in this document.

REST API

Use the Assess Patches API to assess available patches for your virtual machine.

```
POST on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVirtualMachine/assessPatches?api-version=2020-12-01`
```

Azure PowerShell

Use the Invoke-AzVmPatchAssessment cmdlet to assess available patches for your virtual machine.

Azure PowerShell

```
Invoke-AzVmPatchAssessment -ResourceGroupName "myResourceGroup" -VMName "myVM"
```

Azure CLI

Use az vm assess-patches to assess available patches for your virtual machine.

Azure CLI

```
az vm assess-patches --resource-group myResourceGroup --name myVM
```

On-demand patch installation

If automatic VM guest patching is already enabled for your VM, a periodic patch installation of Security and Critical patches is performed on the VM during the VM's off-peak hours. This process is automatic and the results of the latest installation can be reviewed through the VM's instance view as described earlier in this document.

You can also trigger an on-demand patch installation for your VM at any time. Patch installation can take a few minutes to complete and the status of the latest installation is

updated on the VM's instance view.

You can use on-demand patch installation to install all patches of one or more patch classifications. You can also choose to include or exclude specific packages for Linux or specific KB IDs for Windows. When triggering an on-demand patch installation, ensure that you specify at least one patch classification or at least one patch (package for Linux, KB ID for Windows) in the inclusion list.

REST API

Use the Install Patches API to install patches on your virtual machine.

```
POST on
`/subscriptions/subscription_id/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVirtualMachine/installPatches?api-version=2020-12-01`
```

Example request body for Linux:

JSON

```
{
  "maximumDuration": "PT1H",
  "Setting": "IfRequired",
  "linuxParameters": {
    "classificationsToInclude": [
      "Critical",
      "Security"
    ]
  }
}
```

Example request body for Windows:

JSON

```
{
  "maximumDuration": "PT1H",
  "rebootSetting": "IfRequired",
  "windowsParameters": {
    "classificationsToInclude": [
      "Critical",
      "Security"
    ]
  }
}
```

Azure PowerShell

Use the Invoke-AzVMInstallPatch cmdlet to install patches on your virtual machine.

Example to install certain packages on a Linux VM:

Azure PowerShell

```
Invoke-AzVmInstallPatch -ResourceGroupName "myResourceGroup" -VMName "myVM" -
MaximumDuration "PT90M" -RebootSetting "Always" -Linux -
ClassificationToIncludeForLinux "Security" -PackageNameMaskToInclude ["package123"]
-PackageNameMaskToExclude ["package567"]
```

Example to install all Critical patches on a Windows VM:

Azure PowerShell

```
Invoke-AzVmInstallPatch -ResourceGroupName "myResourceGroup" -VMName "myVM" -
MaximumDuration "PT2H" -RebootSetting "Never" -Windows   -
ClassificationToIncludeForWindows Critical
```

Example to install all Security patches on a Windows VM, while including and excluding patches with specific KB IDs and excluding any patch that requires a reboot:

Azure PowerShell

```
Invoke-AzVmInstallPatch -ResourceGroupName "myResourceGroup" -VMName "myVM" -
MaximumDuration "PT90M" -RebootSetting "Always" -Windows -
ClassificationToIncludeForWindows "Security" -KBNumberToInclude ["KB1234567",
"KB123567"] -KBNumberToExclude ["KB1234702", "KB1234802"] -ExcludeKBsRequiringReboot
```

Azure CLI

Use az vm install-patches to install patches on your virtual machine.

Example to install all Critical patches on a Linux VM:

Azure CLI

```
az vm install-patches --resource-group myResourceGroup --name myVM --maximum-
duration PT2H --reboot-setting IfRequired --classifications-to-include-linux
Critical
```

Example to install all Critical and Security patches on a Windows VM, while excluding any patch that requires a reboot:

Azure CLI

```
az vm install-patches --resource-group myResourceGroup --name myVM --maximum-duration PT2H --reboot-setting IfRequired --classifications-to-include-win Critical Security --exclude-kbs-requiring-reboot true
```

Strict Safe Deployment on Canonical Images

Microsoft and Canonical have partnered to make it easier for our customers to stay current with Linux OS updates and increase the security and resiliency of their Ubuntu workloads on Azure. By using Canonical's snapshot service, Azure now applies the same set of Ubuntu updates consistently to your fleet across regions.

Azure stores the package related updates within the customer repository for up to 90 days, depending on the available space. This allows customers to update their fleet leveraging Strict Safe Deployment for VMs that are up to three months behind on updates.

There's no action required for customers that have enabled Auto Patching. The platform installs a package that is snapped to a point-in-time by default. In the event a snapshot-based update can't be installed, Azure applies the latest package on the VM to ensure the VM remains secure. The point-in-time updates are consistent on all VMs across regions to ensure homogeneity. Customers can view the published date information related to the applied update in Azure Resource Graph and the Instance View of the VM.

Image End-of-Life (EOL)

Publishers may no longer support generating new updates for their images after a certain date. This is commonly referred to as End-of-life (EOL) for the image. Azure doesn't recommend using images after their EOL date, since it exposes the service to security vulnerabilities or performance issues. The Azure Guest Patching Service (AzGPS) communicates necessary steps for customers and impacted partners. AzGPS removes the image from the support list after the EOL date. VMs that use an end of life image on Azure might continue to work beyond their date. However, any issues experienced by these VMs aren't eligible for support.

Next steps

Learn more about creating and managing Windows virtual machines

Last updated on 12/01/2025

Hotpatch for Windows Server

07/17/2025Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022

Hotpatching is a way to install OS security updates on Windows Server without having to restart your machine. Hotpatching patches the in-memory code of running processes without the need to restart the process. Hotpatching also provides the following benefits:

- Fewer binaries mean updates install faster and consume less disk and CPU resources.
- Lower workload impact with less need to restart your machine.
- Better protection, as the Hotpatch update packages are scoped to Windows security updates that install faster without requiring you to restart your machine.
- Reduces the time exposed to security risks and change windows, and easier patch orchestration with Azure Update Manager.

Supported platforms

Azure and Azure Local virtual machines

The following table lists the exact combinations of publisher, OS offer, and SKU that support Hotpatching for Windows Server 2022 and Windows Server 2025 on Azure. Virtual machines (VMs) you create on Azure Local using these combinations also support Hotpatching.

> **Note**
>
> Windows Server container base images, custom images, or any other combination of publisher, offer, and SKU aren't supported.

Expand table

Publisher	OS Offer	SKU
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-Azure-Edition-Core
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-Azure-Edition-Core-smalldisk
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-Azure-Edition-Hotpatch
MicrosoftWindowsServer	WindowsServer	2022-Datacenter-Azure-Edition-Hotpatch-smalldisk
MicrosoftWindowsServer	WindowsServer	2025-Datacenter-Azure-Edition

Publisher	OS Offer	SKU
MicrosoftWindowsServer	WindowsServer	2025-Datacenter-Azure-Edition-smalldisk
MicrosoftWindowsServer	WindowsServer	2025-Datacenter-Azure-Edition-Core
MicrosoftWindowsServer	WindowsServer	2025-Datacenter-Azure-Edition-Core-smalldisk

For more information about the available images, see Windows Server on Azure Marketplace.

Azure Arc-connected machines

> **Important**
>
> Azure Arc-enabled Hotpatch for Windows Server 2025 is now available for a monthly subscription fee. To learn more about pricing, see Tired of all the restarts? Get hotpatching for Windows Server.

Azure Arc-connected Windows Server 2025 machines can receive Hotpatches if you enable the feature on Azure Arc Portal. To start using Azure Arc-enabled Hotpatch, connect Azure Arc to machines using one of the following editions:

- Windows Server 2025 Datacenter Edition
- Windows Server 2025 Standard Edition

How Hotpatch works

Hotpatch first establishes a baseline with the current Cumulative Update for Windows Server. Every three months, the baseline periodically refreshes with the latest Cumulative Update. You then receive Hotpatch releases for the next two months after the Cumulative Update. For example, if January is a Cumulative Update, February and March would have Hotpatch releases. For more information about the Hotpatch release schedule, see the Windows Server Hotpatch Calendar.

There are two types of baselines: **Planned baselines** and **Unplanned baselines**.

- **Planned baselines** are released on a regular cadence, with Hotpatch releases in between. Planned baselines include all the updates in a comparable Latest Cumulative Update for that month and require you to restart your machine.

    - For example, a planned one year release period might include four planned baseline releases in a calendar year and eight Hotpatch releases.

- **Unplanned baselines** are released during an unplanned important update, such as a zero-day fix, when that particular update can't be released as a Hotpatch. When unplanned baselines release, a Hotpatch release is replaced with an unplanned baseline for that month. Unplanned baselines also include all the updates in a comparable Latest Cumulative Update for that month, and therefore require you to restart your machine.
    - Because these events are unplanned, developers can't predict unplanned baselines in advance.

Hotpatch updates don't require you to restart your machine. Because Hotpatches patch the in-memory code of running processes without needing to restart them, your applications aren't affected. This lack of restarting doesn't affect the performance or functionality implications of the patch itself.

Supported updates

Hotpatch covers Windows Security updates and maintains parity with the content of security updates issued to in the regular non-Hotpatch Windows update channel.

There are a few important things you need to consider when enabling Hotpatch on a supported version of Windows Server. You still need to restart your machine to install updates that aren't included in the Hotpatch program. You also need to periodically restart after installing a new baseline. Restarting keeps your VM in sync with nonsecurity patches included in the latest cumulative updates.

The following patches currently aren't included in the Hotpatch program and require you to update your machine during Hotpatch release months:

- Nonsecurity updates for Windows
- .NET updates
- Non-Windows updates, such as drivers, firmware updates, and so on.

Patch orchestration process

Hotpatch is an extension of Windows Update and typical management processes. However, the tools for patch management vary depending on which platform you're using.

Azure

- VMs you create in Azure using a supported Windows Server image have Automatic VM Guest Patching enabled by default.
- Hotpatch automatically downloads and applies patches classified as Critical or Security to your VM.
- Hotpatch applies patches during off-peak hours in the VM time zone.
- Azure manages patches for you, applying patches according to availability-first principles.
- Azure monitors VM health through platform health signals to detect patching failures.

> **Note**
>
> You can't create Azure Virtual Machine Scale Sets with Uniform orchestration on Azure Edition images with Hotpatch. To learn more about which features Uniform orchestration for scale sets supports, see A comparison of Flexible, Uniform, and availability sets.

Azure Local

Azure Local can orchestrate Hotpatch updates for VMs using the following tools:

- Group Policy configures Windows Update client settings.
- SCONFIG configures Windows Update client settings for Server Core.
- Non-Microsoft patch management solutions.

Azure Arc-connected machines

Azure Arc-connected machines can install and manage Hotpatch updates using the following tools:

- Azure Update Manager
- Group Policy configures Windows Update client settings.
- SCONFIG configures Windows Update client settings for Server Core.
- Non-Microsoft patch management solutions.

For more information about which tools Hotpatch uses, check out our Azure Update Manager documentation.

Understand the patch status for your VM in Azure

To view the patch status for your VM, open the Overview page for your VM in the Azure portal. From there, under **Operations**, select **Updates**. You should see the patch status and most recently installed patches under **Recommended updates**.

In the **Recommended updates** page, you can see the Hotpatch status of your VM and if there are any available patches for your VM. As we stated in How Hotpatch works, Automatic VM Guest Patching automatically installs all Critical and Security patches on your VM.

Patches outside of those two categories aren't automatically installed, and are instead displayed in the **Update compliance** tab as a list of available patches. You can also check the **Update history** tab to view patch installation details for update deployments on your VM from the past 30 days.

Automatic VM Guest Patching regularly runs assessments of available patches, which you can view in the **Updates** tab. You can manually start an assessment by selecting the **Assess Now** button. You can also install patches on-demand by selecting the **Install updates now** button. This option lets you choose whether to install all updates under specific patch classifications. You can also select individual updates to include or exclude by providing a list of knowledge base articles. However, keep in mind that patches you install manually don't follow availability-first principles and might require you to restart your VM.

You can also view installed patches by running the Get-HotFix cmdlet in PowerShell or by viewing the **Settings** menu in Desktop Experience.

Rollback support for Hotpatching

Hotpatch updates don't support automatic rollback. If you experience an issue during or after an update, you must uninstall the latest update and install the last functional baseline update. This process requires that you restart the VM.

Next steps

- Enable Hotpatch for Azure Arc-enabled servers
- Automatic VM Guest Patching
- Enable Hotpatch for Azure Edition virtual machines built from ISO
- Azure Update Management

Automatic Extension Upgrade for virtual machines and scale sets in Azure

07/24/2025

Automatic Extension Upgrade is available for Azure Virtual Machines and Azure Virtual Machine Scale Sets. When Automatic Extension Upgrade is enabled on a virtual machine (VM) or scale set, Azure automatically monitors for new extension versions, assess the quality & safety of the new version, approves the version for rollout and then automatically upgrades all VMs using the extension. Azure upgrades all the VMs gradually following Safe Deployment Practices (SDP) to prevent bad version from causing outage and ensure high availabilty for applications. This gradual rollout can take 1-2 months to complete based on time required for assessment, the severity of the upgrade and the scale of VMs needed to be upgraded.

Automatic Extension Upgrade has the following features:

- Azure VMs, virtual machine scale sets and Arc VMs are supported.
- Upgrades are applied in an availability-first deployment model following (Safe Deployment Practices)[/azure/well-architected/operational-excellence/safe-deployments] (SDP).
- For a virtual machine scale set, by default, no more than 20% of the scale set VMs are upgraded in a single batch. This batch size can be changed by defining VMSS Rolling Upgrade Policy.
- For Azure VMs & Arc VMs, no more than 20% of the VMs are upgraded in a single batch. This batch size cannot be changed.
- All failed upgrades are automatically rolled back to try and bring the VM back to healthy state.
- All upgrades are rebootless. Therefore VM does not need a reboot after the upgrade.
- All VM sizes are supported.
- Both Windows and Linux extensions are compatible.
- Automatic upgrades are enabled by default starting API version `2025-04-01`.
- Each supported extension is enrolled individually. You can choose which extensions to upgrade automatically.
- All public, sovereign and airgapped cloud regions are supported.

How does Automatic Extension Upgrade work?

The extension upgrade process removes the existing extension on a VM and reinstalls the extension with the new version.

1. When the extension publisher publishes a new version of the same extension, Azure detects the new version and starts a quality assessment process. This process validates the version for security, quality and safe upgrades without failures. The version is approved for rollout once it passes the assessment.
2. Azure starts with one region or zone (Rollout starts from canary region) and once it is completely upgraded, it moves to the next region or zone.
3. Within a region/zone, Azure groups VMs into smaller batches (based on default or rolling upgrade policies) and upgrades one batch at a time. Azure moves to the next batch once the previous batch is successfully upgraded.
4. Azure monitors for VM and App health (using App health extension) after the upgrade to detect failures. If the VM isn't healthy within five minutes of the upgrade, the upgrade is rollback and previous version of the extension is installed. The upgrade is automatically retried after some time without needing customer intervention.

Availability-first Upgrades

The availability-first model for platform-orchestrated upgrades ensures that availability configurations in Azure are respected across multiple availability levels.

For a group of VMs undergoing an upgrade, the Azure platform orchestrates upgrades across regions, within a region, and within a set.

Across regions

- An upgrade moves across Azure globally in a phased manner to prevent Azure-wide deployment failures.
- A phase can have one or more regions, and an upgrade moves across phases only if eligible VMs in the previous phase upgrade successfully.
- Geo-paired regions aren't upgraded concurrently and can't be in the same regional phase.
- The success of an upgrade is measured by tracking the health of a VM post upgrade. VM health is tracked through platform health indicators for the VM (VM health & App Health reported by App Health Extension)

Within a region

- VMs in different availability zones aren't upgraded concurrently with the same upgrade.
- Single VMs that aren't part of an availability set are batched on a best-effort basis to avoid concurrent upgrades for all VMs in a subscription.

Within a set

- All VMs in a common availability set or scale set aren't upgraded concurrently.
- VMs in a common availability set are upgraded within update domain boundaries. VMs across multiple update domains aren't upgraded concurrently.
- VMs in a common virtual machine scale set are grouped in batches and upgraded within update domain boundaries. Upgrade policies defined on the scale set are honored during the upgrade. Each group is upgraded by using a rolling upgrade strategy.

Upgrade process for virtual machine scale sets

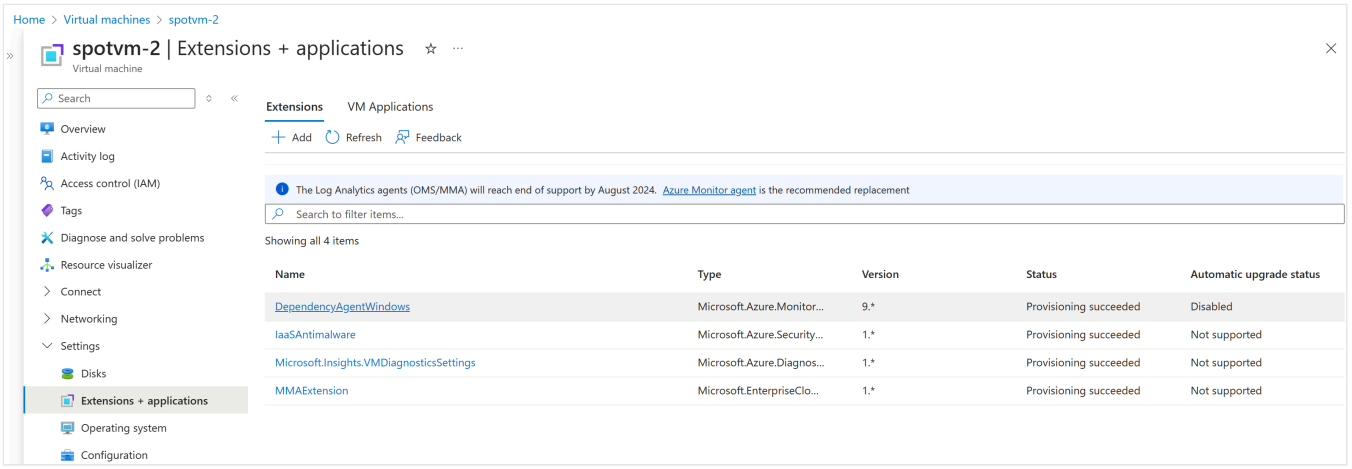
- Before the upgrade process starts, the orchestrator ensures that no more than 20% of VMs in the entire scale set are unhealthy (for any reason).
- The upgrade orchestrator identifies the batch of VM instances to upgrade. An upgrade batch can have a maximum of 20% of the total VM count, subject to a minimum batch size of one VM. The orchestrator considers the definition of the upgrade policy and availability zones while the batch is identified.
- After the upgrade, the VM health is always monitored before moving to the next batch. For scale sets with configured application health probes or the Application Health extension, application health is also monitored. The upgrade waits up to five minutes (or the defined health probe configuration) for the VM to become healthy before upgrading the next batch. If a VM doesn't recover its health after an upgrade, then by default, the previous extension version on the VM is reinstalled.
- The upgrade orchestrator also tracks the percentage of VMs that become unhealthy after an upgrade. The upgrade stops if more than 20% of upgraded instances become unhealthy during the upgrade process.

This process continues until all instances in the scale set are upgraded.

The scale set upgrade orchestrator checks for the overall scale set health before upgrading every batch. During a batch upgrade, other concurrent planned or unplanned maintenance activities could affect the health of your scale set VMs. In such cases, if more than 20% of the scale set's instances become unhealthy, the scale set upgrade stops at the end of the current batch.

Supported extensions

To check if your extensions are supported for automatic upgrade, view Automatic Upgrade status on Azure Portal - Extension blade.



Following are popular extensions supported for automatic upgrades (and more are added periodically):

Expand table

Publisher	Type
Microsoft.Azure.Automation.HybridWorker	HybridWorkerForLinux
Microsoft.Azure.Automation.HybridWorker	HybridWorkerForWindows
Microsoft.Azure.AzureDefenderForSQL	AdvancedThreatProtection.Windows
Microsoft.Azure.AzureDefenderForSQL	VulnerabilityAssessment.Windows
Microsoft.Azure.AzureDefenderForServers	MDE.Linux
Microsoft.Azure.AzureDefenderForServers	MDE.Windows
Microsoft.Azure.ChangeTrackingAndInventory	ChangeTracking-Linux
Microsoft.Azure.ChangeTrackingAndInventory	ChangeTracking-Windows
Microsoft.Azure.Diagnostics	LinuxDiagnostic
Microsoft.Azure.Extensions.Edp	LinuxHibernateTestExtension
Microsoft.Azure.Extensions.Edp	WindowsHibernateTestExtension
Microsoft.Azure.FleetDiagnostics	FleetDiagnosticsForWindows
Microsoft.Azure.Geneva	GenevaMonitoring
Microsoft.Azure.KeyVault	KeyVaultForLinux
Microsoft.Azure.KeyVault	KeyVaultForWindows
Microsoft.Azure.Labservices	Agent.Linux

Publisher	Type
Microsoft.Azure.Labservices	Agent.Windows
Microsoft.Azure.Monitor	AzureMonitorLinuxAgent
Microsoft.Azure.Monitor	AzureMonitorWindowsAgent
Microsoft.Azure.Monitoring.DependencyAgent.EDP	DependencyAgentLinux
Microsoft.Azure.Monitoring.DependencyAgent.EDP	DependencyAgentWindows
Microsoft.Azure.Monitoring.DependencyAgent	DependencyAgentLinux
Microsoft.Azure.Monitoring.DependencyAgent	DependencyAgentWindows
Microsoft.Azure.NetworkWatcher	NetworkWatcherAgentLinux
Microsoft.Azure.NetworkWatcher	NetworkWatcherAgentWindows
Microsoft.Azure.Networking.DNS	DNSClientCache
Microsoft.Azure.SCOMMI	GatewayServer
Microsoft.Azure.SCOMMI	WindowsAgent
Microsoft.Azure.Security.AntimalwareSignature	AntimalwareConfiguration
Microsoft.Azure.Security.Dsms	DSMSForWindows
Microsoft.Azure.Security.LinuxAttestation	GuestAttestation
Microsoft.Azure.Security.Monitoring	AzureSecurityLinuxAgent
Microsoft.Azure.Security.Monitoring	AzureSecurityWindowsAgent
Microsoft.Azure.Security.WindowsAttestation	GuestAttestation
Microsoft.Azure.Security.WindowsCodeIntegrity	CodeIntegrityAgent
Microsoft.Azure.ServiceFabric	ServiceFabricLinuxNode
Microsoft.Azure.Watson	WatsonLinuxAgent
Microsoft.Azure.Workloads	MonitoringExtensionLinux
Microsoft.Azure.Workloads	MonitoringExtensionWindows
Microsoft.CPlat.Core	LinuxHibernateExtension
Microsoft.CPlat.Core	WindowsHibernateExtension
Microsoft.CPlat.ProxyAgent	ProxyAgentLinux

Publisher	Type
Microsoft.CPlat.ProxyAgent	ProxyAgentWindows
Microsoft.EnterpriseCloud.Monitoring	MicrosoftMonitoringAgent
Microsoft.EnterpriseCloud.Monitoring	OmsAgentForLinux
Microsoft.GuestConfiguration	ConfigurationForLinux
Microsoft.GuestConfiguration	ConfigurationForWindows
Microsoft.ManagedServices	ApplicationHealthLinux
Microsoft.ManagedServices	ApplicationHealthWindows
Microsoft.OSTCExtensions	DSCForLinux
Microsoft.Sentinel.AzureMonitorAgentExtensions	MicrosoftDnsAgent
Microsoft.SqlServer.Management	SqlIaaSAgent
Microsoft.SqlServer.Management	SqlIaaSAgentLinux

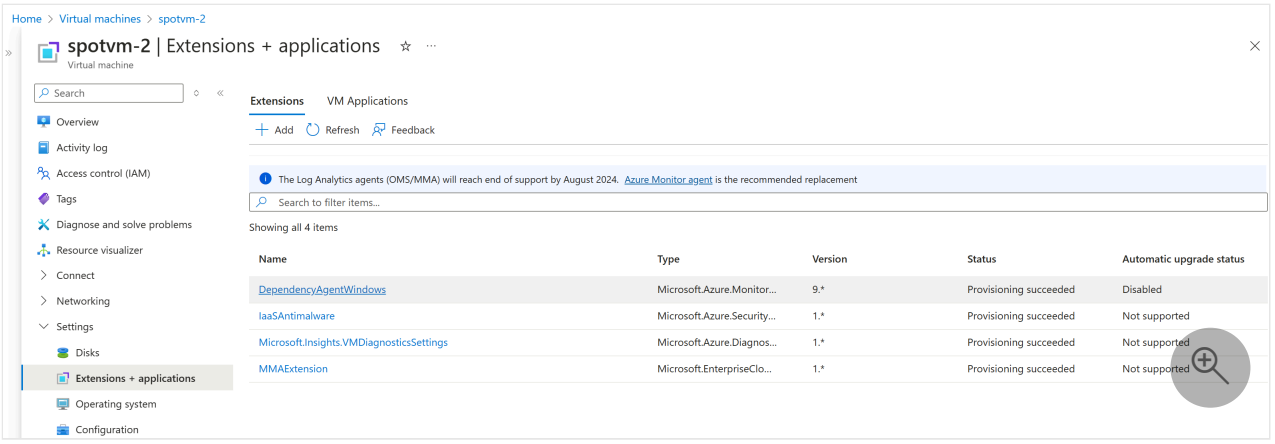
Enable Automatic Extension Upgrade

To enable Automatic Extension Upgrade for an extension, you must ensure that the property `enableAutomaticUpgrade` is set to `true` and added to every extension definition individually.

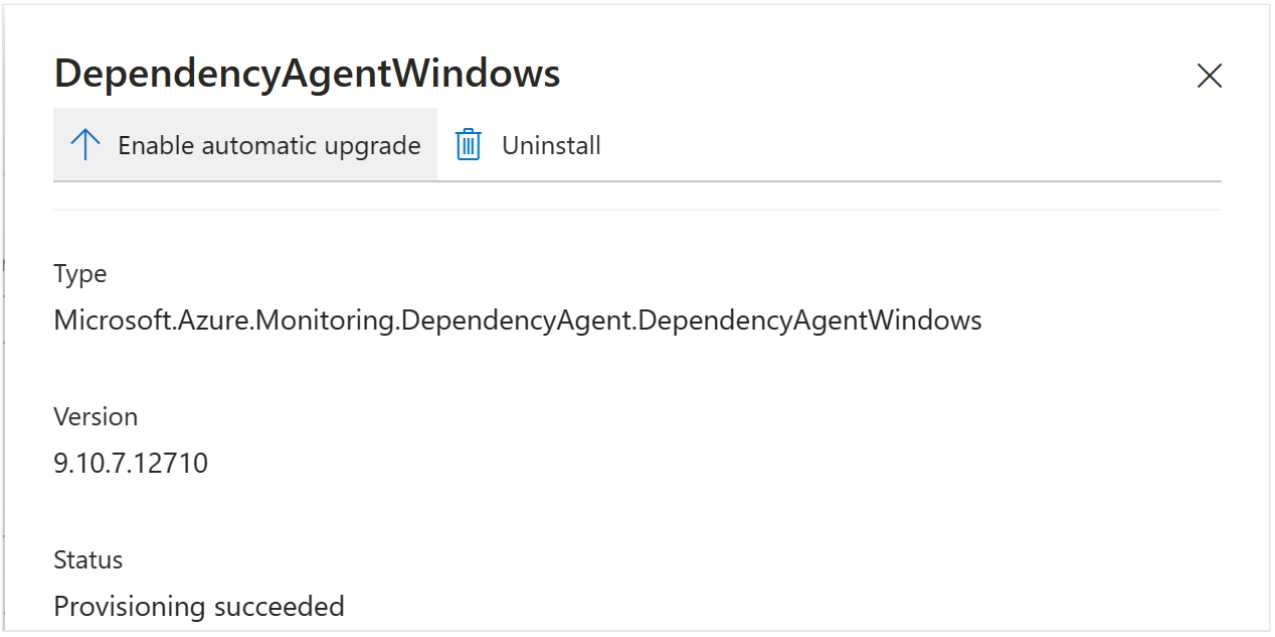
Use the Azure portal

In the Azure portal, use the **Extension** pane to enable automatic upgrade of extensions on existing VMs and virtual machine scale sets.

1. Go to the Virtual Machines or Virtual Machines Scale Sets pane, and select the resource name.
2. Under **Settings**, go to the **Extensions + applications** pane, which shows all extensions installed on the resource. The **Automatic upgrade status** column shows you if the automatic upgrade of the extension is enabled, disabled, or not supported.
3. Select the extension name to open the **Extensions** details pane.



4. Select **Enable automatic upgrade** to enable automatic upgrade of the extension. Use this button to disable an automatic upgrade, if necessary.



For virtual machines

REST API

To enable Automatic Extension Upgrade for an extension (in this example, the Dependency Agent extension) on an Azure VM, use the following call:

```
PUT on
`/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/
Microsoft.Compute/virtualMachines/<vmName>/extensions/<extensionName>?api-
version=2019-12-01`
```

JSON

```
{
    "name": "extensionName",
    "type": "Microsoft.Compute/virtualMachines/extensions",
    "location": "<location>",
    "properties": {
        "autoUpgradeMinorVersion": true,
        "enableAutomaticUpgrade": true,
        "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
        "type": "DependencyAgentWindows",
        "typeHandlerVersion": "9.5"
        }
}
```

For virtual machine scale sets

REST API

```
PUT on
`/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/
Microsoft.Compute/virtualMachineScaleSets/<vmssName>?api-version=2019-12-01`
```

JSON

```
{
   "location": "<location>",
   "properties": {
        "virtualMachineProfile": {
            "extensionProfile": {
                "extensions": [
                {
                "name": "<extensionName>",
                  "properties": {
                        "autoUpgradeMinorVersion": true,
                        "enableAutomaticUpgrade": true,
                    "publisher": "Microsoft.Azure.Monitoring.DependencyAgent",
                    "type": "DependencyAgentWindows",
                    "typeHandlerVersion": "9.5"
                    }
                }
                ]
            }
        }
    }
}
```

> ⓘ Note
>
> These operations set the `enableAutomaticUpgrade` property to `true` on the virtual machine scale set resource but not on the underlying VMs.

If the virtual machine scale set defines automatic or rolling upgrade mode in the upgradeProfile, the virtual machine scale set automatically propagates the change to each underlying VM.

If the virtual machine scale set defines manual mode in the `upgradeProfile`, you also need to manually update each instance and propagate the change to each underlying VM.

Extension upgrades with multiple extensions

A VM or virtual machine scale set can have multiple extensions with Automatic Extension Upgrade enabled. The same VM or scale set can also have other extensions without Automatic Extension Upgrade enabled.

If multiple extension upgrades are available for a VM, the upgrades might be batched together, but each extension upgrade is applied individually on a VM. A failure on one extension doesn't affect the other extensions that might be upgrading. For example, if two extensions are scheduled for an upgrade, and the first extension upgrade fails, the second extension is still upgraded.

You can also apply Automatic Extension Upgrade when a VM or virtual machine scale set has multiple extensions configured with extension sequencing. Extension sequencing is for the first-time deployment of the VM. Any future extension upgrades on an extension are applied independently.

Difference between EnableAutomaticUpgrade and AutoUpgradeMinorVersion

- `AutoUpgradeMinorVersion`:
  - This property is used during VM creation and while you upgrade the VM with a new configuration.
  - When set to `true`, it ensures that the latest minor version of the extension is automatically installed on the VM.
  - It overrides the `TypeHandlerVersion` with the latest stable minor version available.

    - When you upgrade the VM configuration, if a new minor version is available, it's considered a configuration change. The extension is reinstalled with the latest minor version.
    - In this way, newly created VMs keep up to date with the latest stable minor extension version.
    - If you want to manually set the extension to a specific version, set this property to `false`.

- `EnableAutomaticUpgrade`:
    - This property affects existing VMs.
    - It doesn't affect the version installed during VM creation.
    - After VM creation, if the VM isn't running the latest minor version of the extension, enable this property to trigger an automatic upgrade.
    - Upgrades don't cause VM reboot and are rolled out in a safe rolling manner. Failed upgrades are rolled back immediately to provide high service availability and reliability.
    - Existing VMs stay secure and up to date by automatically updating them to the latest minor version.

We recommend that you enable both properties to help keep all VMs secure and up to date.

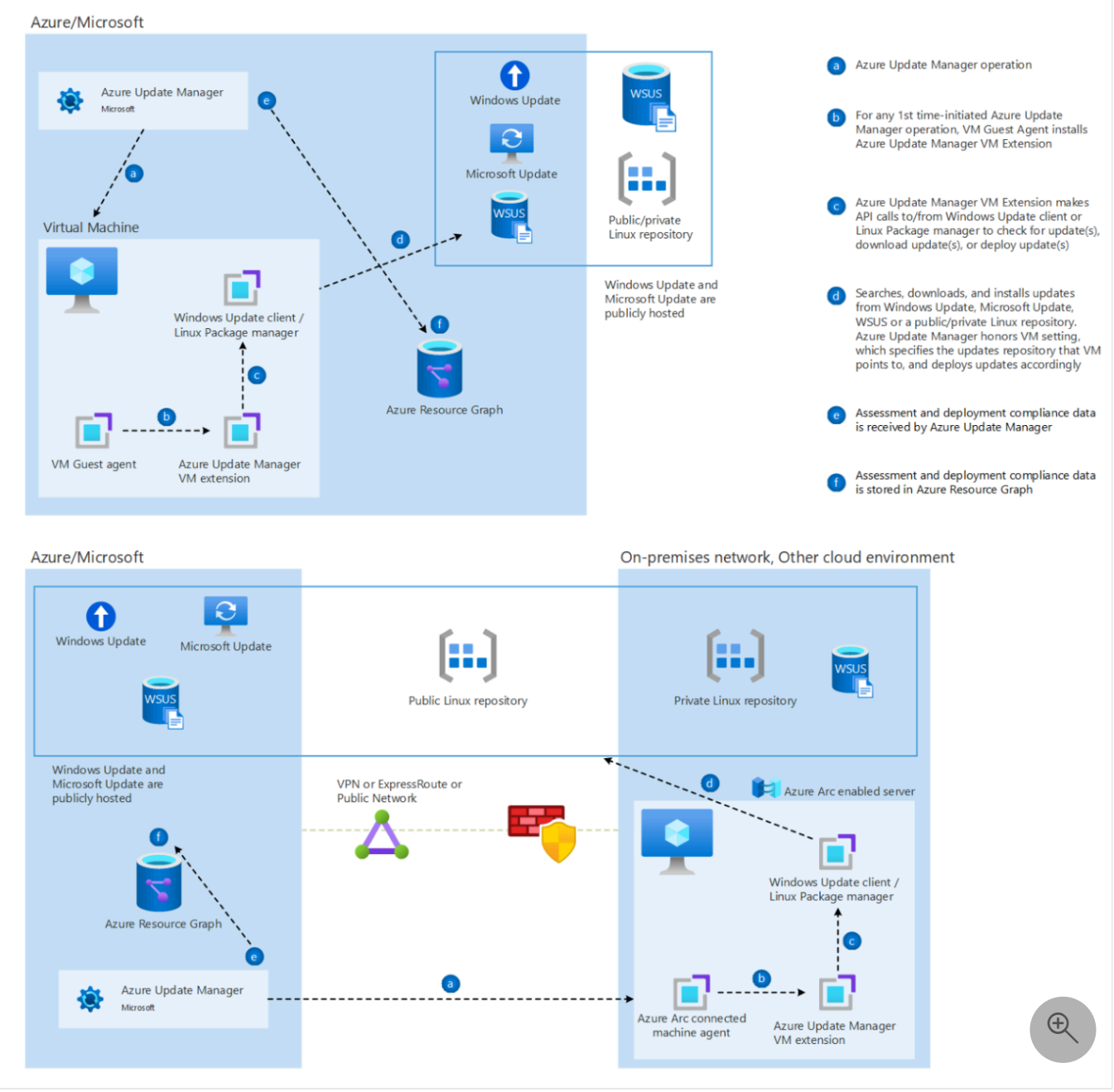
Upgrades to major extension versions are never performed automatically by either properties because major versions can include breaking changes. You must manually set `TypeHandlerVersion` to a major version and manually upgrade each existing VM to the latest major version.

Next step

Learn about the Application Health extension

How Update Manager works

Azure Update Manager assesses and applies updates to all Azure virtual machines (VMs) and Azure Arc-enabled servers for both Windows and Linux.



Update Manager VM extensions

When an Update Manager operation is enabled or triggered on your Azure VM or Azure Arc-enabled server, Update Manager installs an Azure extension or Azure Arc-enabled server extensions (respectively) on your machine to manage the updates.

The extensions are automatically installed when you start any Update Manager operation on your machine for the first time. These operations include **Check for updates**, **Install one-time**

update, and **Periodic Assessment**. The extensions are also automatically installed when a scheduled update deployment runs on your machine for the first time.

You don't have to explicitly install the extensions and their lifecycles. Update Manager manages the installation and configuration by using the following agents. These agents are required for Update Manager to work on your machines.

- For Azure VMs: Azure Windows VM Agent or Azure Linux VM Agent
- For Azure Arc-enabled servers: Azure Connected Machine agent

> **Note**
>
> Azure Arc connectivity is a prerequisite for Update Manager and non-Azure machines, including Azure Arc-enabled VMware vSphere and Azure Arc-enabled System Center Virtual Machine Manager.

For Azure machines, Update Manager installs a single extension. For Azure Arc-enabled machines, Update Manager installs two extensions. The following tabs provide details about the extensions:

Azure VM extensions

Expand table

Operating system	Extension
Windows	Microsoft.CPlat.Core.WindowsPatchExtension
Linux	Microsoft.CPlat.Core.LinuxPatchExtension

To view the available extensions for a VM in the Azure portal:

1. Go to the Azure portal and select a VM.
2. On the VM's home page, under **Settings**, select **Extensions + applications**.
3. On the **Extensions** tab, you can view the available extensions.

Update source

Update Manager honors the update source settings on the machine and fetches updates accordingly. Update Manager doesn't publish or provide updates.

Windows

If the Windows Update Agent (WUA) is configured to fetch updates from the Windows Update repository, the Microsoft Update repository, or Windows Server Update Services (WSUS), Update Manager honors these settings. For more information, see Configure Windows Update settings for Azure Update Manager. By default, *WUA is configured to fetch updates from the Windows Update repository*.

Update process

Update Manager performs the following steps:

1. Retrieve the assessment information about the status of system updates specified by the Windows Update client or Linux package manager.
2. Initiate the download and installation of updates by using the Windows Update client or Linux package manager.

The machines report their update status based on the source that they're configured to synchronize with. If the Windows Update service is configured to report to WSUS, the results in Update Manager might differ from what Microsoft Update shows, depending on when WSUS last synchronized with Microsoft Update. This behavior is the same for Linux machines that are configured to report to a local repository instead of a public package repository.

Update Manager finds only updates that the Windows Update service finds when you select the **Check for updates** button on the local Windows system. On Linux systems, Update Manager discovers only updates in the local repository.

Update Manager uses WUA APIs to install updates. The updates installed via WUA APIs don't appear on the Windows Update page within the Settings app on the machine. As a result, the updates installed through Update Manager aren't visible on the Windows Update page in the Settings app. The Windows Update page in the Settings app shows the progress and history of updates that the Windows Update orchestrator workflow manages. Learn more.

Update-related data stored in Azure Resource Graph

The Update Manager extension pushes all the pending update information and update installation results to Azure Resource Graph. Resource Graph retains data for the following time periods:

Expand table

Data	Retention period in Resource Graph
Pending updates (Resource Graph table name: `patchassessmentresources`)	7 days
Update installation results (Resource Graph table name: `patchinstallationresources`)	30 days

For more information, see the log structure of Azure Resource Graph and sample queries.

Installation of patches in Update Manager

Update Manager installs patches in the following manner:

1. Update Manager makes a fresh assessment of the available updates on the VM.
2. For Windows, the selected updates that meet the customer's criteria are installed one by one. For Linux, they're installed in batches.
3. During update installation, Update Manager checks maintenance window utilization at multiple steps.

   For Windows and Linux, 10 and 15 minutes (respectively) of the maintenance window are reserved for reboot at any point. Before Update Manager proceeds with the installation of the remaining updates, it checks whether the expected reboot time plus the average update installation time (for the next update or the next set of updates) doesn't exceed the maintenance window.

   In the case of Windows, the average update installation time is 10 minutes for all types of updates, except for service pack updates. For service pack updates, it's 15 minutes.
4. An ongoing update installation (after it's started based on the preceding calculation) isn't forcibly stopped, even if it exceeds the maintenance window, to avoid landing the machine in a possibly undetermined state. However, Update Manager doesn't install the remaining updates after the maintenance window ends, and it shows a "Maintenance window exceeded" error.
5. Update Manager marks an installation as successful only if all selected updates are installed and all operations involved (including reboot and assessment) succeed. Otherwise, the installation is marked as **Failed** or **Completed with warnings**. For example:

Expand table

Scenario	Update installation status
Installation of one of the selected updates fails.	**Failed**
A reboot doesn't happen for any reason, and the wait time for reboot times out.	**Failed**
The machine fails to start during a reboot.	**Failed**
Initial or final assessment failed.	**Failed**
Updates require a reboot, but the **Never reboot** option is selected.	**Completed with warnings**
ESM packages skipped patching in Ubuntu 18 or earlier if an Ubuntu Pro license wasn't present.	**Completed with warnings**

6. An assessment happens at the end. Sometimes, the reboot and assessment don't happen; for example, if the maintenance window ends or the update installation fails.

Related content

- Prerequisites for Azure Update Manager
- Check update compliance with Azure Update Manager
- Deploy updates now and track results with Azure Update Manager
- Automate assessment at scale by using Azure Policy
- Schedule recurring updates for machines by using the Azure portal and Azure Policy
- Manage update configuration settings
- Manage multiple machines with Azure Update Manager

Last updated on 11/18/2025

Azure Update Manager documentation

About Azure Update Manager

OVERVIEW

What is Azure Update Manager?

Update options

WHAT'S NEW

Feature releases

Common tasks

TUTORIAL

Enable periodic assessment and scheduled patching using Azure policy

Dynamically schedule updates at scale

Workbooks

CONCEPT

Overview of Workbooks

HOW-TO GUIDE

Create customized reports

Customized image support

HOW-TO GUIDE

Manage updates for customized images

Manage machine updates

QUICKSTART

Schedule updates at scale

HOW-TO GUIDE

View update compliance for machines

Deploy updates to machines

Schedule recurring updates for machines

Get started

TUTORIAL

Check and install on-demand updates

Dynamic Scoping

CONCEPT

About Dynamic Scoping

TUTORIAL

Schedule updates on Dynamic scopes

HOW-TO GUIDE

Manage dynamic scope operations

Pre and Post Events

CONCEPT

About Pre and Post Events

TUTORIAL

Create pre and post events using a webhook with Automation

Create pre and post events using Azure Functions

HOW-TO GUIDE

Create pre and post events

Manage pre and post events maintenance configuration events

Architecture

ARCHITECTURE

Plan deployment for updating Windows VMs in Azure

Maintenance for virtual machines in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure periodically updates its platform to improve the reliability, performance, and security of the host infrastructure for virtual machines. The purpose of these updates ranges from patching software components in the hosting environment to upgrading networking components or decommissioning hardware.

Updates rarely affect the hosted VMs. When updates do have an effect, Azure chooses the least impactful method for updates:

- If the update doesn't require a reboot, the VM is paused while the host is updated, or the VM is live-migrated to an already updated host.
- If maintenance requires a reboot, you're notified of the planned maintenance. Azure also provides a time window in which you can start the maintenance yourself, at a time that works for you. The self-maintenance window is typically 35 days (for Host machines) unless the maintenance is urgent. Azure is investing in technologies to reduce the number of cases in which planned platform maintenance requires the VMs to be rebooted. For instructions on managing planned maintenance, see Handling planned maintenance notifications using the Azure CLI, PowerShell or portal.

This page describes how Azure performs both types of maintenance. For more information about unplanned events (outages), see Manage the availability of VMs for Windows or the corresponding article for Linux.

Within a VM, you can get notifications about upcoming maintenance by using Scheduled Events for Windows or for Linux.

Maintenance that doesn't require a reboot

Most platform updates don't affect customer VMs. When a no-impact update isn't possible, Azure chooses the update mechanism that's least impactful to customer VMs.

When VM impacting maintenance is required it will almost always be completed through a VM pause for less than 10 seconds. In rare circumstances, no more than once every 18 months for general purpose VM sizes, Azure uses a mechanism that will pause

the VM for about 30 seconds. After any pause operation the VM clock is automatically synchronized upon resume.

Memory-preserving maintenance works for more than 90 percent of Azure VMs. It doesn't work for G, M, N, and H series. Azure increasingly uses live-migration technologies and improves memory-preserving maintenance mechanisms to reduce the pause durations.

These maintenance operations that don't require a reboot are applied one fault domain at a time. They stop if they receive any warning health signals from platform monitoring tools. Maintenance operations that do not require a reboot may occur simultaneously in paired regions or Availability Zones. For a given change, the deployment are mostly sequenced across Availability Zones and across Region pairs, but there can be overlap at the tail.

These types of updates can affect some applications. When the VM is live-migrated to a different host, some sensitive workloads might show a slight performance degradation in the few minutes leading up to the VM pause. To prepare for VM maintenance and reduce impact during Azure maintenance, try using Scheduled Events for Windows or Linux for such applications.

For greater control on all maintenance activities including zero-impact and rebootless updates, you can create a Maintenance Configuration feature. Creating a Maintenance Configuration gives you the option to skip all platform updates and apply the updates at your choice of time. For more information, see Managing platform updates with Maintenance Configurations.

Live migration

Live migration is an operation that doesn't require a reboot and that preserves memory for the VM. It causes a pause or freeze, typically lasting no more than 5 seconds. Except for G, L, N, and H series, all infrastructure as a service (IaaS) VMs, are eligible for live migration. Live Migration is available on majority of M-Series SKUs. Eligible VMs represent more than 90 percent of the IaaS VMs that are deployed to the Azure fleet.

> **Note**
>
> You won't receive a notification in the Azure portal for live migration operations that don't require a reboot. To see a list of live migrations that don't require a reboot, query for scheduled events.

The Azure platform starts live migration in the following scenarios:

- Planned maintenance
- Hardware failure
- Allocation optimizations

Some planned-maintenance scenarios use live migration, and you can use Scheduled Events to know in advance when live migration operations will start.

Live migration can also be used to move VMs when Azure Machine Learning algorithms predict an impending hardware failure or when you want to optimize VM allocations. For more information about predictive modeling that detects instances of degraded hardware, see Improving Azure VM resiliency with predictive machine learning and live migration. Live-migration notifications appear in the Azure portal in the Monitor and Service Health logs as well as in Scheduled Events if you use these services.

Maintenance that requires a reboot

In the rare case where VMs need to be rebooted for planned maintenance, you'll be notified in advance. Planned maintenance has two phases: the self-service phase and a scheduled maintenance phase.

During the *self-service phase*, which typically lasts four weeks, you start the maintenance on your VMs. As part of the self-service, you can query each VM to see its status and the result of your last maintenance request.

> **Note**
>
> For VM-series that do not support **Live Migration**, local (ephemeral) disks data can be lost during the maintenance events. See each individual VM-series for information on if Live Migration is supported.

When you start self-service maintenance, your VM is redeployed to an already updated node. Because the VM is redeployed, the temporary disk is lost and public dynamic IP addresses associated with the virtual network interface are updated.

If an error arises during self-service maintenance, the operation stops, the VM isn't updated, and you get the option to retry the self-service maintenance.

When the self-service phase ends, the *scheduled maintenance phase* begins. During this phase, you can still query for the maintenance phase, but you can't start the maintenance yourself.

For more information on managing maintenance that requires a reboot, see **Handling planned maintenance notifications** using the Azure CLI, PowerShell or portal.

Availability considerations during scheduled maintenance

If you decide to wait until the scheduled maintenance phase, there are a few things you should consider to maintain the highest availability of your VMs.

Paired regions

Each Azure region is paired with another region within the same geographical vicinity. Together, they make a region pair. During the scheduled maintenance phase, Azure updates only the VMs in a single region of a region pair. For example, while updating the VM in North Central US, Azure doesn't update any VM in South Central US at the same time. However, other regions such as North Europe can be under maintenance at the same time as East US. Understanding how region pairs work can help you better distribute your VMs across regions. For more information, see Azure region pairs.

Availability zones

Availability zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there’s a minimum of three separate zones in all enabled regions.

An availability zone is a combination of a fault domain and an update domain. If you create three or more VMs across three zones in an Azure region, your VMs are effectively distributed across three fault domains and three update domains. The Azure platform recognizes this distribution across update domains to make sure that VMs in different zones are not updated at the same time.

Each infrastructure update rolls out zone by zone, within a single region. But, you can have deployment going on in Zone 1, and different deployment going in Zone 2, at the same time. Deployments are not all serialized. But, a single deployment that requires a reboot only rolls out one zone at a time to reduce risk. In general, updates that require a reboot are avoided when possible, and Azure attempts to use Live Migration or provide customers control.

Virtual machine scale sets

Virtual machine scale sets in **Flexible** orchestration mode are an Azure compute resource allow you to combine the scalability of virtual machine scale sets in Uniform orchestration mode with the regional availability guarantees of availability sets.

With Flexible orchestration, you can choose whether your instances are spread across multiple zones, or spread across fault domains within a single region.

Availability sets and Uniform scale sets

When deploying a workload on Azure VMs, you can create the VMs within an *availability set* to provide high availability to your application. Using availability sets, you can ensure that during either an outage or maintenance events that require a reboot, at least one VM is available.

Within an availability set, individual VMs are spread across up to 20 update domains. During scheduled maintenance, only one update domain is updated at any given time. Update domains aren't necessarily updated sequentially.

Virtual machine *scale sets* in **Uniform** orchestration mode are an Azure compute resource that you can use to deploy and manage a set of identical VMs as a single resource. The scale set is automatically deployed across UDs, like VMs in an availability set. As with availability sets, when you use Uniform scale sets, only one UD is updated at any given time during scheduled maintenance.

For more information about setting up your VMs for high availability, see Manage the availability of your VMs for Windows or the corresponding article for Linux.

Next steps

You can use the Azure CLI, Azure PowerShell, or the portal to manage planned maintenance.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Handling planned maintenance notifications

07/09/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure periodically performs updates to improve the reliability, performance, and security of the host infrastructure for virtual machines. Updates are changes like patching the hosting environment or upgrading and decommissioning hardware. A majority of these updates are completed without any impact to the hosted virtual machines. However, there are cases where updates do have an impact:

- If the maintenance doesn't require a reboot, Azure pauses the VM for few seconds while the host is updated. These types of maintenance operations are applied fault domain by fault domain. Progress is stopped if any warning health signals are received.
- If maintenance requires a reboot, you get a notice of when the maintenance is planned. You're given a time window of about 35 days where you can start the maintenance yourself, when it works for you.

Planned maintenance that requires a reboot is scheduled in waves. Each wave has different scope (regions).

- A wave starts with a notification to customers. Virtual Machine related Maintenance notifications are available under Service Health in the Azure portal. For a small subset of Virtual Machine Planned Maintenance specific scenarios, Azure may also communicate the schedule by sending an additional email to the members of Subscription Owners group. Azure Service Health enables users to configure their own custom alerts for the Planned Maintenance category. With Azure Service Health alerts, you can add more recipients and messaging options like email, SMS, and webhooks using Activity Log Alerts.
- Except a subset of event types, the Service Health Planned Maintenance Notification now also has an additional tab "Impacted Resources" which lists the VM names in scope of the event. Read more about Impacted Resources Preview Feature.
- Once a notification goes out, a *self-service window* is made available. During this window, you can query which of your virtual machines are affected and start maintenance based on your own scheduling needs. The self-service window is typically about 35 days.
- After the self-service window, a *scheduled maintenance window* begins. At some point during this window, Azure schedules and applies the required maintenance to your virtual machine.

The goal in having two windows is to give you enough time to start maintenance and reboot your virtual machine while knowing when Azure will automatically start maintenance.

You can use the Azure portal, PowerShell, REST API, and CLI to query for the maintenance windows for your VMs and start self-service maintenance.

Should you start maintenance using during the self-service window?

The following guidelines should help you decide whether to use this capability and start maintenance at your own time.

> **Note**
>
> Self-service maintenance might not be available for all of your VMs. To determine if proactive redeploy is available for your VM, look for the **Start now** in the maintenance status. Self-service maintenance is currently not available for Cloud Services (Web/Worker Role) and Service Fabric.

Self-service maintenance isn't recommended for deployments using **availability sets**. Availability sets are already only updated one update domain at a time.

- Let Azure trigger the maintenance. For maintenance that requires reboot, maintenance will be done update domain by update domain. The update domains don't necessarily receive the maintenance sequentially, and that there's a 30-minute pause between update domains.
- If a temporary loss of some capacity (1 update domain) is a concern, you can add instances during the maintenance period.
- For maintenance that doesn't require reboot, updates are applied at the fault domain level.

Don't use self-service maintenance in the following scenarios:

- If you shut down your VMs frequently, either manually, using DevTest Labs, using auto-shutdown, or following a schedule, it could revert the maintenance status and therefore cause additional downtime.
- On short-lived VMs that you know will be deleted before the end of the maintenance wave.
- For workloads with a large state stored in the local (ephemeral) disk that is desired to be maintained upon update.
- For cases where you resize your VM often, as it could revert the maintenance status.

- If you have adopted scheduled events that enable proactive failover or graceful shutdown of your workload, 15 minutes before start of maintenance shutdown

Use self-service maintenance, if you're planning to run your VM uninterrupted during the scheduled maintenance phase and none of the counter-indications mentioned above are applicable.

It's best to use self-service maintenance in the following cases:

- You need to communicate an exact maintenance window to your management or end-customer.
- You need to complete the maintenance by a given date.
- You need to control the sequence of maintenance, for example, multi-tier application to guarantee safe recovery.
- More than 30 minutes of VM recovery time is needed between two update domains (UDs). To control the time between update domains, you must trigger maintenance on your VMs one update domain (UD) at a time.

FAQ

Q: Why do you need to reboot my virtual machines now?

A: While the majority of updates and upgrades to the Azure platform don't impact virtual machine's availability, there are cases where we can't avoid rebooting virtual machines hosted in Azure. We have accumulated several changes that require us to restart our servers that will result in virtual machines reboot.

Q: If I follow your recommendations for High Availability by using an Availability Set, am I safe?

A: Virtual machines deployed in an availability set or virtual machine scale sets have the notion of Update Domains (UD). When performing maintenance, Azure honors the UD constraint and won't reboot virtual machines from different UD (within the same availability set). Azure also waits for at least 30 minutes before moving to the next group of virtual machines.

For more information about high availability, see Availability for virtual machines in Azure.

Q: How do I get notified about planned maintenance?

A: A planned maintenance wave starts by setting a schedule to one or more Azure regions. Virtual Machine related Maintenance notifications are available under Service Health in the Azure portal. For a small subset of Virtual Machine Planned Maintenance specific scenarios, Azure may also communicate the schedule by sending an additional email to the members of Subscription Owners group.

Except a subset of event types, the Service Health Planned Maintenance Notification now also has an additional tab "Impacted Resources" which lists the VM names in scope of the event. Read more about Impacted Resources Preview Feature.

Azure Service Health enables users to configure their own custom alerts for the Planned Maintenance category. With Azure Service Health alerts you can add more recipients and messaging options like email, SMS, and webhooks using Activity Log Alerts.

In case you deploy a virtual machine to a region where planned maintenance is already scheduled, you won't receive the notification but rather need to check the maintenance state of the VM.

Q: I don't see any indication of planned maintenance in the portal, PowerShell, or CLI. What is wrong?

A: Information related to planned maintenance is available during a planned maintenance wave only for the VMs that are going to be impacted by it. In other words, if you see not data, it could be that the maintenance wave has already completed (or not started) or that your virtual machine is already hosted in an updated server.

Q: Is there a way to know exactly when my virtual machine will be impacted?

A: When setting the schedule, we define a time window of several days. However, the exact sequencing of servers (and VMs) within this window is unknown. Customers who would like to know the exact time for their VMs can use scheduled events and query from within the virtual machine and receive a 15-minute notification before a VM reboot.

Q: How long will it take you to reboot my virtual machine?

A: Depending on the size of your VM, reboot may take up to several minutes during the self-service maintenance window. During the Azure initiated reboots in the scheduled maintenance window, the reboot will typically take about 25 minutes. Note that in case you use Cloud Services (Web/Worker Role), Virtual Machine Scale Sets, or availability sets, you'll be given 30 minutes between each group of VMs (UD) during the scheduled maintenance window.

Q: What is the experience in the case of Virtual Machine Scale Sets?

A: Planned maintenance is now available for Virtual Machine Scale Sets. For instructions on how to initiate self-service maintenance refer planned maintenance for virtual machine scale sets document.

Q: What is the experience in the case of Cloud Services (Web/Worker Role) and Service Fabric?

A: While these platforms are impacted by planned maintenance, customers using these platforms are considered safe given that only VMs in a single Upgrade Domain (UD) will be impacted at any given time. Self-service maintenance is currently not available for Cloud Services (Web/Worker Role) and Service Fabric.

Q: I don't see any maintenance information on my VMs. What went wrong?

A: There are several reasons why you're not seeing any maintenance information on your VMs:

1. You're using a subscription marked as Microsoft internal.
2. Your VMs aren't scheduled for maintenance. It could be that the maintenance wave has ended, canceled, or modified so that your VMs are no longer impacted by it.
3. You have deallocated VM and then started it. This can cause VM to move to a location which doesn't have planned maintenance wave scheduled. So the VM won't show maintenance information any more.
4. You don't have the **Maintenance** column added to your VM list view. While we have added this column to the default view, customers who configured to see non-default columns must manually add the **Maintenance** column to their VM list view.

Q: My VM is scheduled for maintenance for the second time. Why?

A: There are several use cases where you'll see your VM scheduled for maintenance after you have already completed your maintenance-redeploy:

1. We have canceled the maintenance wave and restarted it with a different payload. It could be that we've detected faulted payload and we simply need to deploy an additional payload.
2. Your VM was *service healed* to another node due to a hardware fault.
3. You have selected to stop (deallocate) and restart the VM.
4. You have **auto shutdown** turned on for the VM.

Q: Can I find out exactly when my VM will be rebooted during the planned maintenance window?

A: Azure provides a general maintenance window, but doesn't share the exact time when VM will be rebooted within that window. The reboot can happen at any point during that time. You can utilize self-maintenance window to control the maintenance from your end.

Next steps

You can handle planned maintenance using the Azure CLI, Azure PowerShell or portal.

Handling planned maintenance notifications using the Azure CLI

Article • 08/23/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can use the CLI to see when VMs are scheduled for maintenance. Planned maintenance information is available from az vm get-instance-view.

Maintenance information is returned only if there is maintenance planned.

Azure CLI

```
az vm get-instance-view -n myVM -g myResourceGroup --query
instanceView.maintenanceRedeployStatus
```

Output


```
      "maintenanceRedeployStatus": {
      "additionalProperties": {},
      "isCustomerInitiatedMaintenanceAllowed": true,
      "lastOperationMessage": null,
      "lastOperationResultCode": "None",
      "maintenanceWindowEndTime": "2018-06-04T16:30:00+00:00",
      "maintenanceWindowStartTime": "2018-05-21T16:30:00+00:00",
      "preMaintenanceWindowEndTime": "2018-05-19T12:30:00+00:00",
      "preMaintenanceWindowStartTime": "2018-05-14T12:30:00+00:00"
```


Start maintenance

The following call will start maintenance on a VM if `IsCustomerInitiatedMaintenanceAllowed` is set to true.

Azure CLI

```
az vm perform-maintenance -g myResourceGroup -n myVM
```

Classic deployments

> **Important**
>
> VMs created through the classic deployment model will be retired on September 1, 2023.
>
> If you use IaaS resources from Azure Service Management, please complete your migration by September 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.
>
> For more information, see Migrate your IaaS resources to Azure Resource Manager by September 1, 2023.

If you still have legacy VMs that were deployed using the classic deployment model, you can use the Azure classic CLI to query for VMs and initiate maintenance.

Make sure you are in the correct mode to work with classic VM by typing:

```
azure config mode asm
```

To get the maintenance status of a VM named *myVM*, type:

```
azure vm show myVM
```

To start maintenance on your classic VM named *myVM* in the *myService* service and *myDeployment* deployment, type:

```
azure compute virtual-machine initiate-maintenance --service-name myService
--name myDeployment --virtual-machine-name myVM
```

Next steps

You can also handle planned maintenance using the Azure PowerShell or portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Handling planned maintenance notifications using the portal

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Once a planned maintenance wave is scheduled, you can check for a list of virtual machines that are impacted.

You can use the Azure portal and look for VMs scheduled for maintenance.

1. Sign in to the Azure portal.
2. Search for or select **Virtual Machines**.
3. In the Virtual Machines pane, select the **More** menu and then select **Maintenance - > Virtual machine maintenance** to open the list with maintenance columns.

   Maintenance status: Shows the maintenance status for the VM. The following are the potential values:

Expand table

Value	Description
Start now	The VM is in the self-service maintenance window that lets you initiate the maintenance yourself. See below on how to start maintenance on your VM.
Scheduled	The VM is scheduled for maintenance with no option for you to initiate maintenance. You can learn of the maintenance window by selecting the Maintenance - Scheduled window in this view or by clicking on the VM.
Already updated	Your VM is already updated and no further action is required at this time.
Retry later	You have initiated maintenance with no success. You will be able to use the self-service maintenance option at a later time.
Retry now	You can retry a previously unsuccessful self-initiated maintenance.
-	Your VM is not part of a planned maintenance wave.

   Maintenance - Self-service window: Shows the time window when you can self-start maintenance on your VMs.

Maintenance - Scheduled window: Shows the time window when Azure will maintain your VM in order to complete maintenance.

Notification and alerts in the portal

Azure Service Health has a dedicated tab for Planned Maintenance where all Azure services (for example, Virtual Machines) publish their upcoming Maintenance events.

Virtual Machine related Maintenance notifications are available under Service Health in the Azure portal. For some specific Virtual Machine Planned Maintenance scenarios, Azure might communicate the schedule by sending an additional email (besides Service Health) to the Subscription Classic Admin, Co-Admin, and Subscription Owners group.

Azure Service Health enables users to configure their own custom Service Health alerts for the Planned Maintenance category. With Azure Service Health alerts, you can assign different Action-Groups to include additional recipients and channels (such as emails and SMS) based on event or service type, like Virtual Machine maintenance in this context. For more information, see Create activity log alerts on service notifications.

While creating alerts specific to Virtual Machine maintenance, make sure you set the **Event type** as **Planned maintenance** and **Services** as **Virtual Machine Scale Sets** and/or **Virtual Machines**.

Start Maintenance on your VM from the portal

While looking at the VM details, you will be able to see more maintenance-related details.
At the top of the VM details view, a new notification ribbon will be added if your VM is included in a planned maintenance wave. In addition, a new option is added to start maintenance when possible.

Click on the maintenance notification to see the maintenance page with more details on the planned maintenance. From there, you will be able to **start maintenance** on your VM.

Once you start maintenance, your virtual machine will be maintained and the maintenance status will be updated to reflect the result within few minutes.

If you missed the self-service window, you will still be able to see the window when your VM will be maintained by Azure.

> ⓘ **Note**

In most cases, scheduled maintenance cannot be delayed. If you need a scheduled maintenance to be delayed, contact support and Azure will make a best-effort to postpone.

Next steps

You can also handle planned maintenance using the Azure CLI or PowerShell.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Handling planned maintenance using PowerShell

Article • 08/23/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can use Azure PowerShell to see when VMs are scheduled for maintenance. Planned maintenance information is available from the Get-AzVM cmdlet when you use the `-status` parameter.

Maintenance information is returned only if there is maintenance planned. If no maintenance is scheduled that impacts the VM, the cmdlet does not return any maintenance information.

PowerShell

```
Get-AzVM -ResourceGroupName myResourceGroup -Name myVM -Status
```

Output

```
MaintenanceRedeployStatus               :
  IsCustomerInitiatedMaintenanceAllowed : True
  PreMaintenanceWindowStartTime         : 5/14/2018 12:30:00 PM
  PreMaintenanceWindowEndTime           : 5/19/2018 12:30:00 PM
  MaintenanceWindowStartTime            : 5/21/2018 4:30:00 PM
  MaintenanceWindowEndTime              : 6/4/2018 4:30
  LastOperationResultCode               : None
```

The following properties are returned under MaintenanceRedeployStatus:

Expand table

Value	Description
IsCustomerInitiatedMaintenanceAllowed	Indicates whether you can start maintenance on the VM at this time
PreMaintenanceWindowStartTime	The beginning of the maintenance self-service window when you can initiate maintenance on your VM

Value	Description
PreMaintenanceWindowEndTime	The end of the maintenance self-service window when you can initiate maintenance on your VM
MaintenanceWindowStartTime	The beginning of the maintenance scheduled in which Azure initiates maintenance on your VM
MaintenanceWindowEndTime	The end of the maintenance scheduled window in which Azure initiates maintenance on your VM
LastOperationResultCode	The result of the last attempt to initiate maintenance on the VM

You can also get the maintenance status for all VMs in a resource group by using Get-AzVM and not specifying a VM.

PowerShell

```
Get-AzVM -ResourceGroupName myResourceGroup -Status
```

The following PowerShell example takes your subscription ID and returns a list of VMs indicating whether they are scheduled for maintenance.

PowerShell

```
function MaintenanceIterator {
  param (
    $SubscriptionId
  )

  Select-AzSubscription -SubscriptionId $SubscriptionId | Out-Null

  $rgList = Get-AzResourceGroup
  foreach ($rg in $rgList) {
    $vmList = Get-AzVM -ResourceGroupName $rg.ResourceGroupName
    foreach ($vm in $vmList) {
      $vmDetails = Get-AzVM -ResourceGroupName $rg.ResourceGroupName -Name
$vm.Name -Status
      [pscustomobject]@{
        Name                                = $vmDetails.Name
        ResourceGroupName                   = $rg.ResourceGroupName
        IsCustomerInitiatedMaintenanceAllowed =
[bool]$vmDetails.MaintenanceRedeployStatus.IsCustomerInitiatedMaintenanceAll
owed
        LastOperationMessage                =
$vmDetails.MaintenanceRedeployStatus.LastOperationMessage
      }
    }
```

```
  }
}
```

Start maintenance on your VM using PowerShell

Using information from the function in the previous section, the following starts maintenance on a VM if **IsCustomerInitiatedMaintenanceAllowed** is set to true.

PowerShell

```
MaintenanceIterator -SubscriptionId <Subscription ID> |
    Where-Object -FilterScript {$_.IsCustomerMaintenanceAllowed} |
        Restart-AzVM -PerformMaintenance
```

Classic deployments

> **ⓘ Important**
>
> VMs created through the classic deployment model will be retired on September 1, 2023.
>
> If you use IaaS resources from Azure Service Management, please complete your migration by September 1, 2023. We encourage you to make the switch sooner to take advantage of the many feature enhancements in Azure Resource Manager.
>
> For more information, see **Migrate your IaaS resources to Azure Resource Manager by September 1, 2023**.

If you still have legacy VMs that were deployed using the classic deployment model, you can use PowerShell to query for VMs and initiate maintenance.

To get the maintenance status of a VM, type:

```
Get-AzureVM -ServiceName <Service name> -Name <VM name>
```

To start maintenance on your classic VM, type:

```
Restart-AzureVM -InitiateMaintenance -ServiceName <service name> -Name <VM
name>
```

Next steps

You can also handle planned maintenance using the Azure CLI or portal.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Managing VM updates with Maintenance Configurations

Article • 04/02/2025

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Azure frequently updates its infrastructure to improve reliability, performance, and security, or to launch new features. Most updates are transparent to users, but some sensitive workloads can't tolerate even few seconds of a virtual machine (VM) freezing or disconnecting for maintenance. Sensitive workloads might include gaming, media streaming, and financial transactions.

You can use the Maintenance Configurations feature to control and manage updates for many Azure VM resources. Maintenance Configurations is integrated with Azure Resource Graph for a low-latency and high-scale customer experience.

> **ⓘ Important**
>
> To use Maintenance Configurations, you must have a role of at least contributor and ensure that your subscription is registered with a maintenance resource provider.

Scopes

Maintenance Configurations currently supports three scopes: host, OS image, and guest. Although each scope allows scheduling and managing updates, the major difference lies in the resources that they support:

Expand table

Scope	Supported resources
Host	Isolated VMs, isolated virtual machine scale sets, dedicated hosts
OS image	Virtual machine scale sets
Guest	VMs, Azure Arc-enabled servers

Host

With the host scope, you can manage platform updates that don't require a restart on your isolated VMs, isolated virtual machine scale sets, and dedicated hosts.

Features and limitations unique to this scope include:

- You can set schedules anytime within 35 days. After 35 days, updates are automatically applied.
- A minimum of a two-hour maintenance window is required.
- Rack-level maintenance isn't currently supported.

Learn more about Azure dedicated hosts.

OS image

Using the OS image scope with Maintenance Configurations lets you decide when to apply upgrades to OS disks in your virtual machine scale sets through an easier and more predictable experience. An upgrade works by replacing the OS disk of a VM with a new disk created from the latest image version. Any configured extensions and custom data scripts are run on the OS disk, while data disks are retained.

Features and limitations unique to this scope include:

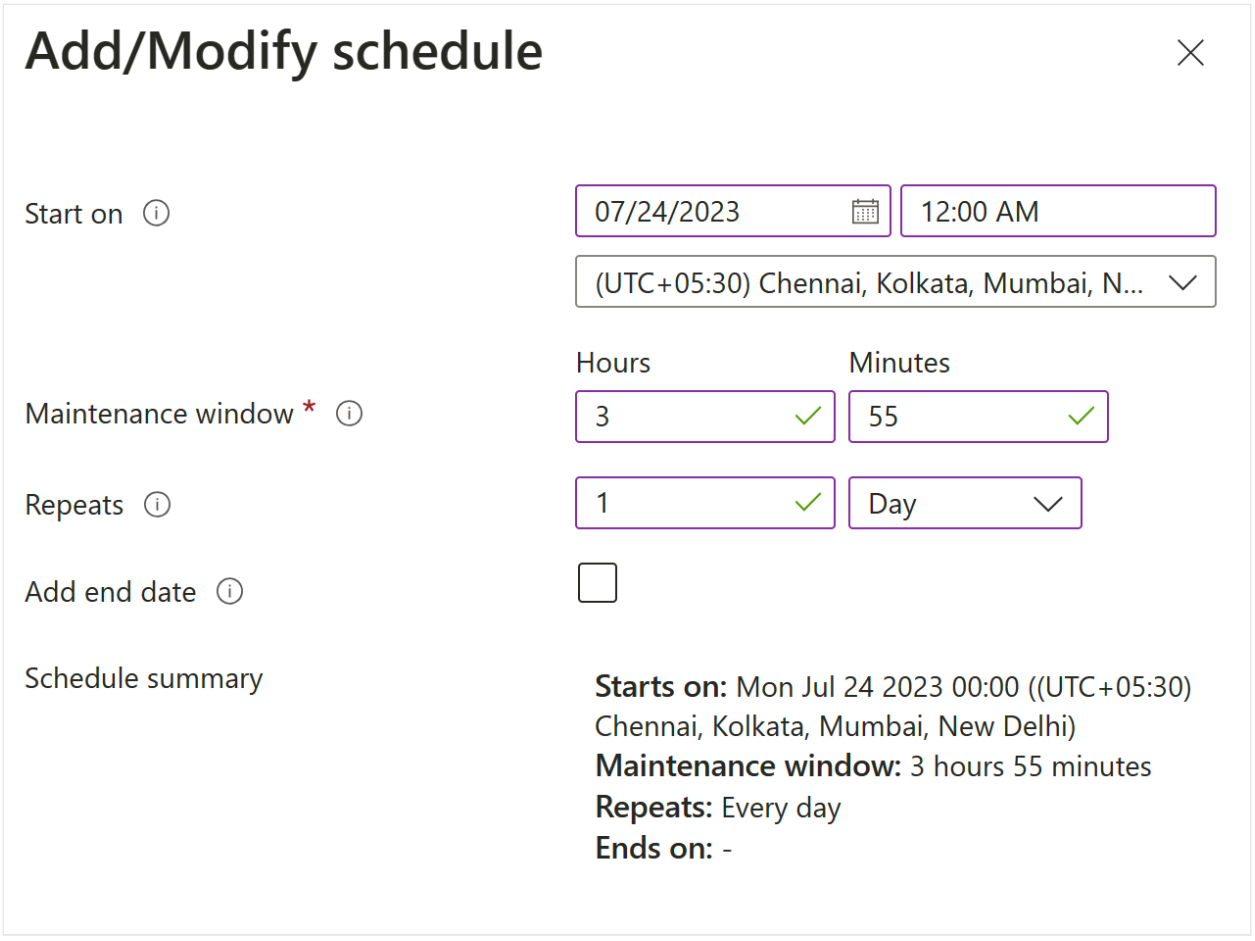
- For scale sets to use Maintenance Configurations, they need to have automatic OS upgrades enabled.
- You can schedule recurrence up to a week (7 days).
- A minimum of 5 hours is required for the maintenance window.

Guest

The guest scope integrates with Azure Update Manager. You can use it to save recurring deployment schedules to install updates for your Windows Server and Linux machines in Azure, in on-premises environments, and in other cloud environments connected through Azure Arc-enabled servers.

Features and limitations unique to this scope include:

- Patch orchestration for virtual machines needs to be set to `AutomaticByPlatform`.
- The upper maintenance window is 3 hours and 55 minutes.
- A minimum of 1 hour and 30 minutes is required for the maintenance window.
- The value of **Repeats** should be at least 6 hours.
- The start time for a schedule should be at least 15 minutes after the schedule's creation time.



Note

- The character count for the resource group name and the maintenance configuration name should be less than 128.
- For newly created schedules, there is a 15-minute delay in evaluating the trigger time, while for existing schedules, there is a 3-minute delay. Therefore, it is advisable to avoid creating schedules between 23:57 to 00:00 for existing schedules and between 23:45 to 00:00 for new schedules. This is because, after the evaluation stage of the trigger time, the system rechecks the start date and time before executing the job. If configured within these time limits, the evaluation and execution times differ, potentially resulting in the schedule not triggering or causing the VM to become unresponsive or shut down.

Maintenance Configurations provides two scheduled patching modes for VMs in the guest scope: Static Mode and Dynamic Scope Mode. By default, the system operates in Static Mode if you don't configure a Dynamic Scope Mode.

To schedule or modify the maintenance configuration in either mode, recommended buffer of 15 minutes before the scheduled patch time is required. For instance, if you

schedule the patching job for 3:00 PM, all modifications (including adding VMs, removing VMs, or altering the dynamic scope) should finish before 2:45 PM.

To learn more about this topic, see Schedule recurring updates for machines by using the Azure portal and Azure Policy.

> **Important**
>
> If you move a resource to a different resource group or subscription, scheduled patching for the resource stops working, because the system currently doesn't support this scenario. As a workaround, follow the steps in the **troubleshooting article**.

Shut-down machines

You can't apply maintenance updates to any shut-down machines. Ensure that your machine is turned on at least 15 minutes before a scheduled update, or the update might not be applied.

If your machine is shut down at the time of your scheduled update, the maintenance configuration might appear to be disassociated in the Azure portal. This is only a display issue. The maintenance configuration isn't disassociated, and you can check it via the Azure CLI.

Management options

You can create and manage maintenance configurations by using any of the following options:

- Azure CLI
- Azure PowerShell
- Azure portal

For an Azure Functions sample, see Scheduling maintenance updates with maintenance configurations and Azure Functions.

Service limits

We recommend the following limits for indicators:

Expand table

Indicator	Public Cloud Limit	Mooncake/Fairfax Limit
Number of schedules per subscription per region	250	250
Total number of resource associations to a schedule	3,000	3,000
Resource associations on each dynamic scope	1,000	1,000
Number of dynamic scopes per resource group or subscription per region	250	250
Number of dynamic scopes per schedule	200	100
Total number of subscriptions attached to all dynamic scopes per schedule	200	100

We recommend the following limits for each dynamic scope in the guest scope only:

Expand table

Resource	Limit
Resource associations	1,000
Number of tag filters	50
Number of resource group filters	50

> **Note**
>
> Maintenance configurations will initiate updates on all resources associated with it simultaneously. Customers have the flexibility to decide on the isolation and maintenance windows for their workloads. They can create single or multiple maintenance configurations based on the isolation that is optimal for their workloads. For example, if a customer wants to isolate updates for their development environment resources from the production environment resources, they can create separate maintenance configurations for each environment. This way, workloads in maintenance configuration with development resources are updated first, while the remaining workloads are updated in another configuration that does not overlap. This approach allows customers to control workload isolation and manage updates independently.

Next steps

- For troubleshooting, see Troubleshoot issues with Maintenance Configurations.
- To learn more, see Maintenance for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Control updates with Maintenance Configurations and the Azure CLI

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can use the Maintenance Configurations feature to control when to apply platform updates to various Azure resources. This article covers the Azure CLI options for using this feature. For more information about the benefits of using Maintenance Configurations, its limitations, and other management options, see Managing platform updates with Maintenance Configurations.

> **ⓘ Important**
>
> Specific *scopes* support certain machine types and schedules. Be sure to select the right scope for your virtual machine (VM).

Create a maintenance configuration

The first step in creating a maintenance configuration is creating a resource group as a container for your configuration. This example creates a resource group named *myMaintenanceRG* in *eastus*. If you already have a resource group that you want to use, you can skip this part and replace the resource group name with your own in the rest of the examples.

Azure CLI

```
az group create \
   --location eastus \
   --name myMaintenanceRG
```

After you create the resource group, use `az maintenance configuration create` to create a maintenance configuration.

Host

This example creates a maintenance configuration named *myConfig* scoped to host machines, with a scheduled window of 5 hours on the fourth Monday of every month:

Azure CLI

```
az maintenance configuration create \
   --resource-group myMaintenanceRG \
   --resource-name myConfig \
   --maintenance-scope host \
   --location eastus \
   --maintenance-window-duration "05:00" \
   --maintenance-window-recur-every "Month Fourth Monday" \
   --maintenance-window-start-date-time "2020-12-30 08:00" \
   --maintenance-window-time-zone "Pacific Standard Time"
```

Using `--maintenance-scope host` ensures that the maintenance configuration is used for controlling updates to the host infrastructure. If you try to create a configuration with the same name but in a different location, you'll get an error. Configuration names must be unique to your resource group.

To check if you successfully created the maintenance configuration, you can query for available maintenance configurations by using `az maintenance configuration list`:

Azure CLI

```
az maintenance configuration list
   --query "[].{Name:name, ID:id}"
   --output table
```

You can express maintenance recurrence as daily, weekly, or monthly. Here are some examples:

- **Daily**: A `maintenance-window-recur-every` value of `"Day"` or `"3Days"`.
- **Weekly**: A `maintenance-window-recur-every` value of `"3Weeks"` or `"Week Saturday,Sunday"`.
- **Monthly**: A `maintenance-window-recur-every` value of `"Month day23,day24"` or `"Month Last Sunday"` or `Month Fourth Monday`.

Virtual machine scale sets

This example creates a maintenance configuration named *myConfig* with the OS image scope for virtual machine scale sets, with a scheduled window of 5 hours on the fourth Monday of every month:

Azure CLI

```
az maintenance configuration create \
   --resource-group myMaintenanceRG \
```

```
    --resource-name myConfig \
    --maintenance-scope osimage \
    --location eastus \
    --maintenance-window-duration "05:00" \
    --maintenance-window-recur-every "Month Fourth Monday" \
    --maintenance-window-start-date-time "2020-12-30 08:00" \
    --maintenance-window-time-zone "Pacific Standard Time"
```

Guest VMs

This example creates a maintenance configuration named *myConfig* scoped to guest machines (VMs and Azure Arc-enabled servers), with a scheduled window of 2 hours every 20 days. Learn more about maintenance configurations on guest VMs.

Azure CLI

```
az maintenance configuration create \
    --resource-group myMaintenanceRG \
    --resource-name myConfig \
    --maintenance-scope InGuestPatch \
    --location eastus \
    --maintenance-window-duration "02:00" \
    --maintenance-window-recur-every "20days" \
    --maintenance-window-start-date-time "2022-12-30 07:00" \
    --maintenance-window-time-zone "Pacific Standard Time" \
    --install-patches-linux-parameters package-name-masks-to-exclude="ppt"
package-name-masks-to-include="apt" classifications-to-include="Other" \
    --install-patches-windows-parameters kb-numbers-to-exclude="KB123456" kb-
numbers-to-include="KB123456" classifications-to-include="FeaturePack" \
    --reboot-setting "IfRequired" \
    --extension-properties InGuestPatchMode="User"
```

Assign the configuration

Use `az maintenance assignment create` to assign the configuration to your machine.

Isolated VM

Apply the configuration to an isolated host VM by using the ID of the configuration. Specify `--resource-type virtualMachines`. Supply the name of the VM for `--resource-name`, the VM's resource group for `--resource-group`, and the location of the VM for `--location`.

Azure CLI

```
az maintenance assignment create \
   --resource-group myMaintenanceRG \
   --location eastus \
   --resource-name myVM \
   --resource-type virtualMachines \
   --provider-name Microsoft.Compute \
   --configuration-assignment-name myConfig \
   --maintenance-configuration-id "/subscriptions/{subscription
ID}/resourcegroups/myMaintenanceRG/providers/Microsoft.Maintenance/maintenan
ceConfigurations/myConfig"
```

Dedicated host

To apply a configuration to a dedicated host, you need to include `--resource-type hosts`, `--resource-parent-name` with the name of the host group, and `--resource-parent-type hostGroups`.

The parameter `--resource-id` is the ID of the host. You can use az-vm-host-get-instance-view to get the ID of your dedicated host.

Azure CLI

```
az maintenance assignment create \
   --resource-group myDHResourceGroup \
   --resource-name myHost \
   --resource-type hosts \
   --provider-name Microsoft.Compute \
   --configuration-assignment-name myConfig \
   --maintenance-configuration-id "/subscriptions/{subscription
ID}/resourcegroups/myDhResourceGroup/providers/Microsoft.Maintenance/mainten
anceConfigurations/myConfig" \
   --location eastus \
   --resource-parent-name myHostGroup \
   --resource-parent-type hostGroups
```

Virtual machine scale sets

Azure CLI

```
az maintenance assignment create \
   --resource-group myMaintenanceRG \
   --location eastus \
   --resource-name myVMSS \
   --resource-type virtualMachineScaleSets \
   --provider-name Microsoft.Compute \
   --configuration-assignment-name myConfig \
   --maintenance-configuration-id "/subscriptions/{subscription
```

```
ID}/resourcegroups/myMaintenanceRG/providers/Microsoft.Maintenance/maintenan
ceConfigurations/myConfig"
```

Guest VMs

Azure CLI

```
az maintenance assignment create \
   --resource-group myMaintenanceRG \
   --location eastus \
   --resource-name myVM \
   --resource-type virtualMachines \
   --provider-name Microsoft.Compute \
   --configuration-assignment-name myConfig \
   --maintenance-configuration-id "/subscriptions/{subscription
ID}/resourcegroups/myMaintenanceRG/providers/Microsoft.Maintenance/maintenan
ceConfigurations/myConfig"
```

Check the configuration

You can verify that the configuration was applied correctly, or check to see what configuration is currently applied, by using `az maintenance assignment list`.

Isolated VM

Azure CLI

```
az maintenance assignment list \
   --provider-name Microsoft.Compute \
   --resource-group myMaintenanceRG \
   --resource-name myVM \
   --resource-type virtualMachines \
   --query "[].{resource:resourceGroup, configName:name}" \
   --output table
```

Dedicated host

Azure CLI

```
az maintenance assignment list \
   --resource-group myDHResourceGroup \
   --resource-name myHost \
   --resource-type hosts \
   --provider-name Microsoft.Compute \
```

```
    --resource-parent-name myHostGroup \
    --resource-parent-type hostGroups \
    --query "[].{ResourceGroup:resourceGroup,configName:name}" \
    --output table
```

Virtual machine scale sets

Azure CLI

```
az maintenance assignment list \
    --provider-name Microsoft.Compute \
    --resource-group myMaintenanceRG \
    --resource-name myVMSS \
    --resource-type virtualMachines \
    --query "[].{resource:resourceGroup, configName:name}" \
    --output table
```

Guest VMs

Azure CLI

```
az maintenance assignment list \
    --provider-name Microsoft.Compute \
    --resource-group myMaintenanceRG \
    --resource-name myVM \
    --resource-type virtualMachines \
    --query "[].{resource:resourceGroup, configName:name}" \
    --output table
```

Check for pending updates

Use `az maintenance update list` to see if there are pending updates. Update `--subscription` to be the ID for the subscription that contains the VM.

If there are no updates, the command returns an error message that contains the text `Resource not found...StatusCode: 404`.

If there are updates, the command returns only one, even if multiple updates are pending. The data for this update is returned in an object:

text

```
[
  {
```

```
    "impactDurationInSec": 9,
    "impactType": "Freeze",
    "maintenanceScope": "Host",
    "notBefore": "2020-03-03T07:23:04.905538+00:00",
    "resourceId": "/subscriptions/9120c5ff-e78e-4bd0-b29f-
75c19cadd078/resourcegroups/DemoRG/providers/Microsoft.Compute/hostGroups/de
moHostGroup/hosts/myHost",
    "status": "Pending"
  }
]
```

Isolated VM

Check for pending updates for an isolated VM. In this example, the output is formatted as a table for readability:

Azure CLI

```azurecli
az maintenance update list \
   --subscription {subscription ID} \
   --resource-group myMaintenanceRg \
   --resource-name myVM \
   --resource-type virtualMachines \
   --provider-name Microsoft.Compute \
   --output table
```

Dedicated host

Check for pending updates for a dedicated host. In this example, the output is formatted as a table for readability. Replace the values for the resources with your own.

Azure CLI

```azurecli
az maintenance update list \
   --subscription {subscription ID} \
   --resource-group myHostResourceGroup \
   --resource-name myHost \
   --resource-type hosts \
   --provider-name Microsoft.Compute \
   --resource-parentname myHostGroup \
   --resource-parent-type hostGroups \
   --output table
```

Apply updates

Use `az maintenance apply update` to apply pending updates. On success, this command returns JSON that contains the details of the update. Calls to apply updates can take up to 2 hours to complete.

Isolated VM

Create a request to apply updates to an isolated VM:

Azure CLI

```
az maintenance applyupdate create \
   --subscription {subscriptionID} \
   --resource-group myMaintenanceRG \
   --resource-name myVM \
   --resource-type virtualMachines \
   --provider-name Microsoft.Compute
```

Dedicated host

Apply updates to a dedicated host:

Azure CLI

```
az maintenance applyupdate create \
   --subscription {subscriptionID} \
   --resource-group myHostResourceGroup \
   --resource-name myHost \
   --resource-type hosts \
   --provider-name Microsoft.Compute \
   --resource-parent-name myHostGroup \
   --resource-parent-type hostGroups
```

Virtual machine scale sets

Apply updates to a scale set:

Azure CLI

```
az maintenance applyupdate create \
   --subscription {subscriptionID} \
   --resource-group myMaintenanceRG \
   --resource-name myVMSS \
   --resource-type virtualMachineScaleSets \
   --provider-name Microsoft.Compute
```

Check the status of applying updates

You can check on the progress of the updates by using `az maintenance applyupdate get`.

To see results for the last update, use `default` as the update name. Or replace `myUpdateName` with the name of the update that was returned when you ran `az maintenance applyupdate create`.

text

```
Status         : Completed
ResourceId     : /subscriptions/12ae7457-4a34-465c-94c1-
17c058c2bd25/resourcegroups/TestShantS/providers/Microsoft.Comp
ute/virtualMachines/DXT-test-04-iso
LastUpdateTime : 1/1/2020 12:00:00 AM
Id             : /subscriptions/12ae7457-4a34-465c-94c1-
17c058c2bd25/resourcegroups/TestShantS/providers/Microsoft.Comp
ute/virtualMachines/DXT-test-04-
iso/providers/Microsoft.Maintenance/applyUpdates/default
Name           : default
Type           : Microsoft.Maintenance/applyUpdates
```

`LastUpdateTime` is the time when the update finished, whether you initiated the update or the platform initiated it because you didn't use the self-maintenance window. If an update was never applied through Maintenance Configurations, `LastUpdateTime` shows the default value.

Isolated VM

Azure CLI

```
az maintenance applyupdate get \
   --subscription {subscriptionID} \
   --resource-group myMaintenanceRG \
   --resource-name myVM \
   --resource-type virtualMachines \
   --provider-name Microsoft.Compute \
   --apply-update-name myUpdateName \
   --query "{LastUpdate:lastUpdateTime, Name:name,
ResourceGroup:resourceGroup, Status:status}" \
   --output table
```

Dedicated host

Azure CLI

```
az maintenance applyupdate get \
   --subscription {subscriptionID} \
   --resource-group myMaintenanceRG \
   --resource-name myHost \
   --resource-type hosts \
   --provider-name Microsoft.Compute \
   --resource-parent-name myHostGroup \
   --resource-parent-type hostGroups \
   --apply-update-name myUpdateName \
   --query "{LastUpdate:lastUpdateTime, Name:name,
ResourceGroup:resourceGroup, Status:status}" \
   --output table
```

Virtual machine scale sets

Azure CLI

```
az maintenance applyupdate get \
   --subscription {subscriptionID} \
   --resource-group myMaintenanceRG \
   --resource-name myVMSS \
   --resource-type virtualMachineScaleSets \
   --provider-name Microsoft.Compute \
   --apply-update-name myUpdateName \
   --query "{LastUpdate:lastUpdateTime, Name:name,
ResourceGroup:resourceGroup, Status:status}" \
   --output table
```

Delete a maintenance configuration

To delete a maintenance configuration, use `az maintenance configuration delete`. Deleting the configuration removes the maintenance control from the associated resources.

Azure CLI

```
az maintenance configuration delete \
   --subscription 1111abcd-1a11-1a2b-1a12-123456789abc \
   -resource-group myResourceGroup \
   --resource-name myConfig
```

Next steps

To learn more, see Maintenance for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Control updates with Maintenance Configurations and Azure PowerShell

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can use the Maintenance Configurations feature to control when to apply platform updates to various Azure resources. This article covers the Azure PowerShell options for dedicated hosts and isolated virtual machines (VMs). For more information about the benefits of using the Maintenance Configurations feature, its limitations, and other management options, see Managing platform updates with Maintenance Configurations.

If you're looking for information about using Maintenance Configurations for scale sets, see Maintenance control for Azure Virtual Machine Scale Sets.

> **ⓘ Important**
>
> Specific *scopes* support certain machine types and schedules. Be sure to select the right scope for your VM.

Enable the Azure PowerShell module

Make sure that `PowerShellGet` is up to date:

Azure PowerShell

```
Install-Module -Name PowerShellGet -Repository PSGallery -Force
```

Install the `Az.Maintenance` Azure PowerShell module:

Azure PowerShell

```
Install-Module -Name Az.Maintenance
```

Check that you're running the latest version of `Az.Maintenance` (version 1.2.0):

Azure PowerShell

```
Get-Module -ListAvailable -Name Az.Maintenance
```

Ensure that you're running the appropriate version of `Az.Maintenance`:

Azure PowerShell

```
Import-Module -Name Az.Maintenance -RequiredVersion 1.2.0
```

If you're installing locally, be sure to open your Azure PowerShell prompt as an administrator.

You might be asked to confirm that you want to install from an untrusted repository. Enter **Y** or select **Yes to All** to install the module.

Create a maintenance configuration

The first step in creating a maintenance configuration is creating a resource group as a container for your configuration. This example creates a resource group named *myMaintenanceRG* in *eastus*. If you already have a resource group that you want to use, you can skip this part and replace the resource group name with your own in the rest of the examples.

Azure PowerShell

```
New-AzResourceGroup `
   -Location eastus `
   -Name myMaintenanceRG
```

You can declare a scheduled window when Azure will recurrently apply the updates on your resources. After you create a scheduled window, you no longer have to apply the updates manually.

You can express maintenance recurrence as daily, weekly, or monthly. Here are some examples:

- **Daily**: A `RecurEvery` value of `"Day"` or `"3Days"`.
- **Weekly**: A `RecurEvery` value of `"3Weeks"` or `"Week Saturday,Sunday"`.
- **Monthly**: A `RecurEvery` value of `"Month day23,day24"` or `"Month Last Sunday"` or `"Month Fourth Monday"`.

Host

This example creates a maintenance configuration named *myConfig* scoped to `Host`, with a scheduled window of 5 hours on the fourth Monday of every month. The `duration` value of the schedule for this scope should be at least two hours. To begin, define the parameters for `New-AzMaintenanceConfiguration`:

Azure PowerShell

```
$RGName = "myMaintenanceRG"
$configName = "myConfig"
$scope = "Host"
$location = "eastus"
$timeZone = "Pacific Standard Time"
$duration = "05:00"
$startDateTime = "2022-11-01 00:00"
$recurEvery = "Month Fourth Monday"
```

After you define the parameters, you can use the `New-AzMaintenanceConfiguration` cmdlet to create your configuration:

Azure PowerShell

```
New-AzMaintenanceConfiguration
   -ResourceGroup $RGName `
   -Name $configName `
   -MaintenanceScope $scope `
   -Location $location `
   -StartDateTime $startDateTime `
   -TimeZone $timeZone `
   -Duration $duration `
   -RecurEvery $recurEvery
```

Using `$scope = "Host"` ensures that the maintenance configuration is used for controlling updates on host machines. Be sure to create a configuration for the specific scope of the machines that you're targeting. Learn more about scopes.

OS image

This example creates a maintenance configuration named *myConfig* scoped to `osimage`, with a scheduled window of 8 hours every 5 days. The `duration` value of the schedule for this scope should be at least 5 hours. This scope allows a maximum of 7 days for schedule recurrence.

Azure PowerShell

```
$RGName = "myMaintenanceRG"
$configName = "myConfig"
```

```
$scope = "osimage"
$location = "eastus"
$timeZone = "Pacific Standard Time"
$duration = "08:00"
$startDateTime = "2022-11-01 00:00"
$recurEvery = "5days"
```

After you define the parameters, you can use the `New-AzMaintenanceConfiguration` cmdlet to create your configuration:

Azure PowerShell

```
New-AzMaintenanceConfiguration
   -ResourceGroup $RGName `
   -Name $configName `
   -MaintenanceScope $scope `
   -Location $location `
   -StartDateTime $startDateTime `
   -TimeZone $timeZone `
   -Duration $duration `
   -RecurEvery $recurEvery
```

Guest

The most recent addition to the Maintenance Configurations feature is the `InGuestPatch` scope. This example shows how to create a maintenance configuration for a guest scope by using Azure PowerShell. For more information about this scope, see Guest.

Azure PowerShell

```
$RGName = "myMaintenanceRG"
$configName = "myConfig"
$scope = "InGuestPatch"
$location = "eastus"
$timeZone = "Pacific Standard Time"
$duration = "04:00"
$startDateTime = "2022-11-01 00:00"
$recurEvery = "Week Saturday, Sunday"
$WindowsParameterClassificationToInclude = "FeaturePack","ServicePack";
$WindowParameterKbNumberToInclude = "KB123456","KB123466";
$WindowParameterKbNumberToExclude = "KB123456","KB123466";
$RebootOption = "IfRequired";
$LinuxParameterClassificationToInclude = "Other";
$LinuxParameterPackageNameMaskToInclude = "apt","httpd";
$LinuxParameterPackageNameMaskToExclude = "ppt","userpk";
```

After you define the parameters, you can use the `New-AzMaintenanceConfiguration` cmdlet to create your configuration:

Azure PowerShell

```
New-AzMaintenanceConfiguration
   -ResourceGroup $RGName `
   -Name $configName `
   -MaintenanceScope $scope `
   -Location $location `
   -StartDateTime $startDateTime `
   -TimeZone $timeZone `
   -Duration $duration `
   -RecurEvery $recurEvery `
   -WindowParameterClassificationToInclude
$WindowsParameterClassificationToInclude `
   -WindowParameterKbNumberToInclude $WindowParameterKbNumberToInclude `
   -WindowParameterKbNumberToExclude $WindowParameterKbNumberToExclude `
   -InstallPatchRebootSetting $RebootOption `
   -LinuxParameterPackageNameMaskToInclude
$LinuxParameterPackageNameMaskToInclude `
   -LinuxParameterClassificationToInclude
$LinuxParameterClassificationToInclude `
   -LinuxParameterPackageNameMaskToExclude
$LinuxParameterPackageNameMaskToExclude `
   -ExtensionProperty @{"InGuestPatchMode"="User"}
```

If you try to create a configuration with the same name but in a different location, you'll get an error. Configuration names must be unique to your resource group.

You can check if you successfully created the maintenance configurations by using Get-AzMaintenanceConfiguration:

Azure PowerShell

```
Get-AzMaintenanceConfiguration | Format-Table -Property Name,Id
```

Assign the configuration

After you create your configuration, you might want to also assign machines to it by using Azure PowerShell. You can use the New-AzConfigurationAssignment cmdlet.

Isolated VM

Assign the configuration to a VM by using the ID of the configuration. Specify `-ResourceType VirtualMachines`. Supply the name of the VM for `-ResourceName`, and

supply the resource group of the VM for `-ResourceGroupName`.

Azure PowerShell

```
New-AzConfigurationAssignment `
   -ResourceGroupName "myResourceGroup" `
   -Location "eastus" `
   -ResourceName "myVM" `
   -ResourceType "VirtualMachines" `
   -ProviderName "Microsoft.Compute" `
   -ConfigurationAssignmentName "configName" `
   -MaintenanceConfigurationId "configID"
```

Dedicated host

To apply a configuration to a dedicated host, you need to include `-ResourceType hosts`, `-ResourceParentName` with the name of the host group, and `-ResourceParentType hostGroups`:

Azure PowerShell

```
New-AzConfigurationAssignment `
   -ResourceGroupName "myResourceGroup" `
   -Location "eastus" `
   -ResourceName "myHost" `
   -ResourceType "hosts" `
   -ResourceParentName myHostGroup `
   -ResourceParentType hostGroups `
   -ProviderName "Microsoft.Compute" `
   -ConfigurationAssignmentName "configName" `
   -MaintenanceConfigurationId "configID"
```

Virtual machine scale sets

Azure PowerShell

```
New-AzConfigurationAssignment `
   -ResourceGroupName "myResourceGroup" `
   -Location "eastus" `
   -ResourceName "myVMSS" `
   -ResourceType "VirtualMachineScaleSets" `
   -ProviderName "Microsoft.Compute" `
   -ConfigurationAssignmentName "configName" `
   -MaintenanceConfigurationId "configID"
```

Guest

Azure PowerShell

```
New-AzConfigurationAssignment `
   -ResourceGroupName "myResourceGroup" `
   -Location "eastus" `
   -ResourceName "myGuest" `
   -ResourceType "VirtualMachines" `
   -ProviderName "Microsoft.Compute" `
   -ConfigurationAssignmentName "configName" `
   -MaintenanceConfigurationId "configID"
```

Check for pending updates

The check for pending updates, use Get-AzMaintenanceUpdate. Use `-subscription` to specify the Azure subscription of the VM, if it's different from the one that you're logged in to.

If there are no updates to show, this command returns nothing. Otherwise, it returns a `PSApplyUpdate` object:

JSON

```
{
   "maintenanceScope": "Host",
   "impactType": "Freeze",
   "status": "Pending",
   "impactDurationInSec": 9,
   "notBefore": "2020-02-21T16:47:44.8728029Z",
   "properties": {
      "resourceId": "/subscriptions/39c6cced-4d6c-4dd5-af86-
57499cd3f846/resourcegroups/Ignite2019/providers/Microsoft.Compute/virtualMa
chines/MCDemo3"
}
```

Isolated VM

Check for pending updates for an isolated VM. In this example, the output is formatted as a table for readability:

Azure PowerShell

```
Get-AzMaintenanceUpdate `
  -ResourceGroupName "myResourceGroup" `
  -ResourceName "myVM" `
```

```
    -ResourceType "VirtualMachines" `
    -ProviderName "Microsoft.Compute" | Format-Table
```

Dedicated host

Check for pending updates for a dedicated host. In this example, the output is formatted as a table for readability:

Azure PowerShell

```
Get-AzMaintenanceUpdate `
    -ResourceGroupName "myResourceGroup" `
    -ResourceName "myHost" `
    -ResourceType "hosts" `
    -ResourceParentName "myHostGroup" `
    -ResourceParentType "hostGroups" `
    -ProviderName "Microsoft.Compute" | Format-Table
```

Virtual machine scale sets

Azure PowerShell

```
Get-AzMaintenanceUpdate `
    -ResourceGroupName "myResourceGroup" `
    -Location "eastus" `
    -ResourceName "myVMSS" `
    -ResourceType "VirtualMachineScaleSets" `
    -ProviderName "Microsoft.Compute" | Format-Table
```

Apply updates

Use New-AzApplyUpdate to apply pending updates. Applying update calls can take up to 2 hours to complete.

This cmdlet works only for the host and OS image scopes. It doesn't work for the guest scope.

Isolated VM

Create a request to apply updates to an isolated VM:

Azure PowerShell

```
New-AzApplyUpdate `
  -ResourceGroupName "myResourceGroup" `
  -ResourceName "myVM" `
  -ResourceType "VirtualMachines" `
  -ProviderName "Microsoft.Compute"
```

On success, this command returns a `PSApplyUpdate` object. You can use the `Name` attribute in the `Get-AzApplyUpdate` command to check the update status, as described later in this article.

Dedicated host

Apply updates to a dedicated host:

Azure PowerShell

```
New-AzApplyUpdate `
   -ResourceGroupName "myResourceGroup" `
   -ResourceName "myHost" `
   -ResourceType "hosts" `
   -ResourceParentName "myHostGroup" `
   -ResourceParentType "hostGroups" `
   -ProviderName Microsoft.Compute
```

Virtual machine scale sets

Azure PowerShell

```
New-AzApplyUpdate `
   -ResourceGroupName "myResourceGroup" `
   -Location "eastus" `
   -ResourceName "myVMSS" `
   -ResourceType "VirtualMachineScaleSets" `
   -ProviderName "Microsoft.Compute"
```

Check update status

To check the status of an update, use Get-AzApplyUpdate. The following commands show the status of the latest update by using `default` for the `-ApplyUpdateName` parameter. You can substitute the name of the update (returned by the New-AzApplyUpdate command) to get the status of a specific update.

This cmdlet works only for the host and OS image scopes. It doesn't work for the guest scope.

text

```
Status         : Completed
ResourceId     : /subscriptions/12ae7457-4a34-465c-94c1-
17c058c2bd25/resourcegroups/TestShantS/providers/Microsoft.Comp
ute/virtualMachines/DXT-test-04-iso
LastUpdateTime : 1/1/2020 12:00:00 AM
Id             : /subscriptions/12ae7457-4a34-465c-94c1-
17c058c2bd25/resourcegroups/TestShantS/providers/Microsoft.Comp
ute/virtualMachines/DXT-test-04-
iso/providers/Microsoft.Maintenance/applyUpdates/default
Name           : default
Type           : Microsoft.Maintenance/applyUpdates
```

`LastUpdateTime` is the time when the update finished, whether you initiated the update or the platform initiated it because you didn't use the self-maintenance window. If an update was never applied through Maintenance Configurations, `LastUpdateTime` shows the default value.

Isolated VM

Check for updates to a specific virtual machine:

Azure PowerShell

```
Get-AzApplyUpdate `
  -ResourceGroupName "myResourceGroup" `
  -ResourceName "myVM" `
  -ResourceType "VirtualMachines" `
  -ProviderName "Microsoft.Compute" `
  -ApplyUpdateName "applyUpdateName"
```

Dedicated host

Check for updates to a dedicated host:

Azure PowerShell

```
Get-AzApplyUpdate `
   -ResourceGroupName "myResourceGroup" `
   -ResourceName "myHost" `
   -ResourceType "hosts" `
   -ResourceParentName "myHostGroup" `
```

```
    -ResourceParentType "hostGroups" `
    -ProviderName "Microsoft.Compute" `
    -ApplyUpdateName "applyUpdateName"
```

Virtual machine scale sets

Azure PowerShell

```
New-AzApplyUpdate `
    -ResourceGroupName "myResourceGroup" `
    -Location "eastus" `
    -ResourceName "myVMSS" `
    -ResourceType "VirtualMachineScaleSets" `
    -ProviderName "Microsoft.Compute" `
    -ApplyUpdateName "applyUpdateName"
```

Delete a maintenance configuration

To delete a maintenance configuration, use Remove-AzMaintenanceConfiguration:

Azure PowerShell

```
Remove-AzMaintenanceConfiguration `
    -ResourceGroupName "myResourceGroup" `
    -Name "configName"
```

Next steps

To learn more, see Maintenance for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Control updates with Maintenance Configurations and Bicep

Article • 11/11/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

You can use the Maintenance Configurations feature to control when to apply platform updates to various Azure resources. This article covers the Bicep language for using this feature. For more information about the benefits of using the Maintenance Configurations feature, its limitations, and other management options, see Managing platform updates with Maintenance Configurations.

> **Important**
>
> Specific *scopes* support certain machine types and schedules. Be sure to select the right scope for your VM.

Prerequisites

Before proceeding, visit Bicep Modules to learn more about how bicep deployment can be organized into modules

- Scope of resource deployment should be understood before proceeding with the Bicep template. For more information, see Scope Functions for Bicep.
- It's essential to know how and where resources are deployed.
- Bicep modules must be used to perform deployments. Deploying resources at different scopes require the use of modules.

> **Tip**
>
> Always provide the right set of dependencies since module execute on different scopes. For example - the scope for configuration assignment in maintenance configuration when using dynamic scope should be `Subscription` and when using static assignment (no dynamic scopes configured) should be `ResourceId`

Bicep resource definition

Refer to the below guides for the resource format and resource definition of the Bicep template before proceeding ahead with the working example.

- Maintenance Configurations
- Configuration Assignments
- Apply Updates

Example

There are three bicep files that need to be created. We recommend keeping all the three files in the same folder. You can also keep them in different folders but make sure to update the path of the file names in `example.bicep` file mentioned.

Template execution

PowerShell

```
az deployment group create --name samplebicep --template-file .\ds.bicep --
resource-group  testrg --parameters '{}' --debug --subscription abc4def8-
er57-2345-f4t8-dff65f67afr3
```

Example template

example.bicep

Bicep

```
metadata description = 'Demo Main'

module mrpmc1 './maintenanceconfig.bicep' = {

  name: 'mrpmc'

  params: {

   configName: 'test-MaintConfig-Bicep'

  }

}

module ca1 './config_assignment.bicep' = {
```

```
  name: 'mrpmcconfigassign2'

  scope: subscription()

  params : {

  configId:
'${resourceGroup().id}/providers/microsoft.maintenance/maintenanceConfigurat
ions/test-MaintConfig-Bicep'

  }

  dependsOn: [

    mrpmc1

  ]

}
```

maintenanceconfig.bicep

Bicep

```
metadata description = 'Demo1'

param location string = 'Canada Central'

param configName string = ''

targetScope = 'resourceGroup'

resource MaintConfig4 'Microsoft.Maintenance/maintenanceConfigurations@2023-
04-01' = {

  name: configName //'test-MaintConfig-Bicep'

  location: location

  tags: null

  properties: {

    extensionProperties: {

      InGuestPatchMode: 'user'

    }

    installPatches: {

      linuxParameters: {
```

```
        classificationsToInclude: [
          'Critical'
          'Security'
        ]
        packageNameMasksToExclude: null
        packageNameMasksToInclude: null
      }
      rebootSetting: 'RebootIfRequired'
      windowsParameters: {
        classificationsToInclude: [
          'Critical'
          'Security'
        ]
      //  excludeKbsRequiringReboot: false
        kbNumbersToExclude: null
        kbNumbersToInclude: null
      }
    }
    maintenanceScope: 'InGuestPatch'
    maintenanceWindow: {
      duration: '02:00'
      expirationDateTime: '9999-12-31 23:59:59'
      recurEvery: 'Day'
      startDateTime: '2023-12-29 12:00'
      timeZone: 'Mountain Standard Time'
    }
  //  namespace: 'string'
```

```
   //visibility: 'public'

  }

}

output mrpConfId string = MaintConfig4.id
```

config_assignment.bicep

Bicep

```
metadata description = 'Demo'

param location string = 'Canada Central'

param configId string = ''

targetScope = 'subscription'

resource symbolicname 'Microsoft.Maintenance/configurationAssignments@2023-04-01' = {

  name: 'myconfig'

  location: 'global'

  scope: subscription()

  properties: {

    filter: {

      locations: []

      osTypes: [

        'Linux'

        'Windows'

      ]

     // resourceGroups: null

      resourceTypes: [

        'microsoft.compute/virtualmachines'

        'microsoft.hybridcompute/machines'

      ]
```

```
      resourceGroups:[
        'RG-MaintConfigs'
      ]
     // tagSettings:{
    //  filterOperator: 'All'
     //   tags: {updates: ['true']}
     // }
    }
    maintenanceConfigurationId: configId
    //resourceId: ''

  }
}
```

Sample output

Bicep


```
{
  "id": "/subscriptions/eee2cef4-bc47-4278-b4f8-
cfc65f25dfd8/resourceGroups/testrg/providers/Microsoft.Resources/deployments
/samplebicep",
  "location": null,
  "name": "samplebicep",
  "properties": {
    "correlationId": "b125fc6f-f771-46d7-9b88-b31e0da959f5",
    "debugSetting": null,
    "dependencies": [
      {
        "dependsOn": [
          {
            "id": "/subscriptions/eee2cef4-bc47-4278-b4f8-
cfc65f25dfd8/resourceGroups/testrg/providers/Microsoft.Resources/deployments
/mrpmc",
            "resourceGroup": "testrg",
            "resourceName": "mrpmc",
            "resourceType": "Microsoft.Resources/deployments"
          }
        ],
```

```
        "id": "/subscriptions/eee2cef4-bc47-4278-b4f8-
cfc65f25dfd8/providers/Microsoft.Resources/deployments/mrpmcconfigassign2",
        "resourceName": "mrpmcconfigassign2",
        "resourceType": "Microsoft.Resources/deployments"
      }
    ],
    "duration": "PT51.5773913S",
    "error": null,
    "mode": "Incremental",
    "onErrorDeployment": null,
    "outputResources": [
      {
        "id": "/subscriptions/eee2cef4-bc47-4278-b4f8-
cfc65f25dfd8/providers/Microsoft.Maintenance/configurationAssignments/myconf
ig"
      },
      {
        "id": "/subscriptions/eee2cef4-bc47-4278-b4f8-
cfc65f25dfd8/resourceGroups/testrg/providers/Microsoft.Maintenance/maintenan
ceConfigurations/test-MaintConfig-Bicep",
        "resourceGroup": "testrg"
      }
    ],
    "outputs": null,
    "parameters": null,
    "parametersLink": null,
    "providers": [
      {
        "id": null,
        "namespace": "Microsoft.Resources",
        "providerAuthorizationConsentState": null,
        "registrationPolicy": null,
        "registrationState": null,
        "resourceTypes": [
          {
            "aliases": null,
            "apiProfiles": null,
            "apiVersions": null,
            "capabilities": null,
            "defaultApiVersion": null,
            "locationMappings": null,
            "locations": [
              null,
              "canadacentral"
            ],
            "properties": null,
            "resourceType": "deployments",
            "zoneMappings": null
          }
        ]
      }
    ],
    "provisioningState": "Succeeded",
    "templateHash": "9584585695667229579",
    "templateLink": null,
```

```
    "timestamp": "2024-02-23T12:22:14.801966+00:00",
    "validatedResources": null
  },
  "resourceGroup": "testrg",
  "tags": null,
  "type": "Microsoft.Resources/deployments"
}
```

Next steps

To learn more, see Maintenance for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Control updates with Maintenance Configurations and the Azure portal

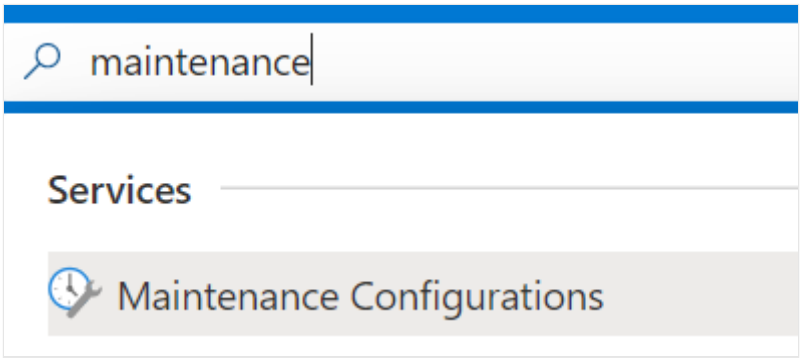
Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

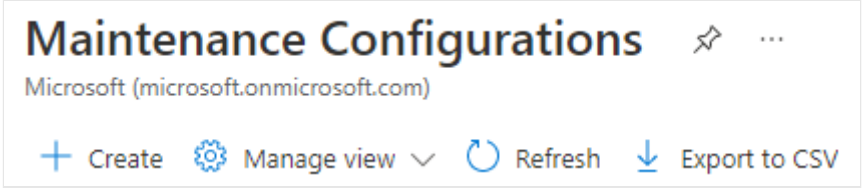
With the Maintenance Configurations feature, you can control when to apply updates to various Azure resources. This article covers the Azure portal options for using this feature. For more information about the benefits of using Maintenance Configurations, its limitations, and other management options, see Managing platform updates with Maintenance Configurations.

Create a maintenance configuration

1. Sign in to the Azure portal.
2. Search for **maintenance configurations**, and then open the **Maintenance Configurations** result.

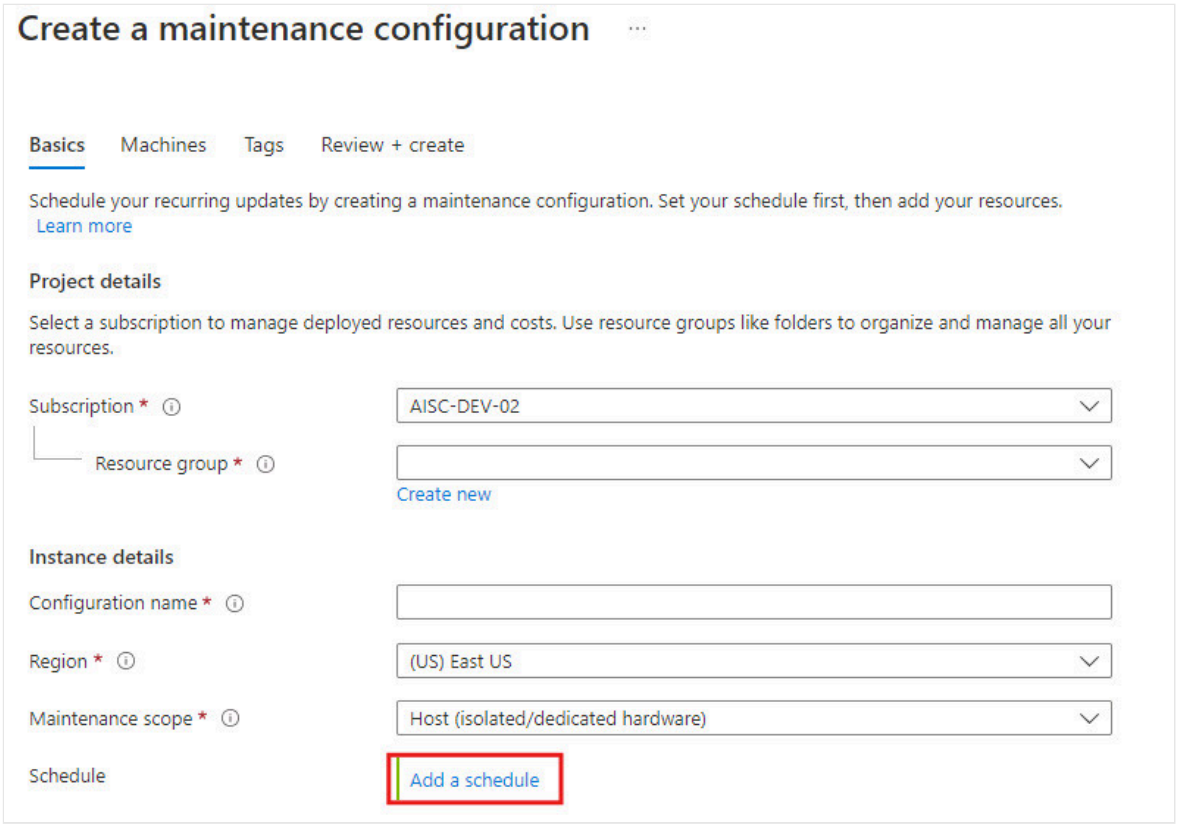


3. Select **Create**.



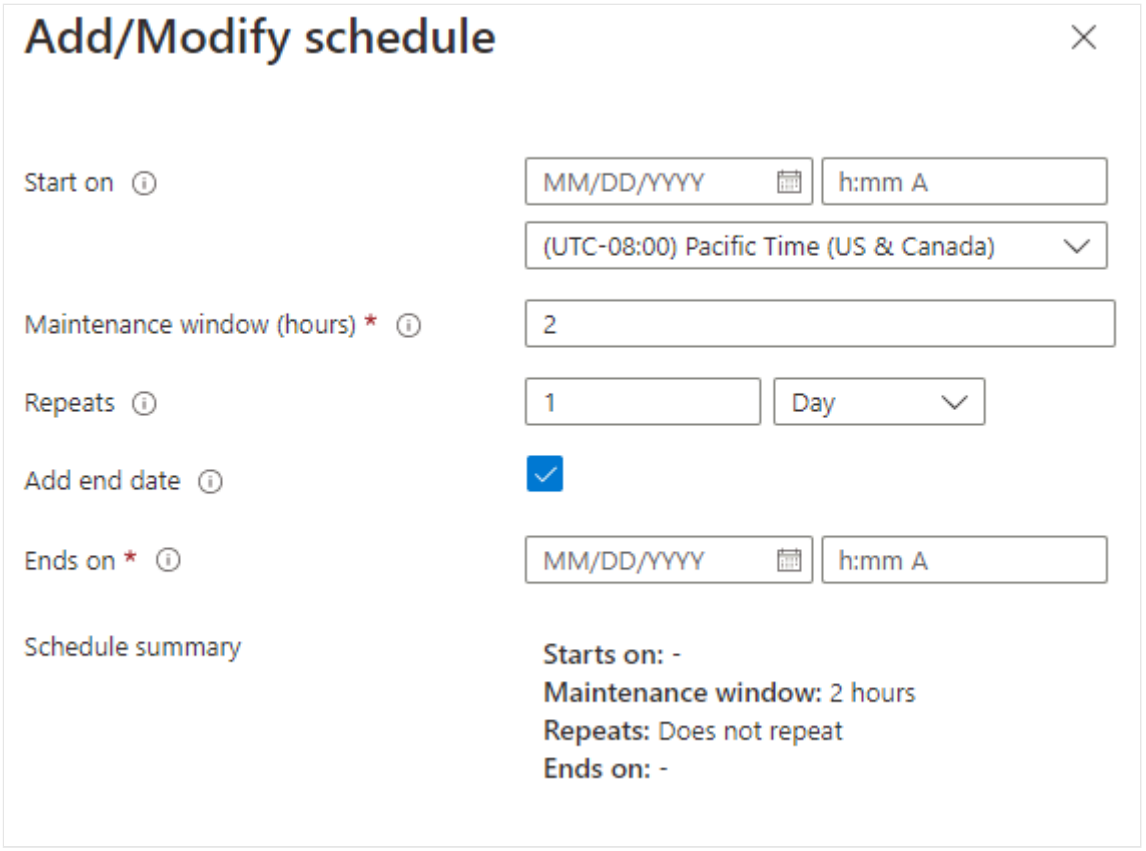
4. On the **Basics** tab, choose a subscription and resource group, provide a name for the configuration, choose a region, and select one of the scopes that you want to apply updates for. Then select **Add a schedule** to add or modify the schedule for your configuration.

> **Important**
>
> Certain virtual machine types and schedules require a specific kind of scope. To find the right scope for your virtual machine, see Scopes.



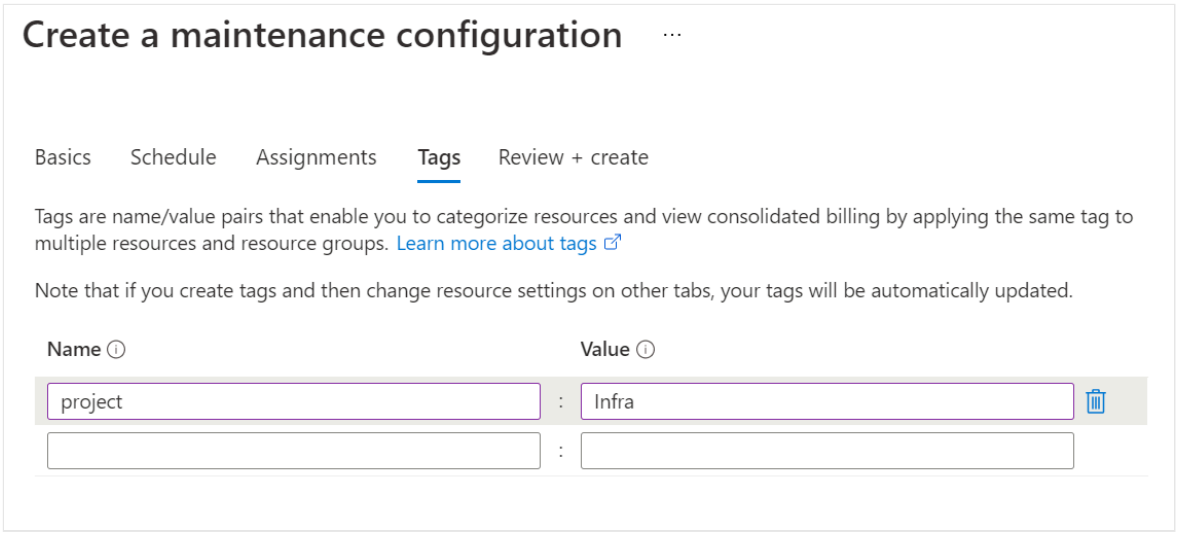
5. On the **Add/Modify schedule** pane, declare a scheduled window when Azure will apply the updates on your resources. Set a start date, maintenance window, and recurrence if your resource requires it. After you create a scheduled window, you no longer have to apply the updates manually. When you finish, select **Next**.

> **Important**
>
> The duration of the maintenance window must be 2 hours or longer.



6. On the **Machines** tab, assign resources now or skip this step and assign resources later (after you deploy the maintenance configuration). Then select **Next**.

7. On the **Tags** tab, add tags and values. Then select **Next**.

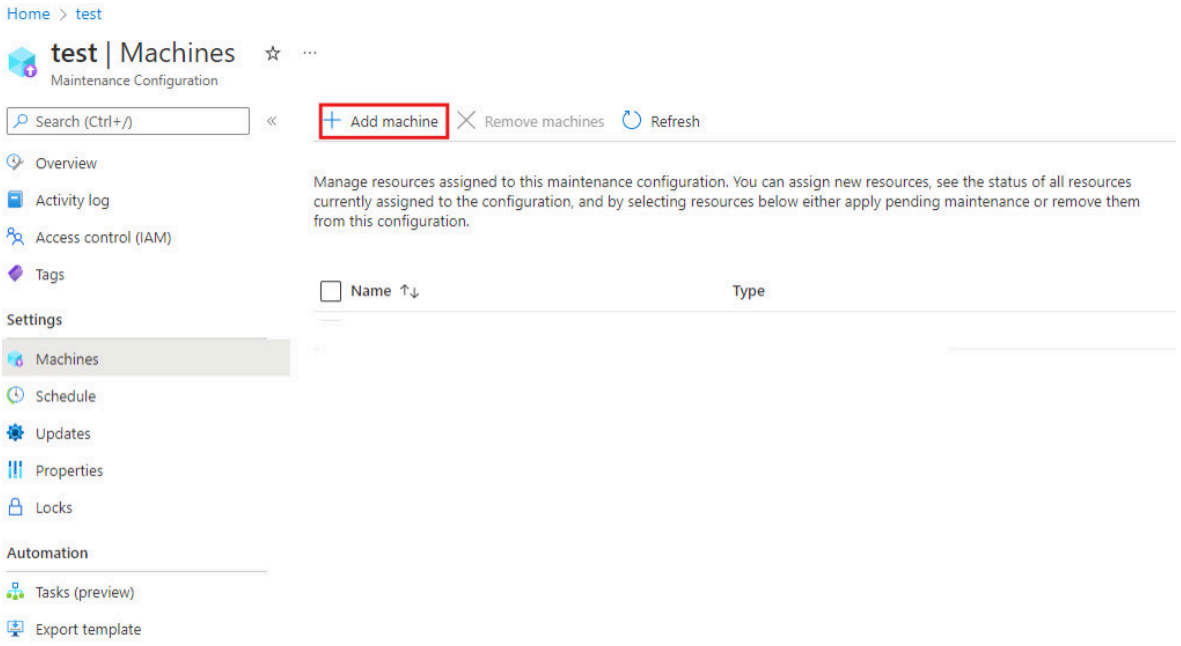


8. On the **Review + create** tab, review the summary. Then select **Create**.

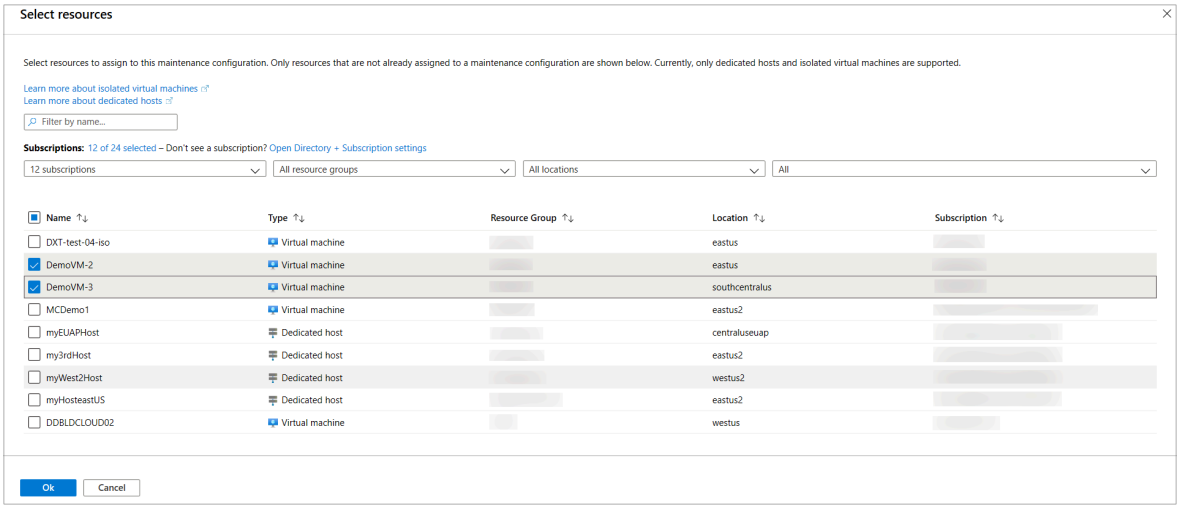
9. After the deployment is complete, select **Go to resource**.

Assign the configuration

1. On the details page of the maintenance configuration, select **Machines**, and then select **Add machine**.

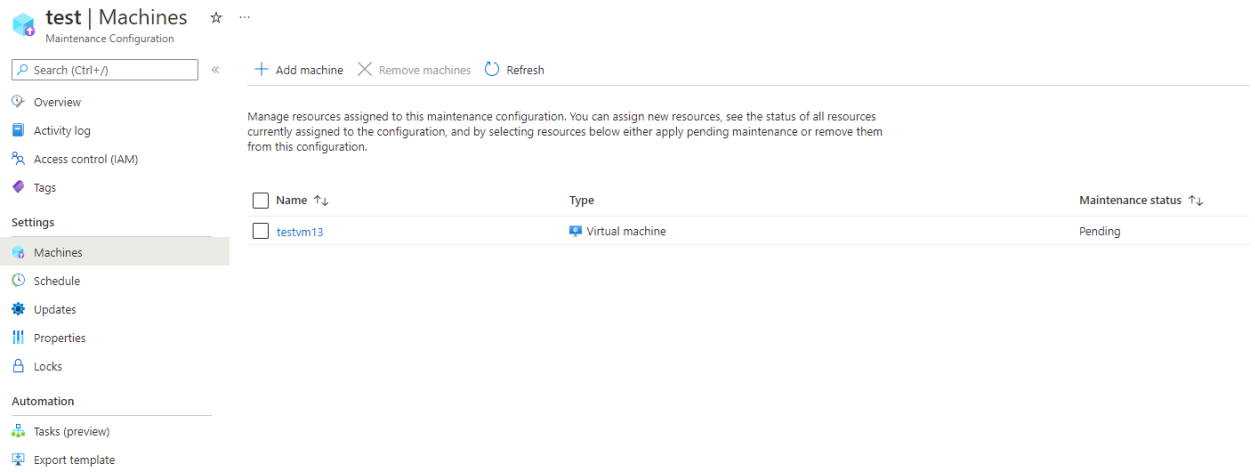


2. On the **Select resources** pane, select the resources that you want the maintenance configuration assigned to. The VMs that you select need to be running. If you try to assign a configuration to a VM that's stopped, an error occurs. When you finish, select **Ok**.

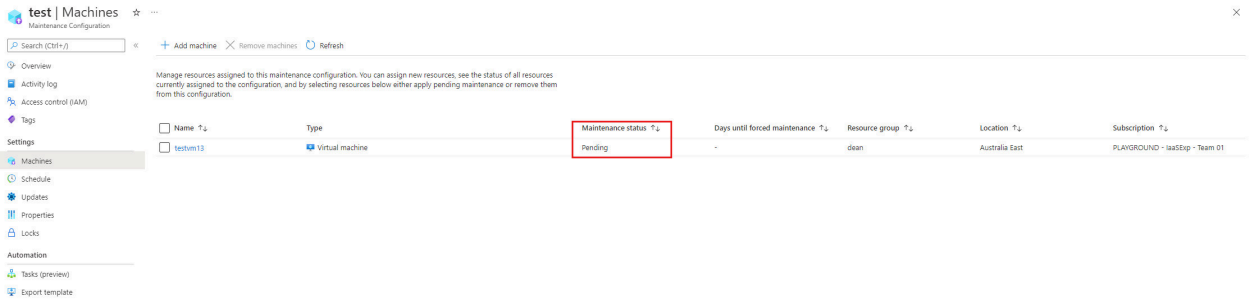


Check the configuration and status

You can verify that the configuration was applied correctly, or check which machines are assigned to a maintenance configuration, by going to **Maintenance Configurations** > **Machines**.

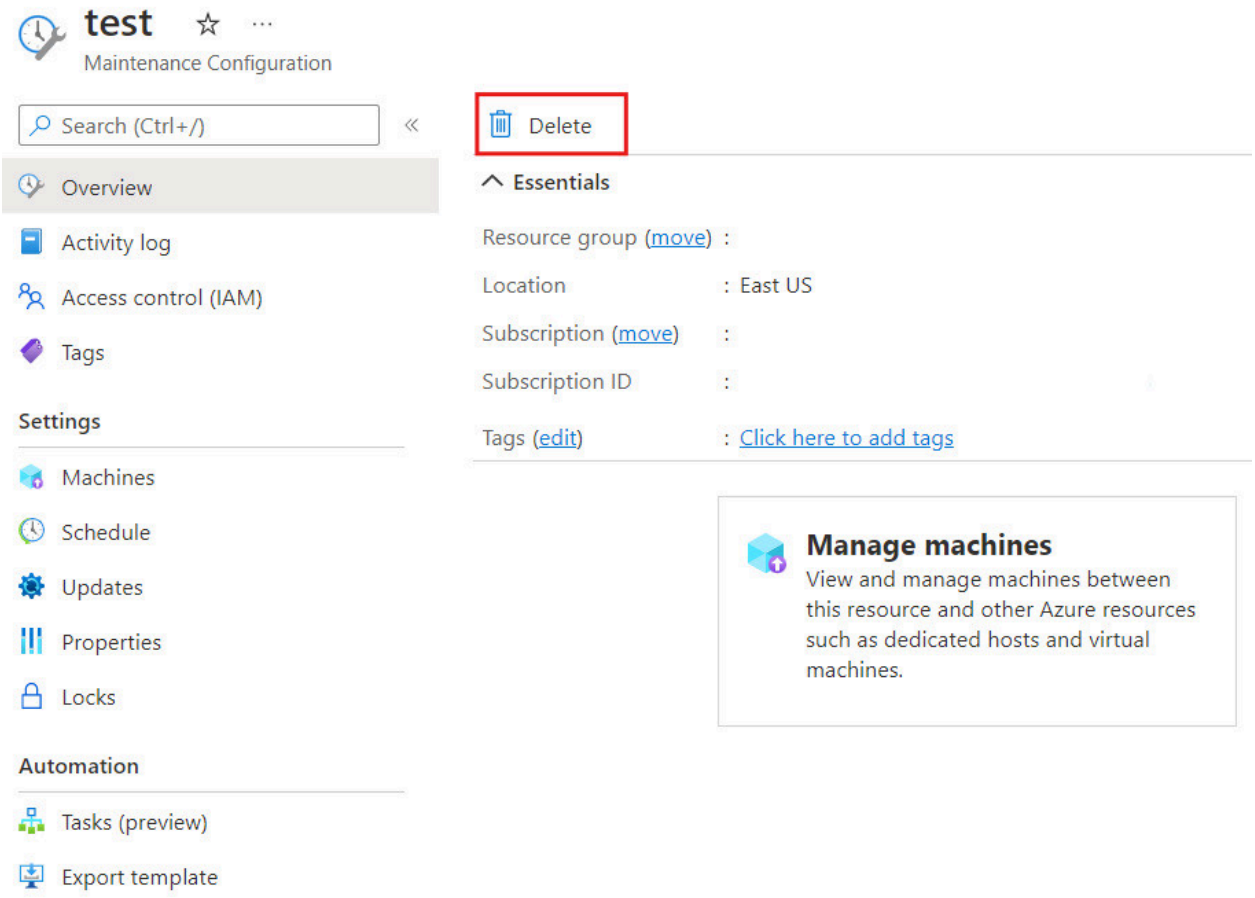


The **Maintenance status** column shows whether any updates are pending for a maintenance configuration.



Delete a maintenance configuration

To delete a maintenance configuration, open the configuration details and select **Delete**.



Next steps

To learn more, see Maintenance for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshoot problems with Maintenance Configurations

Article • 04/02/2025

This article outlines common problems and errors that might arise during the deployment or use of Maintenance Configurations for scheduled patching on virtual machines (VMs), along with strategies to address them.

A VM shuts down and is unresponsive when you use a static scope in guest maintenance

Problem

A maintenance configuration doesn't install a scheduled patch on the VMs and gives a `ShutdownOrUnresponsive` error.

Resolution

In a static scope, it's essential to avoid relying on outdated VM configurations. Therefore, it's imperative to ensure that the VM is up and running while the patch is being installed. **If the VM instance is recreated with the same name, it's crucial to prioritize reassigning the configuration after the VM instance is recreated.**

The system will automatically perform this action within 12 hours of recreation if not performed by the user. If the maintenance configuration of the attached recreated VM is triggered within 12 hours of recreating the VM with the same name, a `ShutdownOrUnresponsive` error is displayed. Therefore, it's recommended to either perform the previously mentioned action or wait for 12 hours before applying the patch to the machine via maintenance configuration.

Scheduled patching times out or fails

Problem

Scheduled patching fails with a `TimeOut` or `Failed` error after you move a VM by re-creating it with the same name in a different region. The portal might show the same VM twice because the previously created VM is removed from the back end.

Resolution

If the VM instance is recreated with the same name, it's crucial to prioritize reassigning the configuration after the VM instance is recreated in a different region.

The system will automatically perform this action within 12 hours of recreation if not performed by the user. If the maintenance configuration of the attached recreated VM is triggered within 12 hours of recreating the VM with the same name, a `TimeOut` or `Failed` error is displayed. Therefore, it's recommended to either perform the previously mentioned action or wait for 12 hours before applying the patch to the machine via maintenance configuration.

Unable to delete configuration assignment

Problem

Configuration assignment couldn't be removed or deleted from a particular maintenance configuration

Resolution

Use the following steps to mitigate this issue:

1. Delete the existing maintenance configuration in which you are encountering this issue.
2. Create a new maintenance configuration and assign the required set of dynamic scope and VMs as attached in the deleted maintenance configuration.

If you want to create new maintenance configuration with the same name as the deleted maintenance configuration, you will have to wait 20 minutes for the cleanup to take place in the backend. The system doesn't allow the creation of maintenance configuration with the same name if cleanup isn't performed in the backend.

Scheduled patching stops working after the resource is moved

Problem

If you move a resource to a different resource group or subscription, scheduled patching for the resource stops working.

Resolution

The system currently doesn't support moving resources across resource groups or subscriptions. As a workaround, use the following steps for the resource that you want to move. **As a prerequisite, first remove the assignment before following the steps.**

If you're using a `static` scope:

1. Move the resource to a different resource group or subscription.
2. Re-create the resource assignment.

If you're using a `dynamic` scope:

1. Initiate or wait for the next scheduled run. This action prompts the system to completely remove the assignment, so you can proceed with the next steps.
2. Move the resource to a different resource group or subscription.
3. Re-create the resource assignment.

If any of the steps are missed, move the resource to the previous resource group or subscription ID and reattempt the steps.

> ⓘ **Note**
>
> If the resource group is deleted, recreate it with the same name. If the subscription ID is deleted, reach out to the support team for mitigation.

Maintenance Configuration didn't trigger on the configured date time

Problem

After creating a Maintenance Configuration with a repeat value of either week or month, you expect the schedule to start at the specified date and time and then recur based on the chosen interval. However, the schedule didn't trigger at the start date and time.

Resolution

The Maintenance Configuration first run occurs on the first recurrence value following the specified start date, not necessarily on the start date itself. For example, if the maintenance configuration starts on January 17 (Wednesday) and is set to recur every

Monday, the first run of the schedule will be on the first Monday after January 17, which is January 22.

You can view first 4 instances of the scheduled run when creating a new maintenance configuration from Azure Portal.

Creation of a dynamic scope fails

Problem

You can't create a dynamic scope because of role-based access control (RBAC).

Resolution

To create a dynamic scope, you must have the permission at the subscription level or at the resource group level. Specifically, following are the requirements you need to take care of.

1. The subscription under which dynamic scope is getting created should be registered to Maintenance RP.
2. It's recommended to have the 'Scheduled Patching Contributor' role to be assigned to the following scopes:
   a. The subscription/resource group at which the dynamic scope is being created.
   b. The maintenance configuration scope.

For more information, see the list of permissions list for various resources here.

An update is stuck and not progressing

Problem

Applies to: ✔ Dedicated Hosts ✔ VMs

If you redeploy a resource to a different cluster, and you create a pending update request by using the old cluster value, the request becomes stuck indefinitely.

Resolution

If the status of an operation to apply an update is closed or not found, retry after 120 hours. If the problem persists, contact the support team for assistance.

A dedicated host is updated after a maintenance configuration is attached

Problem

A maintenance configuration doesn't block the update of a dedicated host, and the host is updated even after you attach a maintenance configuration.

Resolution

If you re-create a dedicated host with the same name, Maintenance Configurations retains the old dedicated host ID, which prevents it from blocking updates. You can resolve this problem by removing the maintenance configuration and reassigning it. **The system will automatically perform this action within 12 hours of recreation if not performed by the user.** It's recommended to either perform the previously mentioned action or wait for 12 hours before this action is performed automatically.

A schedule isn't triggered

Problem

If a resource has two maintenance configurations with the same trigger time and patch installation configuration, and both are assigned to the same VM or resource, only one maintenance configuration is triggered.

Resolution

Modify the start time of one of the maintenance configurations to mitigate the problem. This is a workaround to a current system limitation in which Maintenance Configurations can't identify which maintenance configuration to trigger.

You can't create a dynamic scope for a resource group

Problem

Dynamic scope validation fails because of a null value in the location.

Resolution

This problem with dynamic scope validation causes regression in the validation process. We recommend that you provide the required set of locations for a dynamic scope at the resource group level.

A dynamic scope isn't executed and no resources are patched

Problem

Dynamic scope flattening fails because of throttling, and the service can't determine which VMs are associated with the VM.

Resolution

Make sure that the number of subscriptions per dynamic scope isn't more than 200. Learn more about the service limits of dynamic scoping.

Configuration assignment of a dedicated host isn't cleaned up after the host's removal

Problem

After you delete the dedicated hosts, configuration assignments that are attached to dedicated hosts still exist.

Resolution

Before you delete a dedicated host, be sure to delete the maintenance configuration that's associated with it. If the dedicated host is deleted but still appears in the portal, contact the support team for assistance. Cleanup processes are currently in place for dedicated hosts, to help prevent any impact on customers.

Can't provide multiple tag values for dynamic scopes

Problem

If you use the Azure portal, you can't provide multiple tag values for dynamic scopes.

Resolution

This feature currently isn't available in the portal. As a workaround, you can use the Azure CLI or Azure PowerShell to create a dynamic scope. The system accepts multiple values for tags when you use the Azure CLI or Azure PowerShell option.

A maintenance configuration is triggered again with an older trigger time

Problem

There's a known issue in Maintenance Configurations related to the caching of old maintenance policies. If an old policy is cached, and a new instance moves the new policy processing, the old machine might trigger the schedule with the outdated start time.

Resolution

We recommend that you update the maintenance configuration at least 1 hour before the scheduled time. If the problem persists, contact the support team for assistance.

A maintenance configuration times out while waiting for an ongoing update to finish on a resource

Problem

In rare cases, if the host update window happens to coincide with the VM guest patching window, and if the guest patching window doesn't have sufficient time to run after the host update, the system shows this error message: "Schedule timeout, waiting for an ongoing update to complete the resource." The reason is that the platform allows only one update at a time.

Resolution

Change the maintenance configuration schedule for the guest update for a time after the ongoing update is finished.

New maintenance configuration retains attributes of deleted configuration

Problem

A newly created maintenance configuration with the same name as a previously deleted configuration is inheriting properties from the deleted configuration.

Resolution

Ensure a minimum wait time of 20 minutes between deleting a maintenance configuration and creating a new one with the same name. If the problem persists, contact the support team for assistance.

Maintenance assignment fails with `InternalServerError` error

Problem

Maintenance assignment fails with `InternalServerError` when using either bicep template or MRP API. The assignment can be done successfully using either portal or CLI or PowerShell.

Resolution

It is recommended to use the location/region in a normalized form for the bicep template and MRP API. The normalized form involves removing whitespace and converting the text to lowercase. For example, `EAST US 2` will result in an internal server error, whereas `eastus2` will be processed successfully.

Workload isolation in maintenance configuration

Problem

Unexpected outage due to assumption of maintenance configuration respecting workload isolation

Resolution

Maintenance configurations will initiate updates on all resources associated with it simultaneously. Customers have the flexibility to decide on the isolation and

maintenance windows for their workloads. They can create single or multiple maintenance configurations based on the isolation that is optimal for their workloads.

For example, if a customer wants to isolate updates for their development environment resources from the production environment resources, they can create separate maintenance configurations for each environment. This way, workloads in maintenance configuration with development resources are updated first, while the remaining workloads are updated in another configuration that does not overlap. This approach allows customers to control workload isolation and manage updates independently.

Maintenance Configurations doesn't support an API

The feature currently doesn't support the following APIs:

- Get Apply Update at Subscription Level
- Get Apply Update at Resource Group Level
- Get Pending Update at Subscription Level
- Get Pending Update at Resource Group Level

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Maintenance control for Azure Virtual Machine Scale Sets

Article • 10/22/2024

Applies to: ✔ Uniform scale sets

Manage automatic OS image upgrades for your Virtual Machine Scale Sets using maintenance control.

Maintenance control lets you decide when to apply updates to OS disks in your Virtual Machine Scale Sets through an easier and more predictable experience.

Maintenance configurations work across subscriptions and resource groups.

The entire workflow comes down to these steps:

- Create a maintenance configuration.
- Associate a Virtual Machine Scale Set to a maintenance configuration.
- Enable automatic OS upgrades.

Limitations

- VMs must be in a scale set.
- User must have **Resource Contributor** access.
- Maintenance duration must be 5 hours or longer in the maintenance configuration.
- Maintenance recurrence must be set to 'Day' or 'Week' in the maintenance configuration.

Management options

You can create and manage maintenance configurations using any of the following options:

- Azure PowerShell
- Azure CLI
- Azure portal

Next steps

Virtual Machine Scale Set maintenance control by using PowerShell

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Maintenance control for OS image upgrades on Azure Virtual Machine Scale Sets using PowerShell

Article • 08/22/2024

Applies to: ✔ Uniform scale sets

Maintenance control lets you decide when to apply automatic guest OS image upgrades to your Virtual Machine Scale Sets. This topic covers the Azure PowerShell options for Maintenance control. For more information on using Maintenance control, see Maintenance control for Azure Virtual Machine Scale Sets.

Enable the PowerShell module

Make sure `PowerShellGet` is up to date.

Azure PowerShell

```
Install-Module -Name PowerShellGet -Repository PSGallery -Force
```

Install the `Az.Maintenance` PowerShell module.

Azure PowerShell

```
Install-Module -Name Az.Maintenance
```

If you're installing locally, make sure you open your PowerShell prompt as an administrator.

You may also be asked to confirm that you want to install from an *untrusted repository*. Type `Y` or select **Yes to All** to install the module.

Connect to an Azure account

Connect to your desired Azure account using Connect-AzAccount and Set-AzAccount.

Azure PowerShell

```
Connect-AzAccount
Set-AzContext 00a000aa-0a00-0a0a-00aa-a00a000aaa00
```

```
$RGName="myMaintenanceRG"
$MaintenanceConfig="myMaintenanceConfig"
$location="eastus2"
$vmss="myMaintenanceVMSS"
```

Create a maintenance configuration

Create a resource group as a container for your configuration. In this example, a resource group named *myMaintenanceRG* is created in *eastus2*. If you already have a resource group that you want to use, you can skip this part. Just replace the resource group name with your own in the rest of the examples.

Azure PowerShell

```
New-AzResourceGroup `
   -Location $location `
   -Name $RGName
```

Use New-AzMaintenanceConfiguration to create a maintenance configuration. This example creates a maintenance configuration named *myConfig* scoped to the OS image.

Azure PowerShell

```
$config = New-AzMaintenanceConfiguration `
   -ResourceGroup $RGName `
   -Name $MaintenanceConfig `
   -MaintenanceScope OSImage `
   -Location $location `
   -StartDateTime "2020-10-01 00:00" `
   -TimeZone "Pacific Standard Time" `
   -Duration "05:00" `
   -RecurEvery "Day"
```

ⓘ Important

Maintenance **duration** must be *5 hours* or longer. Maintenance **recurrence** must be set to *Day*.

Using `-MaintenanceScope OSImage` ensures that the maintenance configuration is used for controlling updates to the guest OS.

If you try to create a configuration with the same name, but in a different location, you'll get an error. Configuration names must be unique to your resource group.

You can query for available maintenance configurations using Get-AzMaintenanceConfiguration.

Azure PowerShell

```
Get-AzMaintenanceConfiguration | Format-Table -Property Name,Id
```

Associate your Virtual Machine Scale Set to the maintenance configuration

A Virtual Machine Scale Set can be associated to any Maintenance configuration regardless of the region and subscription of the Maintenance configuration. By opting in to the Maintenance configuration, new OS image updates for the scale set will be automatically scheduled on the next available maintenance window.

Use New-AzConfigurationAssignment to associate your Virtual Machine Scale Set the maintenance configuration.

Azure PowerShell

```
New-AzConfigurationAssignment `
   -ResourceGroupName $RGName `
   -Location $location `
   -ResourceName $vmss `
   -ResourceType VirtualMachineScaleSets `
   -ProviderName Microsoft.Compute `
   -ConfigurationAssignmentName $config.Name`
   -MaintenanceConfigurationId $config.Id
```

Enable automatic OS upgrade

You can enable automatic OS upgrades for each Virtual Machine Scale Set that is going to use maintenance control. For more information about enabling automatic OS upgrades on your Virtual Machine Scale Set, see Azure Virtual Machine Scale Set automatic OS image upgrades.

Next steps

Learn about Maintenance and updates for virtual machines running in Azure

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Maintenance control for OS image upgrades on Azure Virtual Machine Scale Sets using Azure CLI

Article • 08/22/2024

Maintenance control lets you decide when to apply automatic guest OS image upgrades to your Virtual Machine Scale Sets. This topic covers the Azure CLI options for Maintenance control. For more information on using Maintenance control, see Maintenance control for Azure Virtual Machine Scale Sets.

Create a maintenance configuration

Use `az maintenance configuration create` to create a maintenance configuration. This example creates a maintenance configuration named *myConfig* scoped to the osimage.

Azure CLI

```
az group create \
   --location eastus \
   --name myMaintenanceRG
az maintenance configuration create \
   -g myMaintenanceRG \
   --resource-name myConfig \
   --maintenance-scope osimage\
   --location eastus
```

Copy the configuration ID from the output to use later.

Using `--maintenance-scope osimage` ensures that the maintenance configuration is used for controlling updates to the guest OS.

If you try to create a configuration with the same name, but in a different location, you will get an error. Configuration names must be unique to your resource group.

You can query for available maintenance configurations using `az maintenance configuration list`.

Azure CLI

```
az maintenance configuration list --query "[].{Name:name, ID:id}" -o table
```

Create a maintenance configuration with a scheduled window

You can also declare a scheduled window when Azure will apply the updates on your resources. This example creates a maintenance configuration named *myConfig* with a scheduled window of 5 hours on the fourth Monday of every month. Once you create a scheduled window, you no longer have to apply the updates manually.

> **ⓘ Important**
>
> Maintenance **duration** must be *5 hours* or longer. Maintenance **recurrence** must be set to *Day*.

Azure CLI

```
az maintenance configuration create \
   -g myMaintenanceRG \
   --resource-name myConfig \
   --maintenance-scope osimage \
   --location eastus \
   --maintenance-window-duration "05:00" \
   --maintenance-window-recur-every "Day" \
   --maintenance-window-start-date-time "2020-12-30 08:00" \
   --maintenance-window-time-zone "Pacific Standard Time"
```

Assign the configuration

Use `az maintenance assignment create` to assign the configuration to your Virtual Machine Scale Set.

Enable automatic OS upgrade

You can enable automatic OS upgrades for each Virtual Machine Scale Set that is going to use maintenance control. For more information about enabling automatic OS upgrades on your Virtual Machine Scale Set, see Azure Virtual Machine Scale Set automatic OS image upgrades.

Next steps

Learn about Maintenance and updates for virtual machines running in Azure

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

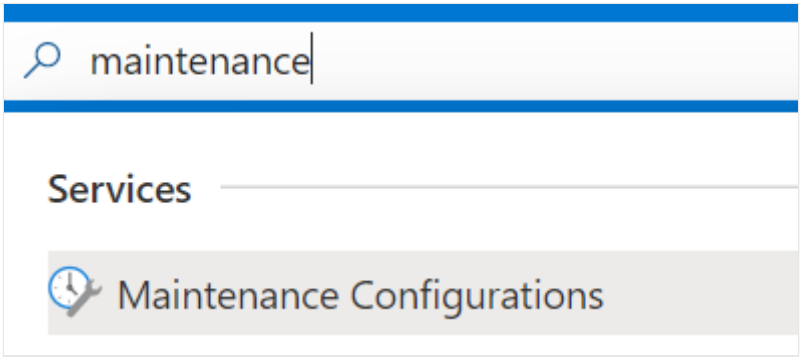
Maintenance control for OS image upgrades on Azure Virtual Machine Scale Sets using Azure portal

Article • 08/22/2024

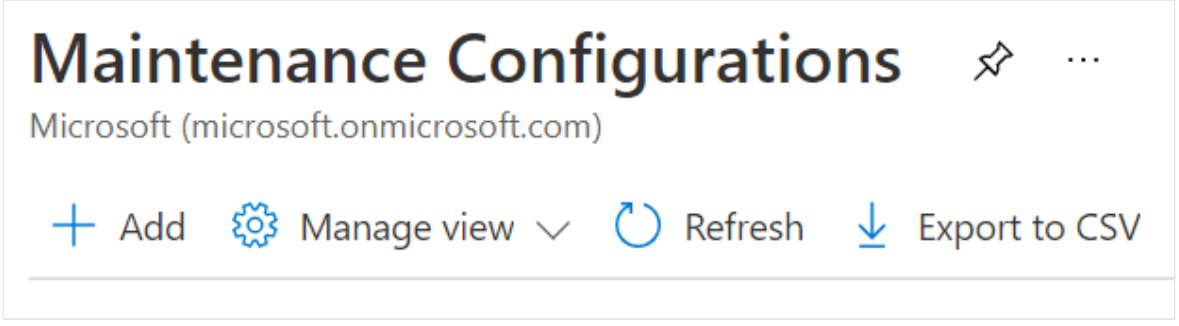
Maintenance control lets you decide when to apply automatic guest OS image upgrades to your Virtual Machine Scale Sets. This topic covers the Azure portal options for Maintenance control. For more information on using Maintenance control, see Maintenance control for Azure Virtual Machine Scale Sets.

Create a maintenance configuration

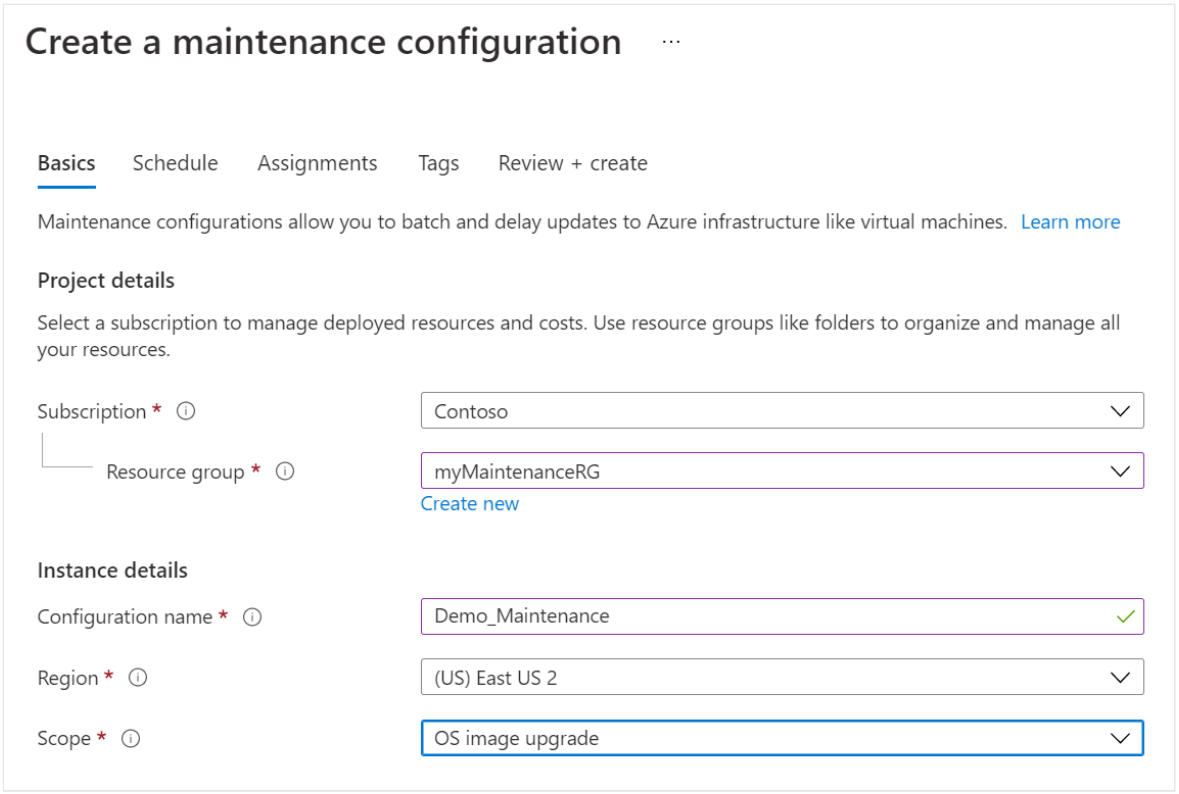
1. Sign in to the Azure portal.
2. Search for **Maintenance Configurations**.



3. Select **Add**.

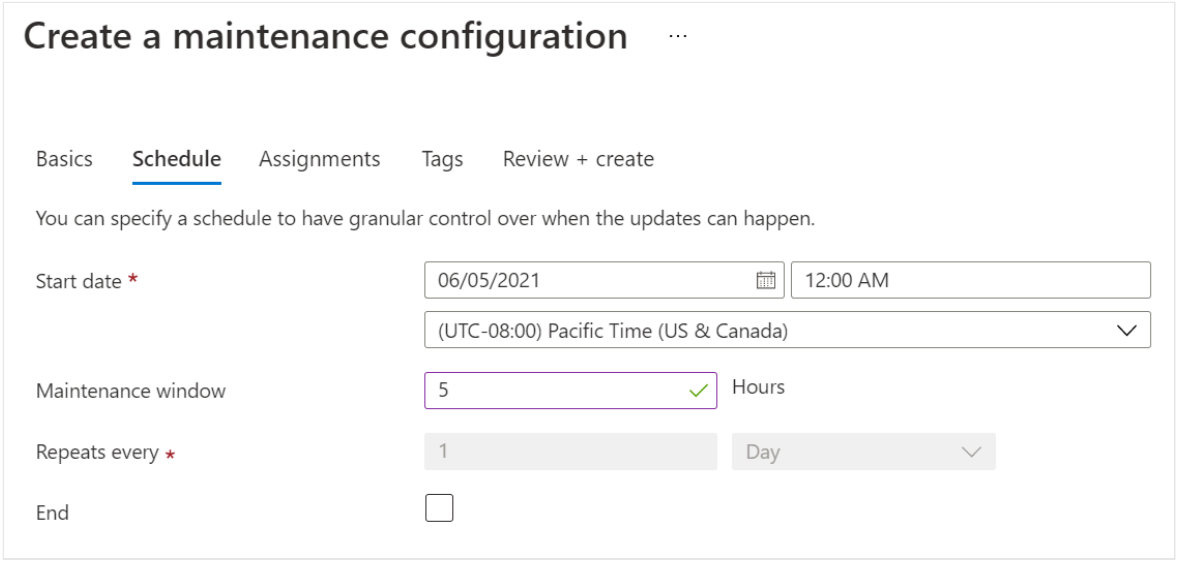


4. In the Basics tab, choose a subscription and resource group, provide a name for the configuration, choose a region, and select *OS image upgrade* for the scope. Select **Next**.



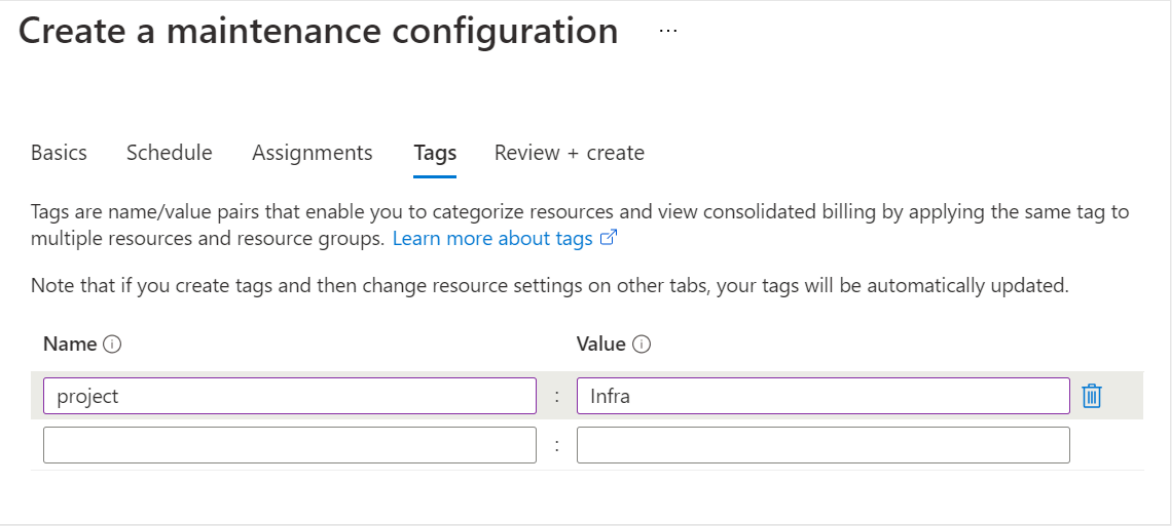
5. In the Schedule tab, declare a scheduled window when Azure will apply the updates on your resources. Set a start date, maintenance window, and recurrence. Once you create a scheduled window, you no longer have to apply the updates manually. Select **Next**.

> **Important**
>
> Maintenance window **duration** must be *5 hours* or longer. Maintenance **recurrence** must be set to repeat at least once a day.



6. In the Assignment tab, assign resources now or skip this step and assign resources after the maintenance configuration deployment. Select **Next**.

7. Add tags and values. Select **Next**.

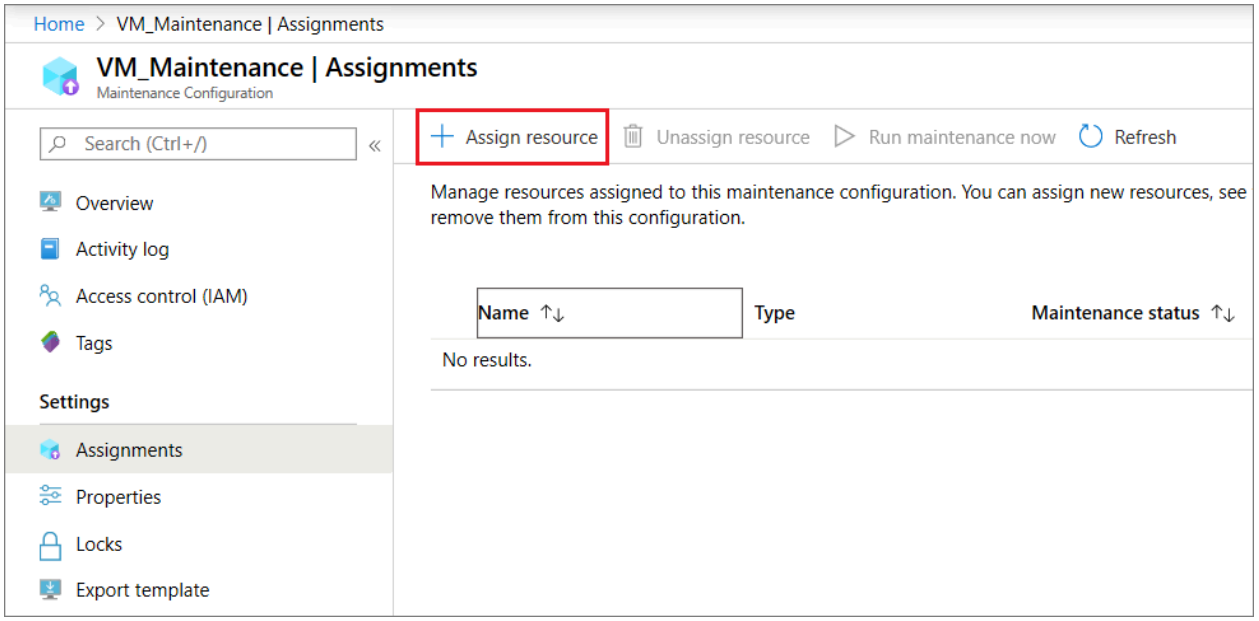


8. Review the summary. Select **Create**.

9. After the deployment is complete, select **Go to resource**.

Assign the configuration

On the details page of the maintenance configuration, select **Assignments** and then select **Assign resource**.



Select the Virtual Machine Scale Set resources that you want the maintenance configuration assigned to and select **Ok**.

Next steps

Learn about Maintenance and updates for virtual machines running in Azure

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Maintenance control for OS image upgrades on Azure Virtual Machine Scale Sets using an ARM template

Article • 08/22/2024

Maintenance control lets you decide when to apply automatic OS image upgrades to your Virtual Machine Scale Sets. For more information on using Maintenance control, see Maintenance control for Azure Virtual Machine Scale Sets.

This article explains how you can use an Azure Resource Manager (ARM) template to create a maintenance configuration. You will learn how to:

- Create the configuration
- Assign the configuration to a virtual machine

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

Create the configuration

While creating the configuration, it is important to note that there are different scopes and each will have unique properties in their creation template. Make sure you are using the right one.

For more information about this Maintenance Configuration template, see maintenanceConfigurations.

Host and OS image

JSON

```
{
  "type": "Microsoft.Maintenance/maintenanceConfigurations",
  "apiVersion": "2021-09-01-preview",
  "name": "string",
  "location": "string",
  "tags": {
    "tagName1": "tagValue1",
    "tagName2": "tagValue2"
  },
```

```
  "properties": {
    "extensionProperties": {},
    "installPatches": {
      "linuxParameters": {
        "classificationsToInclude": [ "string" ],
        "packageNameMasksToExclude": [ "string" ],
        "packageNameMasksToInclude": [ "string" ]
      },
      "rebootSetting": "string",
      "tasks": {
        "postTasks": [
          {
            "parameters": {},
            "source": "string",
            "taskScope": "string"
          }
        ],
        "preTasks": [
          {
            "parameters": {},
            "source": "string",
            "taskScope": "string"
          }
        ]
      },
      "windowsParameters": {
        "classificationsToInclude": [ "string" ],
        "excludeKbsRequiringReboot": "bool",
        "kbNumbersToExclude": [ "string" ],
        "kbNumbersToInclude": [ "string" ]
      }
    },
    "maintenanceScope": "string",
    "maintenanceWindow": {
      "duration": "string",
      "expirationDateTime": "string",
      "recurEvery": "string",
      "startDateTime": "string",
      "timeZone": "string"
    },
    "namespace": "string",
    "visibility": "string"
  }
}
```

Assign the configuration

Assign the configuration to a virtual machine.

For more information, see configurationAssignments.

JSON

```
{
"type": "Microsoft.Maintenance/configurationAssignments",
"apiVersion": "2021-09-01-preview",
"name": "[variables('maintenanceConfigurationAssignmentName')]",
"location": "string (e.g. westeurope)",
"scope": "Resource Id of the resource that is being assigned to the
Maintenance Configuration (e.g. VMSS Id)"
"properties": {
  "maintenanceConfigurationId": "Resource Id of the Maintenance
Configuration"
  "resourceId": "Resource Id of the resource that is being assigned to the
Maintenance Configuration (e.g. VMSS Id)"
}
}
```

Next steps

Learn about maintenance and updates for virtual machines running in Azure

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Metadata Service: Scheduled Events for Linux VMs

06/06/2025

Applies to: ✔ Linux VMs ✔ Flexible scale sets ✔ Uniform scale sets

Scheduled Events is an Azure Metadata Service that gives your application time to prepare for virtual machine (VM) maintenance. It provides information about upcoming maintenance events (for example, reboot) so that your application can prepare for them and limit disruption. It's available for all Azure Virtual Machines types, including PaaS and IaaS on both Windows and Linux.

For information about Scheduled Events on Linux, see Scheduled Events for Linux VMs.

Scheduled events provide proactive notifications about upcoming events, for reactive information about events that happened in the past see VM availability information in Azure Resource Graph and Create an availability alert rule for Azure virtual machine.

> ⓘ **Note**
>
> Scheduled Events is generally available in all Azure Regions. See Version and Region Availability for latest release information.

Why use Scheduled Events?

Many applications can benefit from time to prepare for VM maintenance. The time can be used to perform application-specific tasks that improve availability, reliability, and serviceability, including:

- Checkpoint and restore.
- Connection draining.
- Primary replica failover.
- Removal from a load balancer pool.
- Event logging.
- Graceful shutdown.

With Scheduled Events, your application can discover when maintenance will occur and trigger tasks to limit its impact.

Scheduled Events provides events in the following use cases:

- Platform initiated maintenance (for example, VM reboot, live migration or memory preserving updates for host).
- Virtual machine is running on degraded host hardware that is predicted to fail soon.
- Virtual machine was running on a host that suffered a hardware failure.
- User-initiated maintenance (for example, a user restarts or redeploys a VM).
- Spot VM and Spot scale set instance evictions.

The Basics

Metadata Service exposes information about running VMs by using a REST endpoint that's accessible from within the VM. The information is available via a nonroutable IP and isn't exposed outside the VM.

Scope

Scheduled events are delivered to and can be acknowledged by:

- Standalone Virtual Machines.
- All the VMs in an Azure cloud service (classic).
- All the VMs in an availability set.
- All the VMs in a scale set placement group.

Scheduled Events for all virtual machines (VMs) in an entire Availability Set or a Placement Group for a Virtual Machine Scale Set are delivered to all other VMs in the same group or set regardless of Availability Zone usage.

As a result, check the `Resources` field in the event to identify which VMs are affected.

> **Note**
>
> GPU accelerated Virtual Machines in a scale set using one fault domain (FD = 1) only receive scheduled events for the impacted resource. Events won't be broadcasted to all VMs in the same placement group.

Endpoint discovery

For VNET enabled VMs, Metadata Service is available from a static nonroutable IP, `169.254.169.254`. The full endpoint for the latest version of Scheduled Events is:

```
http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01
```

If the VM isn't created within a Virtual Network, the default cases for cloud services and classic VMs, other logic is required to discover the IP address to use. To learn how to discover the host endpoint, see this sample.

Version and region availability

The Scheduled Events service is versioned. Versions are mandatory; the current version is `2020-07-01`.

Expand table

Version	Release Type	Regions	Release Notes
2020-07-01	General Availability	All	• Added support for Event Duration
2019-08-01	General Availability	All	• Added support for EventSource
2019-04-01	General Availability	All	• Added support for Event Description
2019-01-01	General Availability	All	• Added support for Virtual Machine Scale Sets EventType 'Terminate'
2017-11-01	General Availability	All	• Added support for Spot VM eviction EventType 'Preempt'
2017-08-01	General Availability	All	• Removed prepended underscore from resource names for IaaS VMs • Metadata header requirement enforced for all requests
2017-03-01	Preview	All	• Initial release

> **Note**
>
> Previous preview releases of Scheduled Events supported {latest} as the api-version. This format is no longer supported and will be deprecated in the future.

Enabling and disabling Scheduled Events

Scheduled Events is enabled for your service the first time you make a request for events. You should expect a delayed response in your first call of up to two minutes and you'll start

receiving events within 5 minutes. Scheduled Events is disabled for your service if it doesn't make a request to the endpoint for 24 hours.

User-initiated maintenance

User-initiated VM maintenance via the Azure portal, API, CLI, or PowerShell results in a scheduled event. You then can test the maintenance preparation logic in your application, and your application can prepare for user-initiated maintenance.

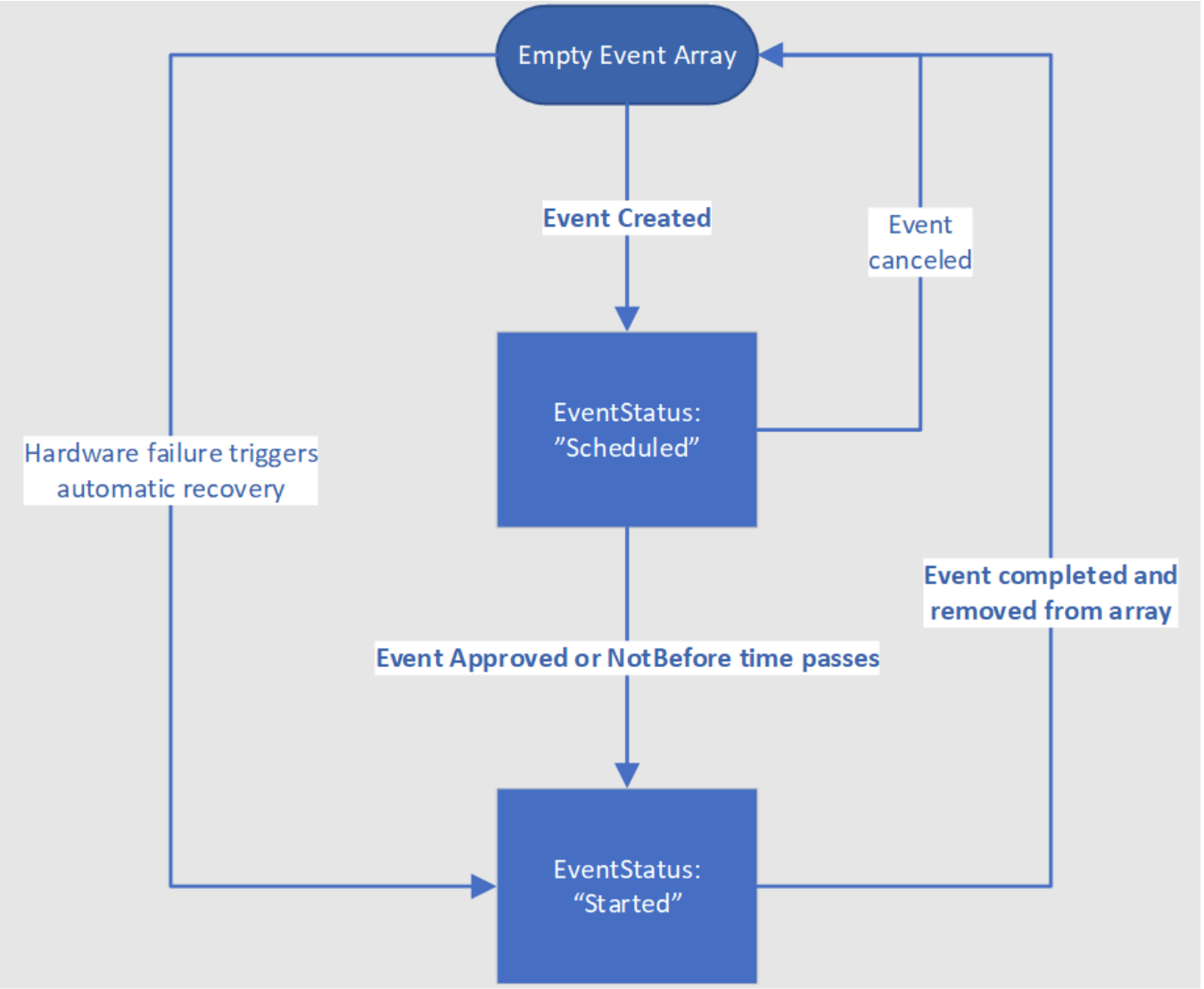
If you restart a VM, an event with the type `Reboot` is scheduled. If you redeploy a VM, an event with the type `Redeploy` is scheduled. Typically events with a user event source can be immediately approved to avoid a delay on user-initiated actions. We advise having a primary and secondary VM communicating and approving user generated scheduled events in case the primary VM becomes unresponsive. Immediately approving events prevents delays in recovering your application back to a good state.

Scheduled events for Virtual Machine Scale Sets Guest OS upgrades or reimages are supported for general purpose VM sizes that support memory preserving updates only. It doesn't work for G, M, N, and H series. Scheduled events for Virtual Machine Scale Sets Guest OS upgrades and reimages are disabled by default. To enable scheduled events for these operations on supported VM sizes, first enable them using OSImageNotificationProfile.

Use the API

High level overview

There are two major components to handling Scheduled Events, preparation and recovery. All current scheduled events impacting a VM are available to read via the IMDS Scheduled Events endpoint. When the event reaches a terminal state, it's removed from the list of events. The following diagram shows the various state transitions that a single scheduled event can experience:



For events in the EventStatus:"Scheduled" state, you need to take steps to prepare your workload. Once the preparation is complete, you should then approve the event using the scheduled event API. Otherwise, the event is automatically approved when the NotBefore time is reached. If the VM is on shared infrastructure, the system will then wait for all other tenants on the same hardware to also approve the job or timeout. Once approvals are gathered from all impacted VMs or the NotBefore time is reached, then Azure generates a new scheduled event payload with EventStatus:"Started" and triggers the start of the maintenance event. When the event reaches a terminal state, it's removed from the list of events. That serves as the signal for the customer to recover their VMs.

The psudeo code demonstrates a process for how to read and manage scheduled events in your application:

```
current_list_of_scheduled_events = get_latest_from_se_endpoint()
#prepare for new events
for each event in current_list_of_scheduled_events:
  if event not in previous_list_of_scheduled_events:
    prepare_for_event(event)
```

```
#recover from completed events
for each event in previous_list_of_scheduled_events:
  if event not in current_list_of_scheduled_events:
    receover_from_event(event)
#prepare for future jobs
previous_list_of_scheduled_events = current_list_of_scheduled_events
```

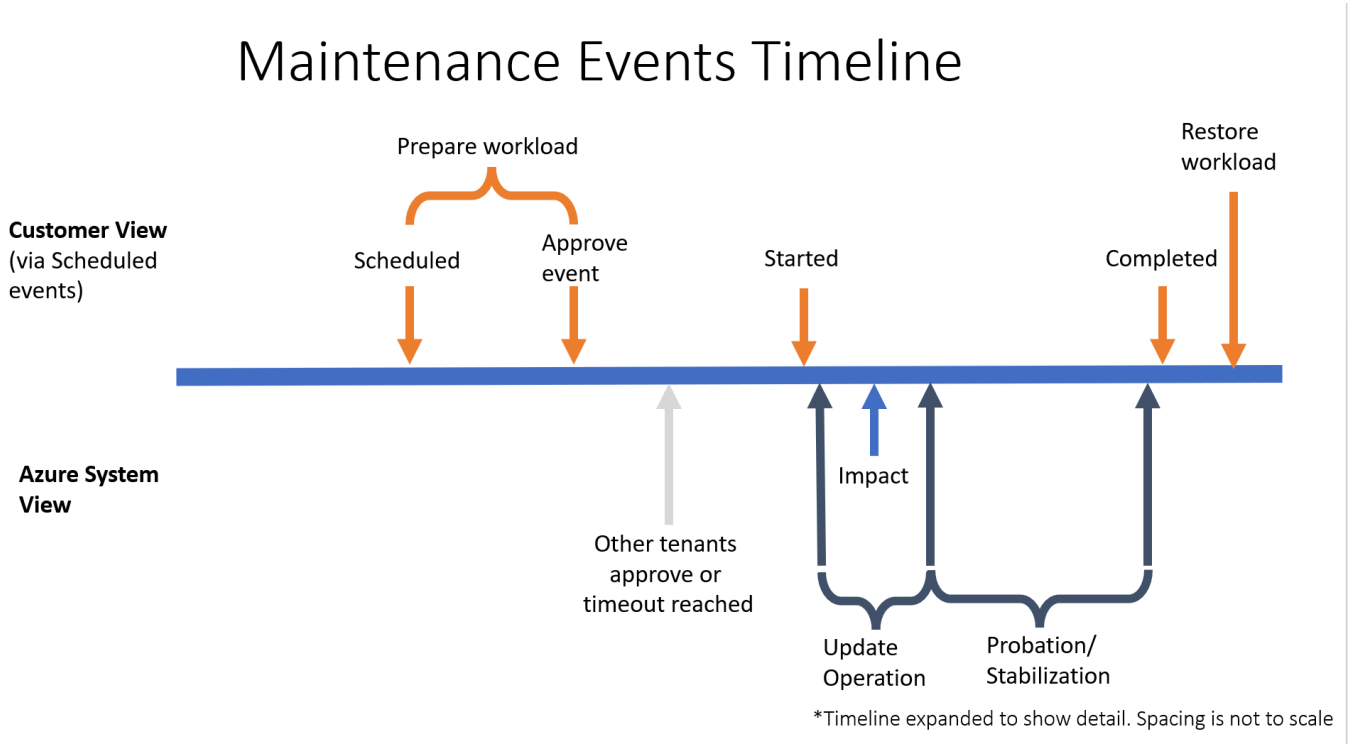
As scheduled events are often used for applications with high availability requirements, there are a few exceptional cases that should be considered:

1. Once a scheduled event is completed and removed from the array, there will be no further impacts without a new event including another EventStatus:"Scheduled" event
2. Azure monitors maintenance operations across the entire fleet and in rare circumstances determines that a maintenance operation too high risk to apply. In that case the scheduled event goes directly from "Scheduled" to being removed from the events array
3. If there is a hardware failure, Azure bypasses the "Scheduled" state and immediately move to the EventStatus:"Started" state.
4. While the event is still in EventStatus:"Started" state, there may be another impact of a shorter duration than what was advertised in the scheduled event.

As part of Azure's availability guarantee, VMs in different fault domains won't be impacted by routine maintenance operations at the same time. However, they may have operations serialized one after another. VMs in one fault domain can receive scheduled events with EventStatus:"Scheduled" shortly after another fault domain's maintenance is completed. Regardless of what architecture you chose, always keep checking for new events pending against your VMs.

While the exact timings of events vary, the following diagram provides a rough guideline for how a typical maintenance operation proceeds:

- EventStatus:"Scheduled" to Approval Timeout: 15 minutes
- Impact Duration: 7 seconds
- EventStatus:"Started" to Completed (event removed from Events array): 10 minutes



All operations impacting VM availability create a scheduled event, however not all scheduled events appear in other Azure surfaces such as Azure Activity Logs or Resource Health. Checking scheduled events regularly will ensure that you have the most up-to-date information about any upcoming impacts to your VMs.

Headers

When you query Metadata Service, you must provide the header `Metadata:true` to ensure the request wasn't unintentionally redirected. The `Metadata:true` header is required for all scheduled events requests. Failure to include the header in the request results in a "Bad Request" response from Metadata Service.

Query for events

You can query for scheduled events by making the following call:

Bash sample

```
curl -H Metadata:true http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01
```

PowerShell sample

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -Uri
"http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01" |
ConvertTo-Json -Depth 64
```

Python sample

```
import json
import requests

metadata_url ="http://169.254.169.254/metadata/scheduledevents"
header = {'Metadata' : 'true'}
query_params = {'api-version':'2020-07-01'}

def get_scheduled_events():
    resp = requests.get(metadata_url, headers = header, params = query_params)
    data = resp.json()
    return data
```

A response contains an array of scheduled events. An empty array means that currently no events are scheduled. In the case where there are scheduled events, the response contains an array of events.

```
{
    "DocumentIncarnation": {IncarnationID},
    "Events": [
        {
            "EventId": {eventID},
            "EventType": "Reboot" | "Redeploy" | "Freeze" | "Preempt" |
"Terminate",
            "ResourceType": "VirtualMachine",
            "Resources": [{resourceName}],
            "EventStatus": "Scheduled" | "Started",
            "NotBefore": {timeInUTC},
            "Description": {eventDescription},
            "EventSource" : "Platform" | "User",
            "DurationInSeconds" : {timeInSeconds},
        }
    ]
}
```

Event properties

Expand table

Property	Description
Document Incarnation	Integer that increases when the events array changes. Documents with the same incarnation contain the same event information, and the incarnation is incremented when an event changes.
EventId	Globally unique identifier for this event. Example: • 602d9444-d2cd-49c7-8624-8643e7171297
EventType	Expected impact this event will cause. Values: • `Freeze`: The Virtual Machine is scheduled to pause for a few seconds. CPU and network connectivity may be suspended, but there's no impact on memory or open files. • `Reboot`: The Virtual Machine is scheduled for reboot (non-persistent memory is lost). In rare cases a VM scheduled for EventType:"Reboot" may experience a freeze event instead of a reboot. Follow the instructions above for how to know if the event is complete and it's safe to restore your workload. • `Redeploy`: The Virtual Machine is scheduled to move to another node (ephemeral disks are lost). • `Preempt`: The Spot Virtual Machine is being deleted (ephemeral disks are lost). This event is made available on a best effort basis • `Terminate`: The virtual machine is scheduled to be deleted.
ResourceType	Type of resource this event affects. Values: • `VirtualMachine`
Resources	List of resources this event affects. Example: • ["FrontEnd_IN_0", "BackEnd_IN_0"]
EventStatus	Status of this event.

Property	Description
	Values: • `Scheduled`: This event is scheduled to start after the time specified in the `NotBefore` property. • `Started`: This event has started. No `Completed` or similar status is ever provided. The event is no longer returned when the event is finished.
NotBefore	Time after which this event can start. The event is guaranteed to not start before this time. Will be blank if the event after the event starts. Example: • Mon, 19 Sep 2016 18:29:47 GMT
Description	Description of this event. Example: • Host server is undergoing maintenance. • Host Server infrastructure is undergoing maintenance. • Virtual machine is being paused because of a memory-preserving Live Migration operation. • Virtual machine is going to be restarted as requested by authorized user. • Host server is undergoing an emergency repair.
EventSource	Initiator of the event. Example: • `Platform`: Platform initiated this event. • `User`: The user initiated this event.
DurationInSeconds	The expected duration of the interruption caused by the event. There may be secondary impacts of a shorter duration during the impact window. Example: • `9`: The interruption caused by the event lasts for 9 seconds. • `0`: The event won't interrupt the VM or impact its availability (for example, update to the network) • `-1`: The default value used if the impact duration is either unknown or not applicable.

Event scheduling

Each event is scheduled a minimum amount of time in the future based on the event type. This time is reflected in an event's `NotBefore` property.

Expand table

EventType	Minimum notice
Freeze	15 minutes
Reboot	15 minutes
Redeploy	10 minutes
Preempt	30 seconds
Terminate	User Configurable: 5 to 15 minutes

This means that you can detect a future schedule of event at least by the minimum notice time before the event occurs. Once an event is scheduled, it will move into the `Started` state after it's been approved or the `NotBefore` time passes. However, in rare cases, the operation is canceled by Azure before it starts. In that case the event is removed from the Events array, and the impact won't occur as previously scheduled.

> **Note**
>
> In some cases, Azure is able to predict host failure due to degraded hardware and attempts to mitigate disruption to your service by scheduling a migration. Affected virtual machines will receive a scheduled event with a `NotBefore` that is typically a few days in the future. The actual time varies depending on the predicted failure risk assessment. Azure tries to give 7 days of advance notice when possible. The actual notice time varies and might be shorter if there's a high chance of the hardware failing imminently. To minimize risk to your service in case the hardware fails before the system-initiated migration, we recommend that you self-redeploy your virtual machine as soon as possible.

> **Note**
>
> In the case the host node experiences a hardware failure, Azure bypasses the minimum notice period and immediately begin the recovery process for affected virtual machines. This reduces recovery time in the case that the affected VMs are unable to respond.

During the recovery process, an event is created for all impacted VMs with `EventType = Reboot` and `EventStatus = Started`.

Polling frequency

You can poll the endpoint for updates as frequently or infrequently as you like. However, the longer the time between requests, the more time you potentially lose to react to an upcoming event. Most events have 5 to 15 minutes of advance notice, although in some cases advance notice might be as little as 30 seconds. To ensure that you have as much time as possible to take mitigating actions, we recommend that you poll the service once per second.

Start an event

After you learn of an upcoming event and finish your logic for graceful shutdown, you can approve the outstanding event by making a `POST` call to Metadata Service with `EventId`. This call indicates to Azure that it can shorten the minimum notification time (when possible). The event may not start immediately upon approval. In some cases Azure requires the approval of all the VMs hosted on the node before proceeding with the event.

The following JSON sample is expected in the `POST` request body. The request should contain a list of `StartRequests`. Each `StartRequest` contains `EventId` for the event you want to expedite:

```
{
    "StartRequests" : [
        {
            "EventId": {EventId}
        }
    ]
}
```

The service always returns a 200 success code if it's passed a valid event ID, even if another VM already approved the event. A 400 error code indicates that the request header or payload was malformed.

> **Note**
>
> Events won't proceed unless they are either approved via a POST message or the NotBefore time elapses. This includes user triggered events such as VM restarts from the Azure portal.

Bash sample

```
curl -H Metadata:true -X POST -d '{"StartRequests": [{"EventId": "f020ba2e-3bc0-
4c40-a10b-86575a9eabd5"}]}' http://169.254.169.254/metadata/scheduledevents?api-
version=2020-07-01
```

PowerShell sample

```
Invoke-RestMethod -Headers @{"Metadata" = "true"} -Method POST -body
'{"StartRequests": [{"EventId": "5DD55B64-45AD-49D3-BBC9-F57D4EA97BD7"}]}' -Uri
http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01 |
ConvertTo-Json -Depth 64
```

Python sample

```
import json
import requests

def confirm_scheduled_event(event_id):
   # This payload confirms a single event with id event_id
   payload = json.dumps({"StartRequests": [{"EventId": event_id }]})
   response = requests.post("http://169.254.169.254/metadata/scheduledevents",
                            headers =  {'Metadata' : 'true'},
                            params = {'api-version':'2020-07-01'},
                            data = payload)
   return response.status_code
```

> **Note**
>
> Acknowledging an event allows the event to proceed for all `Resources` in the event, not just the VM that acknowledges the event. Therefore, you can choose to elect a leader to coordinate the acknowledgment, which might be as simple as the first machine in the `Resources` field.

Example responses

The following events are an example that from two VMs which were live migrated to another node.

The `DocumentIncarnation` is changing every time there's new information in `Events`. An approval of the event would allow the freeze to proceed for both WestNO_0 and WestNO_1.

JSON


```
{
    "DocumentIncarnation":  1,
    "Events":  [
               ]
}

{
    "DocumentIncarnation":  2,
    "Events":  [
                   {
                       "EventId":  "C7061BAC-AFDC-4513-B24B-AA5F13A16123",
                       "EventStatus":  "Scheduled",
                       "EventType":  "Freeze",
                       "ResourceType":  "VirtualMachine",
                       "Resources":  [
                                         "WestNO_0",
                                         "WestNO_1"
                                     ],
                       "NotBefore":  "Mon, 11 Apr 2022 22:26:58 GMT",
                       "Description":  "Virtual machine is being paused because of
a memory-preserving Live Migration operation.",
                       "EventSource":  "Platform",
                       "DurationInSeconds":  5
                   }
               ]
}

{
    "DocumentIncarnation":  3,
    "Events":  [
                   {
                       "EventId":  "C7061BAC-AFDC-4513-B24B-AA5F13A16123",
                       "EventStatus":  "Started",
                       "EventType":  "Freeze",
                       "ResourceType":  "VirtualMachine",
                       "Resources":  [
                                         "WestNO_0",
                                         "WestNO_1"
                                     ],
                       "NotBefore":  "",
                       "Description":  "Virtual machine is being paused because of
a memory-preserving Live Migration operation.",
                       "EventSource":  "Platform",
                       "DurationInSeconds":  5
                   }
```

```
                ]
}

{
    "DocumentIncarnation":  4,
    "Events":  [
                ]
}
```

Python Sample

The sample queries Metadata Service for scheduled events and approves each outstanding event. You can find the code in the vm-scheduled-events-mock-server repository in the Listener.py file.

Python

```python
#!/usr/bin/python
import json
import requests
from time import sleep

# The URL to access the metadata service
metadata_url ="http://169.254.169.254/metadata/scheduledevents"
# This must be sent otherwise the request will be ignored
header = {'Metadata' : 'true'}
# Current version of the API
query_params = {'api-version':'2020-07-01'}

def get_scheduled_events():
    resp = requests.get(metadata_url, headers = header, params = query_params)
    data = resp.json()
    return data

def confirm_scheduled_event(event_id):
    # This payload confirms a single event with id event_id
    # You can confirm multiple events in a single request if needed
    payload = json.dumps({"StartRequests": [{"EventId": event_id }]})
    response = requests.post(metadata_url,
                             headers= header,
                             params = query_params,
                             data = payload)
    return response.status_code

def log(event):
    # This is an optional placeholder for logging events to your system
    print(event["Description"])
    return

def advanced_sample(last_document_incarnation):
```

```
    # Poll every second to see if there are new scheduled events to process
    # Since some events may have necessarily short warning periods, it is
    # recommended to poll frequently
    found_document_incarnation = last_document_incarnation
    while (last_document_incarnation == found_document_incarnation):
        sleep(1)
        payload = get_scheduled_events()
        found_document_incarnation = payload["DocumentIncarnation"]

    # We recommend processing all events in a document together,
    # even if you won't be actioning on them right away
    for event in payload["Events"]:

        # Events that already started, logged for tracking
        if (event["EventStatus"] == "Started"):
            log(event)

        # Approve all user initiated events. These are typically created by an
        # administrator and approving them immediately can help to avoid delays
        # in admin actions
        elif (event["EventSource"] == "User"):
            confirm_scheduled_event(event["EventId"])

        # For this application, freeze events less that 9 seconds are considered
        # no impact. This will immediately approve them
        elif (event["EventType"] == "Freeze" and
            int(event["DurationInSeconds"]) >= 0  and
            int(event["DurationInSeconds"]) < 9):
            confirm_scheduled_event(event["EventId"])

        # Events that may be impactful (for example reboot or redeploy) may need
custom
        # handling for your application
        else:
            #TODO Custom handling for impactful events
            log(event)
    print("Processed events from document: " + str(found_document_incarnation))
    return found_document_incarnation

def main():
    # This will track the last set of events seen
    last_document_incarnation = "-1"

    input_text = "\
        Press 1 to poll for new events \n\
        Press 2 to exit \n "
    program_exit = False

    while program_exit == False:
        user_input = input(input_text)
        if (user_input == "1"):
            last_document_incarnation = advanced_sample(last_document_incarnation)
        elif (user_input == "2"):
            program_exit = True
```

```
if __name__ == '__main__':
    main()
```

Testing Scheduled Events

Two common ways to test your applications response to scheduled events are manually triggering user imitated events or using a mock server.

You can manually trigger redeploy and reboot events through the Azure Portal or Azure CLI by selecting the 'reboot' or 'redeploy' VM option from the VM blade. This will create an event and send it to your workload.

Operations that impact multiple VMs, such as host updates, cannot be triggered on demand so you can use a mock server instead. The vm-scheduled-events-mock-server provides a framework for testing your application's response to different scenarios by replaying real events flows back for development and testing. By default the server supports 9 different scenarios, all captured from VMs running in Azure and representing the most common cases. The scenarios can be expanded to include more options depending on your applications particular characteristics.

Next steps

- Review the Scheduled Events code samples in the Azure Instance Metadata Scheduled Events GitHub repository.
- Review the Node.js Scheduled Events code samples in Azure Samples GitHub repository.
- Read more about the APIs that are available in the Instance Metadata Service.
- Learn about planned maintenance for Linux virtual machines in Azure.
- Learn how to log scheduled events by using Azure Event Hubs in the Azure Samples GitHub repository.
- Use the vm-scheduled-events-mock-server to test your application's response to common scheduled events scenarios.

Azure Metadata Service: Scheduled Events for Windows VMs

06/06/2025

Applies to: ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Scheduled Events is an Azure Metadata Service that gives your application time to prepare for virtual machine (VM) maintenance. It provides information about upcoming maintenance events (for example, reboot) so that your application can prepare for them and limit disruption. It's available for all Azure Virtual Machines types, including PaaS and IaaS on both Windows and Linux.

For information about Scheduled Events on Linux, see Scheduled Events for Linux VMs.

Scheduled events provide proactive notifications about upcoming events, for reactive information about events that happened in the past see VM availability information in Azure Resource Graph and Create an availability alert rule for Azure virtual machine.

> ⓘ **Note**
>
> Scheduled Events is generally available in all Azure Regions. See Version and Region Availability for latest release information.

Why use Scheduled Events?

Many applications can benefit from time to prepare for VM maintenance. The time can be used to perform application-specific tasks that improve availability, reliability, and serviceability, including:

- Checkpoint and restore.
- Connection draining.
- Primary replica failover.
- Removal from a load balancer pool.
- Event logging.
- Graceful shutdown.

With Scheduled Events, your application can discover when maintenance will occur and trigger tasks to limit its impact.

Scheduled Events provides events in the following use cases:

- Platform initiated maintenance (for example, VM reboot, live migration or memory preserving updates for host).
- Virtual machine is running on degraded host hardware that is predicted to fail soon.
- Virtual machine was running on a host that suffered a hardware failure.
- User-initiated maintenance (for example, a user restarts or redeploys a VM).
- Spot VM and Spot scale set instance evictions.

The Basics

Metadata Service exposes information about running VMs by using a REST endpoint that's accessible from within the VM. The information is available via a nonroutable IP and isn't exposed outside the VM.

Scope

Scheduled events are delivered to and can be acknowledged by:

- Standalone Virtual Machines.
- All the VMs in an Azure cloud service (classic).
- All the VMs in an availability set.
- All the VMs in a scale set placement group.

Scheduled Events for all virtual machines (VMs) in an entire Availability Set or a Placement Group for a Virtual Machine Scale Set are delivered to all other VMs in the same group or set regardless of Availability Zone usage.

As a result, check the `Resources` field in the event to identify which VMs are affected.

> **Note**
>
> GPU accelerated Virtual Machines in a scale set using one fault domain (FD = 1) only receive scheduled events for the impacted resource. Events won't be broadcasted to all VMs in the same placement group.

Endpoint discovery

For VNET enabled VMs, Metadata Service is available from a static nonroutable IP, `169.254.169.254`. The full endpoint for the latest version of Scheduled Events is:

```
http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01
```

If the VM isn't created within a Virtual Network, the default cases for cloud services and classic VMs, other logic is required to discover the IP address to use. To learn how to discover the host endpoint, see this sample.

Version and region availability

The Scheduled Events service is versioned. Versions are mandatory; the current version is `2020-07-01`.

Expand table

Version	Release Type	Regions	Release Notes
2020-07-01	General Availability	All	• Added support for Event Duration
2019-08-01	General Availability	All	• Added support for EventSource
2019-04-01	General Availability	All	• Added support for Event Description
2019-01-01	General Availability	All	• Added support for Virtual Machine Scale Sets EventType 'Terminate'
2017-11-01	General Availability	All	• Added support for Spot VM eviction EventType 'Preempt'
2017-08-01	General Availability	All	• Removed prepended underscore from resource names for IaaS VMs • Metadata header requirement enforced for all requests
2017-03-01	Preview	All	• Initial release

> **Note**
>
> Previous preview releases of Scheduled Events supported {latest} as the api-version. This format is no longer supported and will be deprecated in the future.

Enabling and disabling Scheduled Events

Scheduled Events is enabled for your service the first time you make a request for events. You should expect a delayed response in your first call of up to two minutes and you'll start

receiving events within 5 minutes. Scheduled Events is disabled for your service if it doesn't make a request to the endpoint for 24 hours.

User-initiated maintenance

User-initiated VM maintenance via the Azure portal, API, CLI, or PowerShell results in a scheduled event. You then can test the maintenance preparation logic in your application, and your application can prepare for user-initiated maintenance.

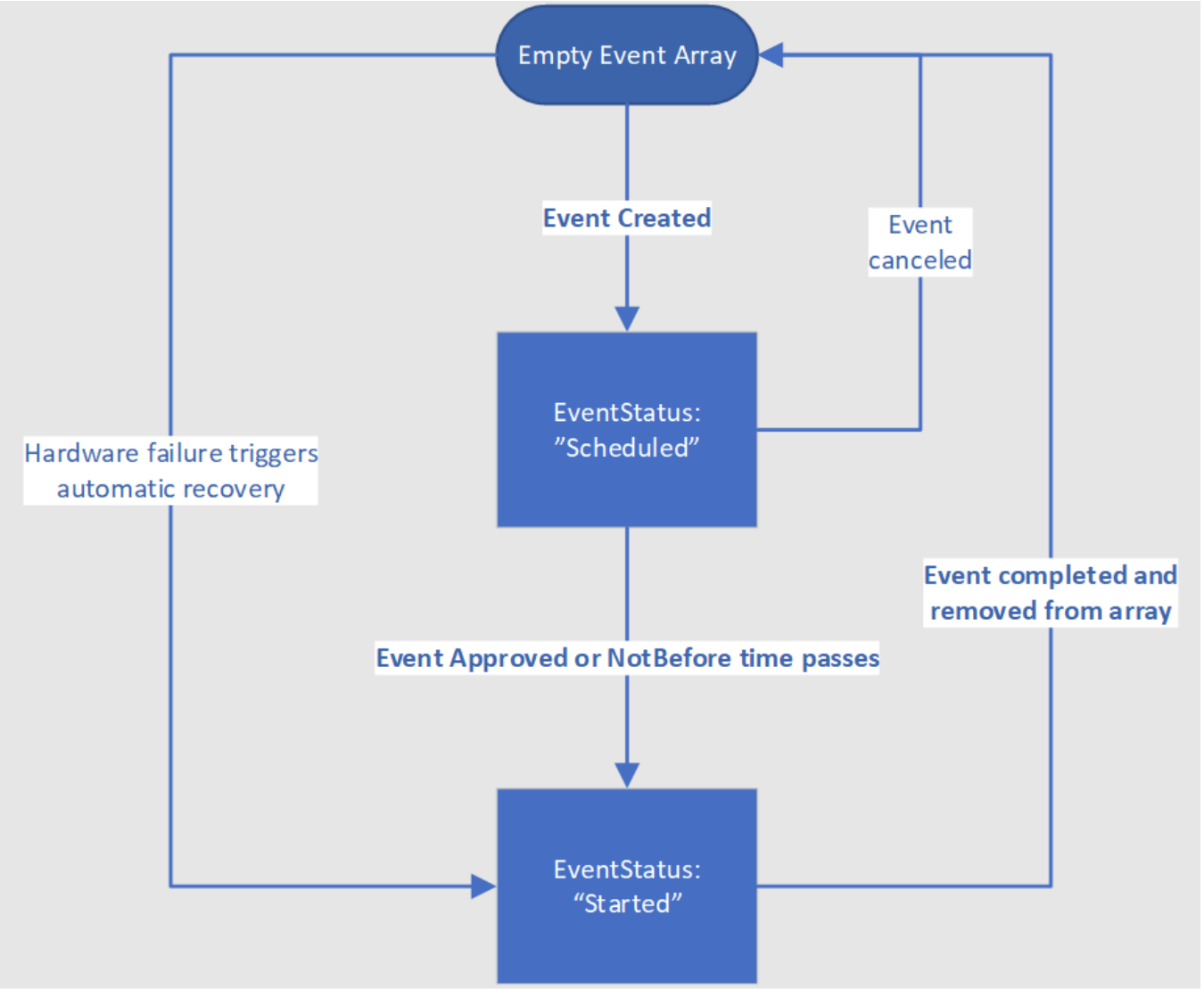
If you restart a VM, an event with the type `Reboot` is scheduled. If you redeploy a VM, an event with the type `Redeploy` is scheduled. Typically events with a user event source can be immediately approved to avoid a delay on user-initiated actions. We advise having a primary and secondary VM communicating and approving user generated scheduled events in case the primary VM becomes unresponsive. Immediately approving events prevents delays in recovering your application back to a good state.

Scheduled events for Virtual Machine Scale Sets Guest OS upgrades or reimages are supported for general purpose VM sizes that support memory preserving updates only. It doesn't work for G, M, N, and H series. Scheduled events for Virtual Machine Scale Sets Guest OS upgrades and reimages are disabled by default. To enable scheduled events for these operations on supported VM sizes, first enable them using OSImageNotificationProfile.

Use the API

High level overview

There are two major components to handling Scheduled Events, preparation and recovery. All current scheduled events impacting a VM are available to read via the IMDS Scheduled Events endpoint. When the event reaches a terminal state, it's removed from the list of events. The following diagram shows the various state transitions that a single scheduled event can experience:



For events in the EventStatus:"Scheduled" state, you need to take steps to prepare your workload. Once the preparation is complete, you should then approve the event using the scheduled event API. Otherwise, the event is automatically approved when the NotBefore time is reached. If the VM is on shared infrastructure, the system will then wait for all other tenants on the same hardware to also approve the job or timeout. Once approvals are gathered from all impacted VMs or the NotBefore time is reached, then Azure generates a new scheduled event payload with EventStatus:"Started" and triggers the start of the maintenance event. When the event reaches a terminal state, it's removed from the list of events. That serves as the signal for the customer to recover their VMs.

The psudeo code demonstrates a process for how to read and manage scheduled events in your application:

```
current_list_of_scheduled_events = get_latest_from_se_endpoint()
#prepare for new events
for each event in current_list_of_scheduled_events:
  if event not in previous_list_of_scheduled_events:
    prepare_for_event(event)
```

```
#recover from completed events
for each event in previous_list_of_scheduled_events:
  if event not in current_list_of_scheduled_events:
    receover_from_event(event)
#prepare for future jobs
previous_list_of_scheduled_events = current_list_of_scheduled_events
```

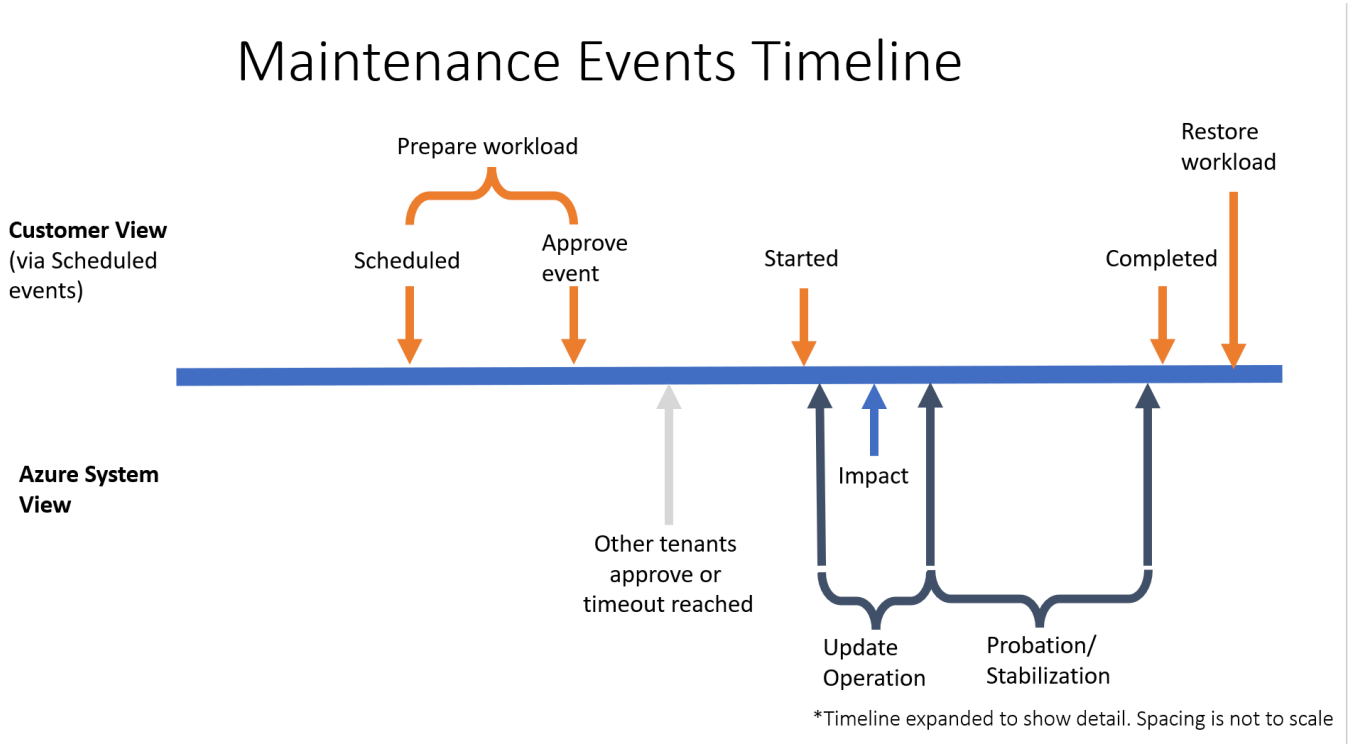
As scheduled events are often used for applications with high availability requirements, there are a few exceptional cases that should be considered:

1. Once a scheduled event is completed and removed from the array, there will be no further impacts without a new event including another EventStatus:"Scheduled" event
2. Azure monitors maintenance operations across the entire fleet and in rare circumstances determines that a maintenance operation too high risk to apply. In that case the scheduled event goes directly from "Scheduled" to being removed from the events array
3. If there is a hardware failure, Azure bypasses the "Scheduled" state and immediately move to the EventStatus:"Started" state.
4. While the event is still in EventStatus:"Started" state, there may be another impact of a shorter duration than what was advertised in the scheduled event.

As part of Azure's availability guarantee, VMs in different fault domains won't be impacted by routine maintenance operations at the same time. However, they may have operations serialized one after another. VMs in one fault domain can receive scheduled events with EventStatus:"Scheduled" shortly after another fault domain's maintenance is completed. Regardless of what architecture you chose, always keep checking for new events pending against your VMs.

While the exact timings of events vary, the following diagram provides a rough guideline for how a typical maintenance operation proceeds:

- EventStatus:"Scheduled" to Approval Timeout: 15 minutes
- Impact Duration: 7 seconds
- EventStatus:"Started" to Completed (event removed from Events array): 10 minutes



All operations impacting VM availability create a scheduled event, however not all scheduled events appear in other Azure surfaces such as Azure Activity Logs or Resource Health. Checking scheduled events regularly will ensure that you have the most up-to-date information about any upcoming impacts to your VMs.

Headers

When you query Metadata Service, you must provide the header `Metadata:true` to ensure the request wasn't unintentionally redirected. The `Metadata:true` header is required for all scheduled events requests. Failure to include the header in the request results in a "Bad Request" response from Metadata Service.

Query for events

You can query for scheduled events by making the following call:

Bash sample

```
curl -H Metadata:true http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01
```

PowerShell sample

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -Uri
"http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01" |
ConvertTo-Json -Depth 64
```

Python sample

```
import json
import requests

metadata_url ="http://169.254.169.254/metadata/scheduledevents"
header = {'Metadata' : 'true'}
query_params = {'api-version':'2020-07-01'}

def get_scheduled_events():
    resp = requests.get(metadata_url, headers = header, params = query_params)
    data = resp.json()
    return data
```

A response contains an array of scheduled events. An empty array means that currently no events are scheduled. In the case where there are scheduled events, the response contains an array of events.

```
{
    "DocumentIncarnation": {IncarnationID},
    "Events": [
        {
            "EventId": {eventID},
            "EventType": "Reboot" | "Redeploy" | "Freeze" | "Preempt" |
"Terminate",
            "ResourceType": "VirtualMachine",
            "Resources": [{resourceName}],
            "EventStatus": "Scheduled" | "Started",
            "NotBefore": {timeInUTC},
            "Description": {eventDescription},
            "EventSource" : "Platform" | "User",
            "DurationInSeconds" : {timeInSeconds},
        }
    ]
}
```

Event properties

Expand table

Property	Description
Document Incarnation	Integer that increases when the events array changes. Documents with the same incarnation contain the same event information, and the incarnation is incremented when an event changes.
EventId	Globally unique identifier for this event. Example: • 602d9444-d2cd-49c7-8624-8643e7171297
EventType	Expected impact this event will cause. Values: • `Freeze`: The Virtual Machine is scheduled to pause for a few seconds. CPU and network connectivity may be suspended, but there's no impact on memory or open files. • `Reboot`: The Virtual Machine is scheduled for reboot (non-persistent memory is lost). In rare cases a VM scheduled for EventType:"Reboot" may experience a freeze event instead of a reboot. Follow the instructions above for how to know if the event is complete and it's safe to restore your workload. • `Redeploy`: The Virtual Machine is scheduled to move to another node (ephemeral disks are lost). • `Preempt`: The Spot Virtual Machine is being deleted (ephemeral disks are lost). This event is made available on a best effort basis • `Terminate`: The virtual machine is scheduled to be deleted.
ResourceType	Type of resource this event affects. Values: • `VirtualMachine`
Resources	List of resources this event affects. Example: • ["FrontEnd_IN_0", "BackEnd_IN_0"]
EventStatus	Status of this event.

Property	Description
	Values: • `Scheduled`: This event is scheduled to start after the time specified in the `NotBefore` property. • `Started`: This event has started. No `Completed` or similar status is ever provided. The event is no longer returned when the event is finished.
NotBefore	Time after which this event can start. The event is guaranteed to not start before this time. Will be blank if the event after the event starts. Example: • Mon, 19 Sep 2016 18:29:47 GMT
Description	Description of this event. Example: • Host server is undergoing maintenance. • Host Server infrastructure is undergoing maintenance. • Virtual machine is being paused because of a memory-preserving Live Migration operation. • Virtual machine is going to be restarted as requested by authorized user. • Host server is undergoing an emergency repair.
EventSource	Initiator of the event. Example: • `Platform`: Platform initiated this event. • `User`: The user initiated this event.
DurationInSeconds	The expected duration of the interruption caused by the event. There may be secondary impacts of a shorter duration during the impact window. Example: • `9`: The interruption caused by the event lasts for 9 seconds. • `0`: The event won't interrupt the VM or impact its availability (for example, update to the network) • `-1`: The default value used if the impact duration is either unknown or not applicable.

Event scheduling

Each event is scheduled a minimum amount of time in the future based on the event type. This time is reflected in an event's `NotBefore` property.

Expand table

EventType	Minimum notice
Freeze	15 minutes
Reboot	15 minutes
Redeploy	10 minutes
Preempt	30 seconds
Terminate	User Configurable: 5 to 15 minutes

This means that you can detect a future schedule of event at least by the minimum notice time before the event occurs. Once an event is scheduled, it will move into the `Started` state after it's been approved or the `NotBefore` time passes. However, in rare cases, the operation is canceled by Azure before it starts. In that case the event is removed from the Events array, and the impact won't occur as previously scheduled.

> **Note**
>
> In some cases, Azure is able to predict host failure due to degraded hardware and attempts to mitigate disruption to your service by scheduling a migration. Affected virtual machines will receive a scheduled event with a `NotBefore` that is typically a few days in the future. The actual time varies depending on the predicted failure risk assessment. Azure tries to give 7 days of advance notice when possible. The actual notice time varies and might be shorter if there's a high chance of the hardware failing imminently. To minimize risk to your service in case the hardware fails before the system-initiated migration, we recommend that you self-redeploy your virtual machine as soon as possible.

> **Note**
>
> In the case the host node experiences a hardware failure, Azure bypasses the minimum notice period and immediately begin the recovery process for affected virtual machines. This reduces recovery time in the case that the affected VMs are unable to respond.

During the recovery process, an event is created for all impacted VMs with `EventType = Reboot` and `EventStatus = Started`.

Polling frequency

You can poll the endpoint for updates as frequently or infrequently as you like. However, the longer the time between requests, the more time you potentially lose to react to an upcoming event. Most events have 5 to 15 minutes of advance notice, although in some cases advance notice might be as little as 30 seconds. To ensure that you have as much time as possible to take mitigating actions, we recommend that you poll the service once per second.

Start an event

After you learn of an upcoming event and finish your logic for graceful shutdown, you can approve the outstanding event by making a `POST` call to Metadata Service with `EventId`. This call indicates to Azure that it can shorten the minimum notification time (when possible). The event may not start immediately upon approval. In some cases Azure requires the approval of all the VMs hosted on the node before proceeding with the event.

The following JSON sample is expected in the `POST` request body. The request should contain a list of `StartRequests`. Each `StartRequest` contains `EventId` for the event you want to expedite:

```
{
    "StartRequests" : [
        {
            "EventId": {EventId}
        }
    ]
}
```

The service always returns a 200 success code if it's passed a valid event ID, even if another VM already approved the event. A 400 error code indicates that the request header or payload was malformed.

> **Note**
>
> Events won't proceed unless they are either approved via a POST message or the NotBefore time elapses. This includes user triggered events such as VM restarts from the Azure portal.

Bash sample

```
curl -H Metadata:true -X POST -d '{"StartRequests": [{"EventId": "f020ba2e-3bc0-
4c40-a10b-86575a9eabd5"}]}' http://169.254.169.254/metadata/scheduledevents?api-
version=2020-07-01
```

PowerShell sample

```
Invoke-RestMethod -Headers @{"Metadata" = "true"} -Method POST -body
'{"StartRequests": [{"EventId": "5DD55B64-45AD-49D3-BBC9-F57D4EA97BD7"}]}' -Uri
http://169.254.169.254/metadata/scheduledevents?api-version=2020-07-01 |
ConvertTo-Json -Depth 64
```

Python sample

```
import json
import requests

def confirm_scheduled_event(event_id):
   # This payload confirms a single event with id event_id
   payload = json.dumps({"StartRequests": [{"EventId": event_id }]})
   response = requests.post("http://169.254.169.254/metadata/scheduledevents",
                            headers =  {'Metadata' : 'true'},
                            params = {'api-version':'2020-07-01'},
                            data = payload)
   return response.status_code
```

> **Note**
>
> Acknowledging an event allows the event to proceed for all `Resources` in the event, not just the VM that acknowledges the event. Therefore, you can choose to elect a leader to coordinate the acknowledgment, which might be as simple as the first machine in the `Resources` field.

Example responses

The following events are an example that from two VMs which were live migrated to another node.

The `DocumentIncarnation` is changing every time there's new information in `Events`. An approval of the event would allow the freeze to proceed for both WestNO_0 and WestNO_1.

JSON


```
{
    "DocumentIncarnation":  1,
    "Events":  [
               ]
}

{
    "DocumentIncarnation":  2,
    "Events":  [
                   {
                       "EventId":  "C7061BAC-AFDC-4513-B24B-AA5F13A16123",
                       "EventStatus":  "Scheduled",
                       "EventType":  "Freeze",
                       "ResourceType":  "VirtualMachine",
                       "Resources":  [
                                         "WestNO_0",
                                         "WestNO_1"
                                     ],
                       "NotBefore":  "Mon, 11 Apr 2022 22:26:58 GMT",
                       "Description":  "Virtual machine is being paused because of
a memory-preserving Live Migration operation.",
                       "EventSource":  "Platform",
                       "DurationInSeconds":  5
                   }
               ]
}

{
    "DocumentIncarnation":  3,
    "Events":  [
                   {
                       "EventId":  "C7061BAC-AFDC-4513-B24B-AA5F13A16123",
                       "EventStatus":  "Started",
                       "EventType":  "Freeze",
                       "ResourceType":  "VirtualMachine",
                       "Resources":  [
                                         "WestNO_0",
                                         "WestNO_1"
                                     ],
                       "NotBefore":  "",
                       "Description":  "Virtual machine is being paused because of
a memory-preserving Live Migration operation.",
                       "EventSource":  "Platform",
                       "DurationInSeconds":  5
                   }
```

```
                ]
}

{
    "DocumentIncarnation":  4,
    "Events":  [
                ]
}
```

Python Sample

The sample queries Metadata Service for scheduled events and approves each outstanding event. You can find the code in the vm-scheduled-events-mock-server repository in the Listener.py file.

Python

```python
#!/usr/bin/python
import json
import requests
from time import sleep

# The URL to access the metadata service
metadata_url ="http://169.254.169.254/metadata/scheduledevents"
# This must be sent otherwise the request will be ignored
header = {'Metadata' : 'true'}
# Current version of the API
query_params = {'api-version':'2020-07-01'}

def get_scheduled_events():
    resp = requests.get(metadata_url, headers = header, params = query_params)
    data = resp.json()
    return data

def confirm_scheduled_event(event_id):
    # This payload confirms a single event with id event_id
    # You can confirm multiple events in a single request if needed
    payload = json.dumps({"StartRequests": [{"EventId": event_id }]})
    response = requests.post(metadata_url,
                             headers= header,
                             params = query_params,
                             data = payload)
    return response.status_code

def log(event):
    # This is an optional placeholder for logging events to your system
    print(event["Description"])
    return

def advanced_sample(last_document_incarnation):
```

```
    # Poll every second to see if there are new scheduled events to process
    # Since some events may have necessarily short warning periods, it is
    # recommended to poll frequently
    found_document_incarnation = last_document_incarnation
    while (last_document_incarnation == found_document_incarnation):
        sleep(1)
        payload = get_scheduled_events()
        found_document_incarnation = payload["DocumentIncarnation"]

    # We recommend processing all events in a document together,
    # even if you won't be actioning on them right away
    for event in payload["Events"]:

        # Events that already started, logged for tracking
        if (event["EventStatus"] == "Started"):
            log(event)

        # Approve all user initiated events. These are typically created by an
        # administrator and approving them immediately can help to avoid delays
        # in admin actions
        elif (event["EventSource"] == "User"):
            confirm_scheduled_event(event["EventId"])

        # For this application, freeze events less that 9 seconds are considered
        # no impact. This will immediately approve them
        elif (event["EventType"] == "Freeze" and
            int(event["DurationInSeconds"]) >= 0  and
            int(event["DurationInSeconds"]) < 9):
            confirm_scheduled_event(event["EventId"])

        # Events that may be impactful (for example reboot or redeploy) may need
custom
        # handling for your application
        else:
            #TODO Custom handling for impactful events
            log(event)
    print("Processed events from document: " + str(found_document_incarnation))
    return found_document_incarnation

def main():
    # This will track the last set of events seen
    last_document_incarnation = "-1"

    input_text = "\
        Press 1 to poll for new events \n\
        Press 2 to exit \n "
    program_exit = False

    while program_exit == False:
        user_input = input(input_text)
        if (user_input == "1"):
            last_document_incarnation = advanced_sample(last_document_incarnation)
        elif (user_input == "2"):
            program_exit = True
```

```
if __name__ == '__main__':
    main()
```

Testing Scheduled Events

Two common ways to test your applications response to scheduled events are manually triggering user imitated events or using a mock server.

You can manually trigger redeploy and reboot events through the Azure Portal or Azure CLI by selecting the 'reboot' or 'redeploy' VM option from the VM blade. This will create an event and send it to your workload.

Operations that impact multiple VMs, such as host updates, cannot be triggered on demand so you can use a mock server instead. The vm-scheduled-events-mock-server provides a framework for testing your application's response to different scenarios by replaying real events flows back for development and testing. By default the server supports 9 different scenarios, all captured from VMs running in Azure and representing the most common cases. The scenarios can be expanded to include more options depending on your applications particular characteristics.

Next steps

- Review the Scheduled Events code samples in the Azure Instance Metadata Scheduled Events GitHub repository.
- Review the Node.js Scheduled Events code samples in Azure Samples GitHub repository.
- Read more about the APIs that are available in the Instance Metadata Service.
- Learn about planned maintenance for Windows virtual machines in Azure.
- Learn how to monitor scheduled events for your VMs through Log Analytics.
- Learn how to log scheduled events using Azure Event Hubs in the Azure Samples GitHub repository.
- Use the vm-scheduled-events-mock-server to test your application's response to common scheduled events scenarios.

Monitor scheduled events for your Windows VMs

Article • 12/19/2024

Applies to: VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

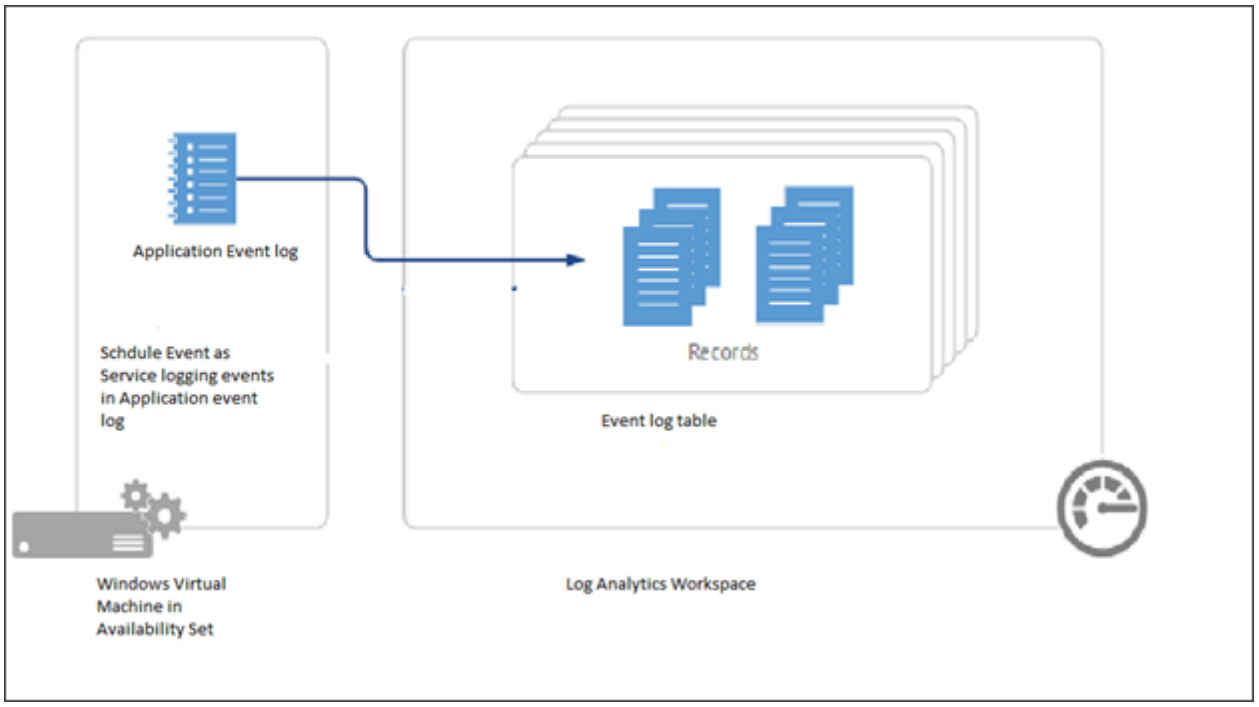
Updates are applied to different parts of Azure every day, to keep the services running on them secure, and up-to-date. In addition to planned updates, unplanned events may also occur. For example, if any hardware degradation or fault is detected, Azure services may need to perform unplanned maintenance. Using live migration, memory preserving updates and keeping a strict limit on the impact of updates, allow these events to be almost transparent to customers. At most, they can cause a few seconds of virtual machine freeze. For some applications however, even a few seconds of virtual machine freeze could cause an impact. Knowing in advance about upcoming Azure maintenance is important to ensure the best experience for those applications. Scheduled Events service provides you with a programmatic interface to be notified about upcoming maintenance, and enables you to gracefully handle the maintenance.

This article shows how you can use scheduled events to be notified about maintenance events that could affect your VMs and build some basic automation that can help with monitoring and analysis.

Routing scheduled events to Log Analytics

Scheduled Events is available as part of the Azure Instance Metadata Service, which is available on every Azure virtual machine. Customers can write automation to query the endpoint of their virtual machines to find scheduled maintenance notifications and perform mitigations, like saving the state and taking the virtual machine out of rotation. We recommend building automation to record the Scheduled Events so you can have an auditing log of Azure maintenance events.

In this article, we walk you through how to capture maintenance Scheduled Events to Log Analytics. Then, we trigger some basic notification actions, like sending an email to your team and getting a historical view of all events that have affected your virtual machines. For the event aggregation and automation we use Log Analytics, but you can use any monitoring solution to collect these logs and trigger automation.



Prerequisites

For this example, you need to create a Windows Virtual Machine in an Availability Set. Scheduled Events provide notifications about changes that can affect any of the virtual machines in your availability set, Cloud Service, Virtual Machine Scale Set or standalone VMs. A service is run that polls for scheduled events on one of the VMs that act as a collector, to get events for all of the other VMs in the availability set.

Don't delete the group resource group at the end of the tutorial.

You also need to create a Log Analytics workspace that we use to aggregate information from the VMs in the availability set.

Set up the environment

You should now have 2 initial VMs in an availability set. Now we need to create a 3rd VM, called `myCollectorVM`, in the same availability set.

Azure PowerShell

```
New-AzVm `
    -ResourceGroupName "myResourceGroupAvailability" `
    -Name "myCollectorVM" `
    -Location "East US" `
    -VirtualNetworkName "myVnet" `
    -SubnetName "mySubnet" `
    -SecurityGroupName "myNetworkSecurityGroup" `
    -OpenPorts 3389 `
```

```
    -PublicIpAddressName "myPublicIpAddress3" `
    -AvailabilitySetName "myAvailabilitySet" `
    -Credential $cred
```

1. Download the installation .zip file of the project from GitHub.

2. Connect to **myCollectorVM** and copy the .zip file to the virtual machine and extract all of the files. On your VM, open a PowerShell prompt. Move your prompt into the folder containing `SchService.ps1`, for example: `PS C:\Users\azureuser\AzureScheduledEventsService-master\AzureScheduledEventsService-master\Powershell>`, and set up the service.

   PowerShell

   ```
   .\SchService.ps1 -Setup
   ```

3. Start the service.

   PowerShell

   ```
   .\SchService.ps1 -Start
   ```

4. Validate the service status and make sure it is running.

   PowerShell

   ```
   .\SchService.ps1 -status
   ```

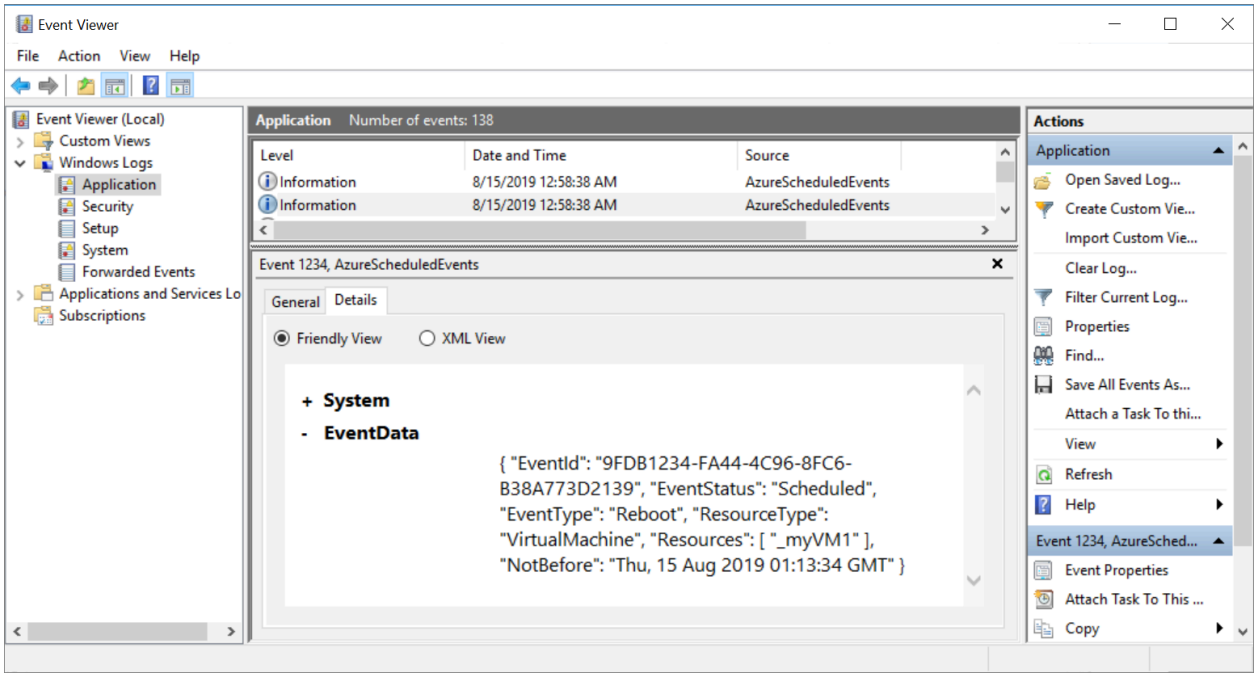
   The validation command should return `Running`.

The service will now start polling every 10 seconds for any scheduled events and approve the events to expedite the maintenance. Freeze, Reboot, Redeploy, and Preempt are the events captured by Schedule events. You can extend the script to trigger some mitigations prior to approving the event.

When any of the above events are captured by Schedule Event service, it will get logged in the Application Event Log Event Status, Event Type, Resources (Virtual machine names) and NotBefore (minimum notice period). You can locate the events with ID 1234 in the Application Event Log.

Once the service is set up and started, it will log events in the Windows Application logs. To verify this works, restart one of the virtual machines in the availability set and you

should see an event being logged in Event viewer in Windows Logs > Application log showing the VM restarted.



When events are captured by the Schedule Event service, it will get logged in the application even log with Event Status, Event Type, Resources (VM name) and NotBefore (minimum notice period). You can locate the events with ID 1234 in the Application Event Log.

> ⓘ **Note**
>
> In this example, the virtual machines were are in an availability set, which enabled us to designate a single virtual machine as the collector to listen and route scheduled events to our log analytics works space. If you have standalone virtual machines, you can run the service on every virtual machine, and then connect them individually to your log analytics workspace.
>
> For our set up, we chose Windows, but you can design a similar solution on Linux.

At any point you can stop/remove the Scheduled Event Service by using the switches `-stop` and `-remove`.

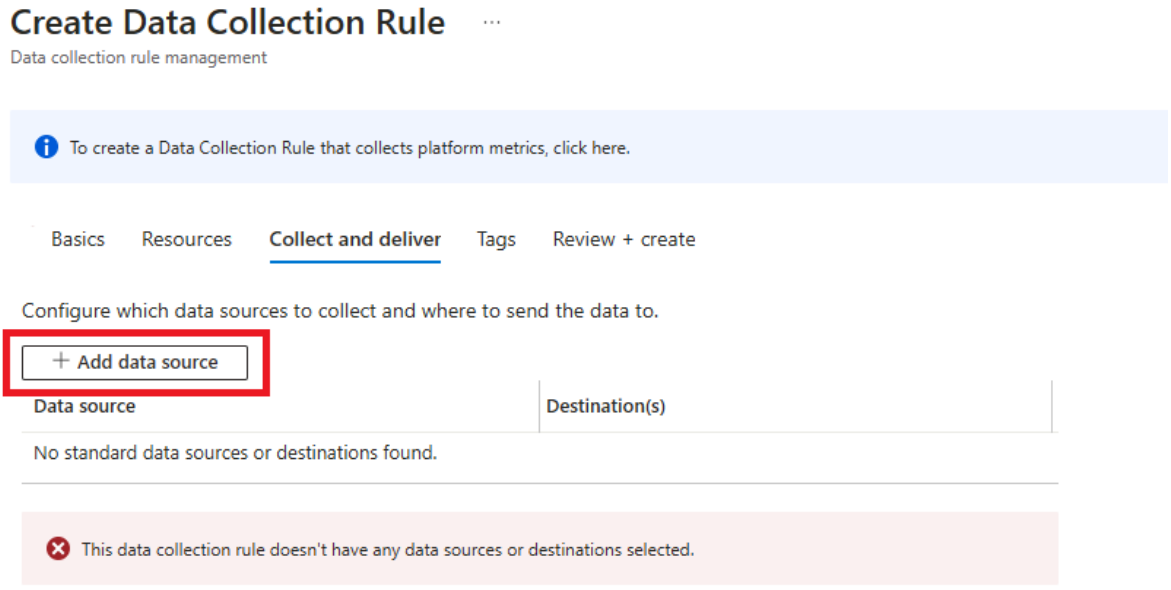
Connect to the Log Analytics Workspace

We now want to connect a Log Analytics Workspace to the collector VM. The Log Analytics workspace acts as a repository and we will configure event log collection to capture the application logs from the collector VM.

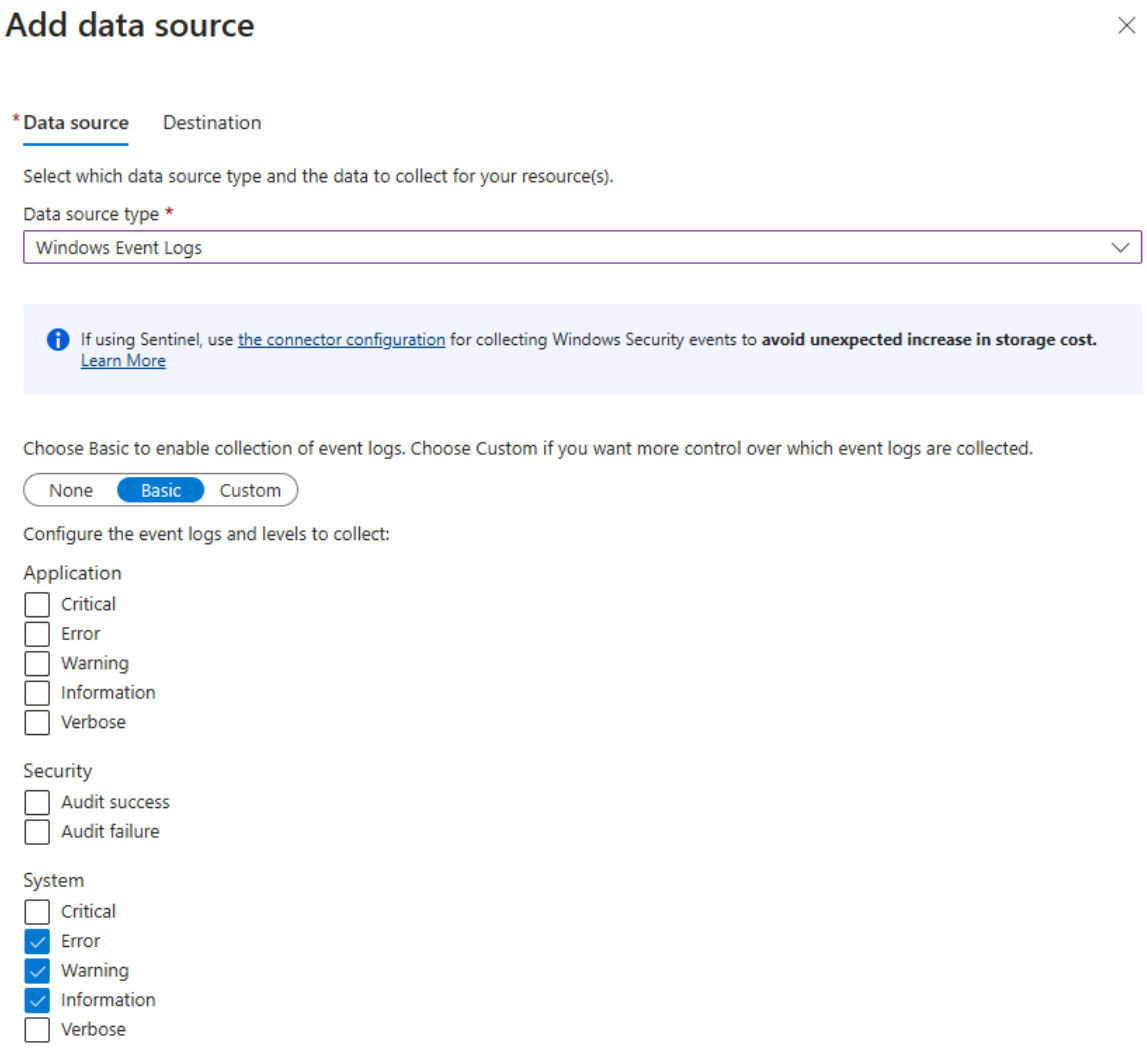
To route the Scheduled Events to the Events Log, which is saved as Application log by our service, you will need to connect your virtual machine to your Log Analytics workspace.

Set up data collection

1. Open the Azure portal.
2. In the search bar at the top, type **Log Analytics Workspaces** and select it from the search results.
3. Choose the workspace you created to open its page.
4. Under Settings, select **Agents** and then click **Virtual Machines**.
5. Under the **Windows servers** tab, click **Data Collection Rules**.
6. Enter the **Collect and Deliver** tab and click **Add data source**



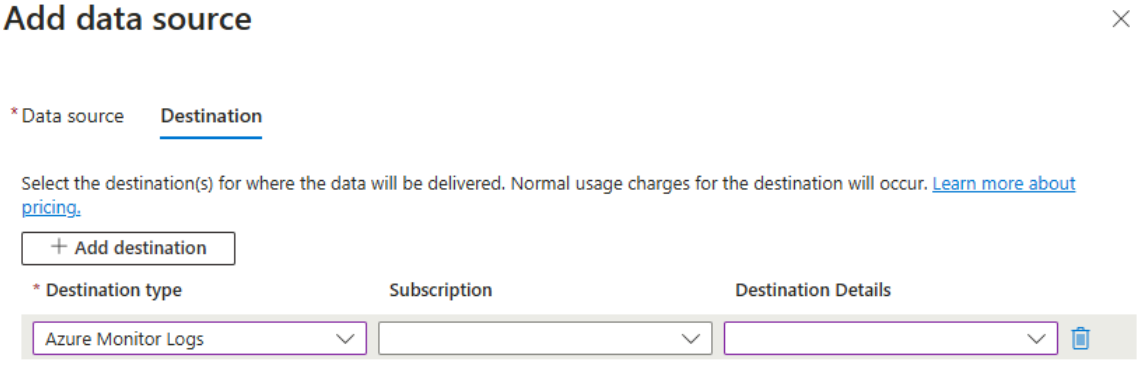
7. Under the **Data source** tab, select **Windows Event Logs** from the dropdown.
8. Select the event logs you'd like to collect. Ensure that **ERROR**, **WARNING**, and **INFORMATION** are selected.



9. Click **Next : Destination >**

10. Under the **Destination** tab, click **Add destination**.

11. Fill out the **Destination Type**, **Subscription**, and **Destination Details** sections with your collector VM and its subscription's details.



12. Once you've selected the correct VM, the Microsoft Monitoring agent will be automatically installed on your virtual machine. It will take a few minutes to

connect your VM to the workspace and install the extension.

> **Note**
>
> There is some delay, and it may take up to 10 minutes before the log is available.

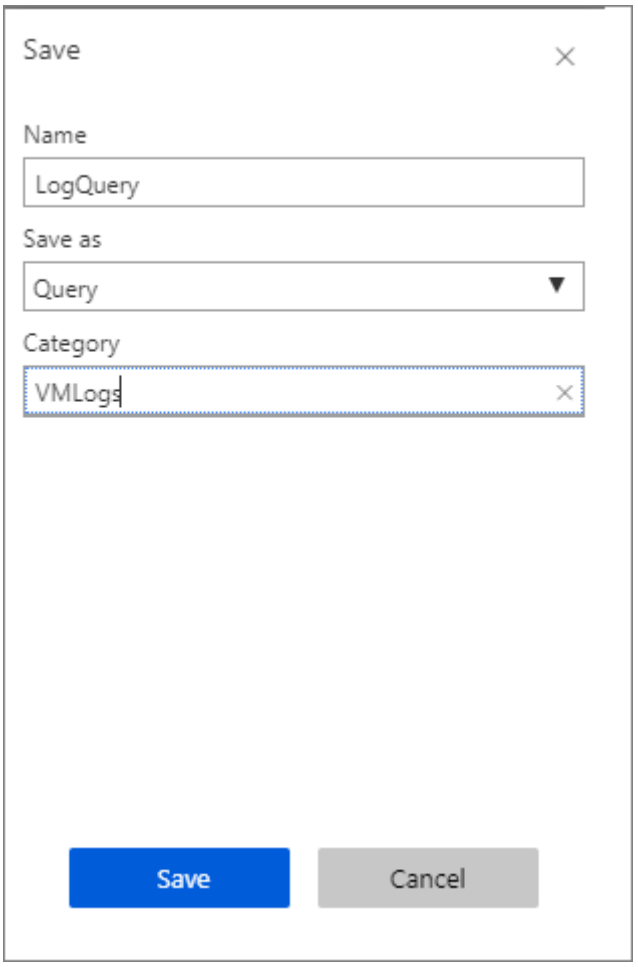
Creating an alert rule with Azure Monitor

Once the events are pushed to Log Analytics, you can run the following query to look for the schedule Events.

1. At the top of the page, select **Logs** and paste the following into the text box:

```
Event
| where EventLog == "Application" and Source contains
"AzureScheduledEvents" and RenderedDescription contains "Scheduled" and
RenderedDescription contains "EventStatus"
| project TimeGenerated, RenderedDescription
| extend ReqJson= parse_json(RenderedDescription)
| extend EventId = ReqJson["EventId"]
,EventStatus = ReqJson["EventStatus"]
,EventType = ReqJson["EventType"]
,NotBefore = ReqJson["NotBefore"]
,ResourceType = ReqJson["ResourceType"]
,Resources = ReqJson["Resources"]
| project-away RenderedDescription,ReqJson
```

2. Select **Save**, and then type `ogQuery` for the name, leave **Query** as the type, type `VMLogs` as the **Category**, and then select **Save**.



3. Select **New alert rule**.
4. In the **Create rule** page, leave `collectorworkspace` as the **Resource**.
5. Under **Condition**, select the entry *Whenever the customer log search is <login undefined>*. The **Configure signal logic** page will open.
6. Under **Threshold value**, enter *0* and then select **Done**.
7. Under **Actions**, select **Create action group**. The **Add action group** page will open.
8. In **Action group name**, type *myActionGroup*.
9. In **Short name**, type *myActionGroup*.
10. In **Resource group**, select **myResourceGroupAvailability**.
11. Under Actions, in **ACTION NAME** type **Email**, and then select **Email/SMS/Push/Voice**. The **Email/SMS/Push/Voice** page will open.
12. Select **Email**, type in your e-mail address, then select **OK**.
13. In the **Add action group** page, select **OK**.

14. In the **Create rule** page, under **ALERT DETAILS**, type *myAlert* for the **Alert rule name**, and then type *Email alert rule* for the **Description**.

15. When you're finished, select **Create alert rule**.

16. Restart one of the VMs in the availability set. Within a few minutes, you should get an e-mail that the alert has been triggered.

To manage your alert rules, go to the resource group, select **Alerts** from the left menu, and then select **Manage alert rules** from the top of the page.

Next steps

To learn more, see the Scheduled events service page on GitHub.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Monitor Azure Virtual Machines

Article • 08/22/2024

This article describes:

- The types of monitoring data you can collect for this service.
- Ways to analyze that data.

> **Note**
>
> If you're already familiar with this service and/or Azure Monitor and just want to know how to analyze monitoring data, see the **Analyze** section near the end of this article.

When you have critical applications and business processes that rely on Azure resources, you need to monitor and get alerts for your system. The Azure Monitor service collects and aggregates metrics and logs from every component of your system. Azure Monitor provides you with a view of availability, performance, and resilience, and notifies you of issues. You can use the Azure portal, PowerShell, Azure CLI, REST API, or client libraries to set up and view monitoring data.

- For more information on Azure Monitor, see the Azure Monitor overview.
- For more information on how to monitor Azure resources in general, see Monitor Azure resources with Azure Monitor.

This article provides an overview of how to monitor the health and performance of Azure virtual machines (VMs).

> **Note**
>
> This article provides basic information to help you get started with monitoring Azure Virtual Machines. For a complete guide to monitoring your entire environment of Azure and hybrid virtual machines, see the **Monitor virtual machines deployment guide**.

Overview: Monitor VM host and guest metrics and logs

You can collect metrics and logs from:

- The **VM host** - This data relates to the Hyper-V session managing the guest operating systems, and includes information about CPU, network, and disk utilization.
- The **VM guest** - This data relates to the operating system and applications running inside the virtual machine.

Host-level data gives you an understanding of the VM's overall performance and load, while the guest-level data gives you visibility into the applications, components, and processes running on the machine and their performance and health. For example, if you're troubleshooting a performance issue, you might start with host metrics to see which VM is under heavy load, and then use guest metrics to drill down into the details of the operating system and application performance.

VM host data

VM host data is available without additional setup.

Expand table

Scenario	Details	Data collection	Available data	Recommendations
VM host metrics and logs	Monitor the stability, health, and efficiency of the physical host on which the VM is running. Scale up or scale down based on the load on your application.	Available by default without any additional setup.	• Host performance metrics • Activity logs • Boot diagnostics	Enable recommended alert rules to be notified when key host metrics deviate from their expected baseline values.

VM guest data

VM guest data lets you analyze and troubleshoot the performance and operational efficiency of workloads running on your VMs. To monitor VM guest data, you need to install Azure Monitor Agent on the VM and set up a data collection rule (DCR). The VM Insights feature automatically installs Azure Monitor Agent on your VM and sets up a default data collection rule for quick and easy onboarding.

Expand table

Scenario	Details	Data collection	Available data	Recommendations
Basic monitoring: key performance indicators	Identify issues related to operating system performance - including CPU and disk utilization - available memory, and network performance by collecting a predefined, basic set of key performance counters.	Enable VM insights	Predefined set of key guest performance counters	• Use as a starting point. • Enable recommended Azure Monitor Baseline Alerts for VMs. • Add guest performance counters of interest and recommended operating system logs, as needed.
Basic monitoring: application component mapping	Map application components on a particular VM and across VMs, and discover the dependencies that exist between application components. This information is important for troubleshooting, optimizing performance, and planning for changes or updates to the application infrastructure.	Enable the Map feature of VM insights	Dependencies between application components running on the VM	

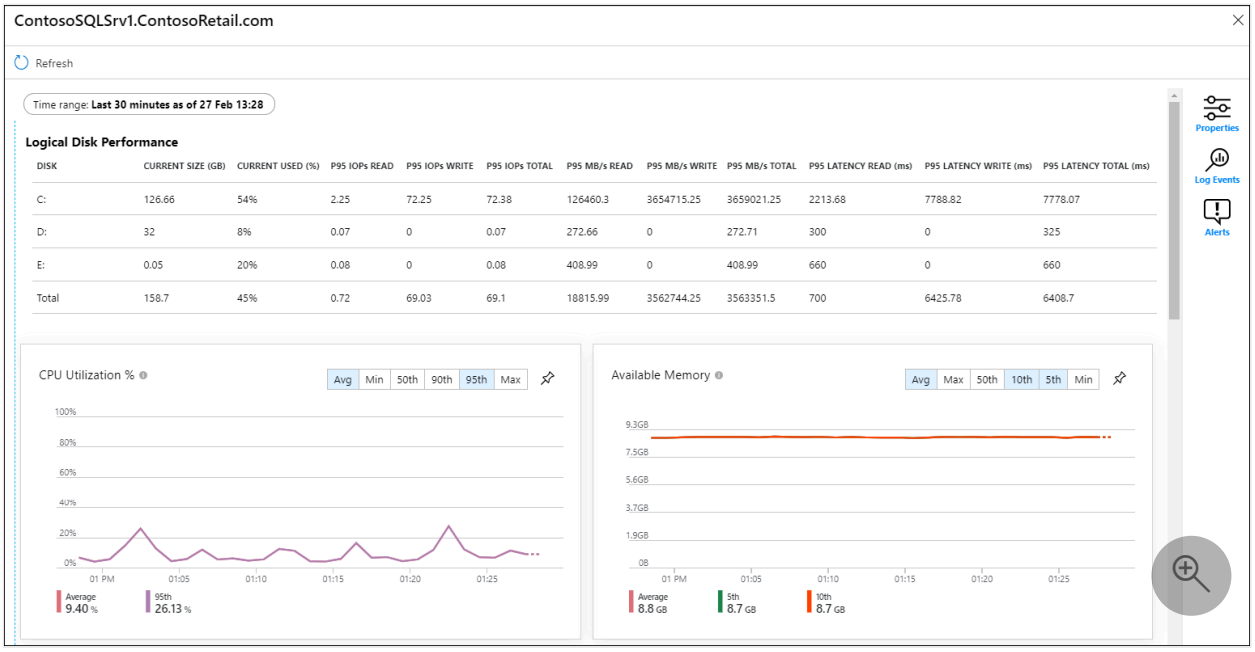
Scenario	Details	Data collection	Available data	Recommendations
VM operating system metrics and logs (recommended)	Monitor application performance and events, resource consumption by specific applications and processes, and operating system-level performance and events. This data is important for troubleshooting application-specific issues, optimizing resource usage within VMs, and ensuring optimal performance for workloads running inside VMs.	Install Azure Monitor Agent on the VM and set up a DCR.	• Guest performance counters • Windows events • Syslog events	• In Windows, collect application logs at the **Critical**, **Error**, and **Warning** levels. • In Linux, collect **LOG_SYSLOG** facility logs at the **LOG_WARNING** level.
Advanced/custom VM guest data	Monitoring of web servers, Linux appliances, and any type of data you want to collect from a VM.	Install Azure Monitor Agent on the VM and set up a DCR.	• IIS logs • SNMP traps • Any data written to a text or JSON file	

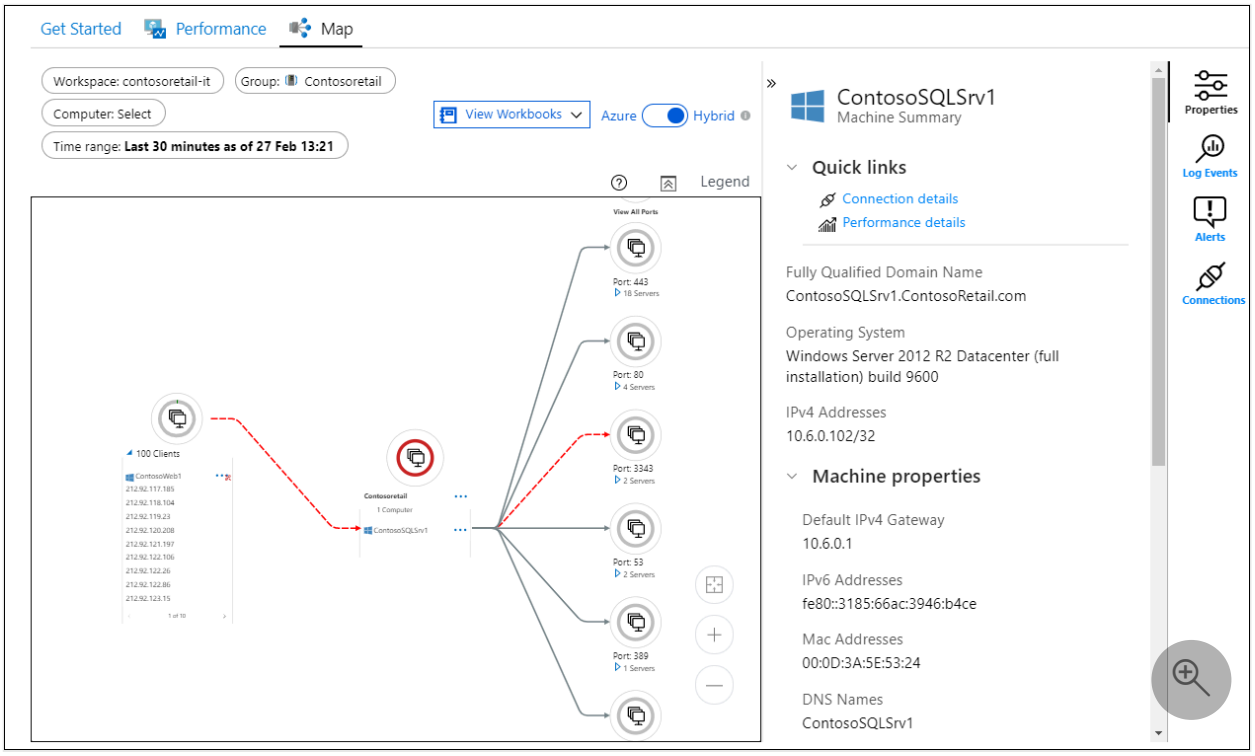
VM insights

VM insights monitors your Azure and hybrid virtual machines in a single interface. VM insights provides the following benefits for monitoring VMs in Azure Monitor:

- Simplified onboarding of the Azure Monitor agent and the Dependency agent, so that you can monitor a virtual machine (VM) guest operating system and workloads.

- Predefined data collection rules that collect the most common set of performance data.
- Predefined trending performance charts and workbooks, so that you can analyze core performance metrics from the virtual machine's guest operating system.
- The Dependency map, which displays processes that run on each virtual machine and the interconnected components with other machines and external sources.





For a tutorial on enabling VM insights for a virtual machine, see Enable monitoring with VM insights for Azure virtual machine. For general information about enabling insights and various methods for onboarding VMs, see Enable VM insights overview.

If you enable VM insights, the Azure Monitor agent is installed and starts sending a predefined set of performance data to Azure Monitor Logs. You can create other data collection rules to collect events and other performance data. To learn how to install the Azure Monitor agent and create a data collection rule (DCR) that defines the data to collect, see Tutorial: Collect guest logs and metrics from an Azure virtual machine.

In VM insights, you can use the Azure Monitor Map feature to view application components that run in Azure or your environment. The Dependency agent VM extension is available for Windows and Linux.

Data storage

For Azure Monitor:

- Metrics data is stored in the Azure Monitor metrics database.
- Log data is stored in the Azure Monitor logs store. Log Analytics is a tool in the Azure portal that can query this store.
- The Azure activity log is a separate store with its own interface in the Azure portal.

You can optionally route metric and activity log data to the Azure Monitor logs store. You can then use Log Analytics to query the data and correlate it with other log data.

Many services can use diagnostic settings to send metric and log data to other storage locations outside Azure Monitor. Examples include Azure Storage, hosted partner systems, and non-Azure partner systems, by using Event Hubs.

For detailed information on how Azure Monitor stores data, see Azure Monitor data platform.

Azure Monitor platform metrics

Azure Monitor provides platform metrics for most services. These metrics are:

- Individually defined for each namespace.
- Stored in the Azure Monitor time-series metrics database.
- Lightweight and capable of supporting near real-time alerting.
- Used to track the performance of a resource over time.

Collection: Azure Monitor collects platform metrics automatically. No configuration is required.

Routing: You can also route some platform metrics to Azure Monitor Logs / Log Analytics so you can query them with other log data. Check the **DS export** setting for

each metric to see if you can use a diagnostic setting to route the metric to Azure Monitor Logs / Log Analytics.

- For more information, see the Metrics diagnostic setting.
- To configure diagnostic settings for a service, see Create diagnostic settings in Azure Monitor.

For a list of all metrics it's possible to gather for all resources in Azure Monitor, see Supported metrics in Azure Monitor.

Platform metrics for Azure VMs include important *host metrics* such as CPU, network, and disk utilization. Host OS metrics relate to the Hyper-V session that's hosting a guest operating system (guest OS) session.

Metrics for the *guest OS* that runs in a VM must be collected through one or more agents, such as the Azure Monitor agent, that run on or as part of the guest OS. Guest OS metrics include performance counters that track guest CPU percentage or memory usage, both of which are frequently used for autoscaling or alerting. For more information, see Guest OS and host OS metrics.

For detailed information about how the Azure Monitor agent collects VM monitoring data, see Monitor virtual machines with Azure Monitor: Collect data.

For a list of available metrics for Virtual Machines, see Virtual Machines monitoring data reference.

Azure activity log

The activity log contains subscription-level events that track operations for each Azure resource as seen from outside that resource; for example, creating a new resource or starting a virtual machine.

Collection: Activity log events are automatically generated and collected in a separate store for viewing in the Azure portal.

Routing: You can send activity log data to Azure Monitor Logs so you can analyze it alongside other log data. Other locations such as Azure Storage, Azure Event Hubs, and certain Microsoft monitoring partners are also available. For more information on how to route the activity log, see Overview of the Azure activity log.

Data collection rules

Data collection rules (DCRs) define data collection from the Azure Monitor Agent and are stored in your Azure subscription. For VMs, DCRs define data such as events and performance counters to collect, and specify locations such as Log Analytics workspaces to send the data. A single VM can be associated with multiple DCRs, and a single DCR can be associated with multiple VMs.

VM insights DCR

VM insights creates a DCR that collects common performance counters for the client operating system and sends them to the InsightsMetrics table in the Log Analytics workspace. For a list of performance counters collected, see How to query logs from VM insights. You can use this DCR with other VMs instead of creating a new DCR for each VM.

You can also optionally enable collection of processes and dependencies, which populates the following tables and enables the VM insights Map feature.

- VMBoundPort: Traffic for open server ports on the machine
- VMComputer: Inventory data for the machine
- VMConnection: Traffic for inbound and outbound connections to and from the machine
- VMProcess: Processes running on the machine

Analyze monitoring data

There are many tools for analyzing monitoring data.

Azure Monitor tools

Azure Monitor supports the following basic tools:

- Metrics explorer, a tool in the Azure portal that allows you to view and analyze metrics for Azure resources. For more information, see Analyze metrics with Azure Monitor metrics explorer.
- Log Analytics, a tool in the Azure portal that allows you to query and analyze log data by using the Kusto query language (KQL). For more information, see Get started with log queries in Azure Monitor.
- The activity log, which has a user interface in the Azure portal for viewing and basic searches. To do more in-depth analysis, you have to route the data to Azure Monitor logs and run more complex queries in Log Analytics.

Tools that allow more complex visualization include:

- Dashboards that let you combine different kinds of data into a single pane in the Azure portal.
- Workbooks, customizable reports that you can create in the Azure portal. Workbooks can include text, metrics, and log queries.
- Grafana, an open platform tool that excels in operational dashboards. You can use Grafana to create dashboards that include data from multiple sources other than Azure Monitor.
- Power BI, a business analytics service that provides interactive visualizations across various data sources. You can configure Power BI to automatically import log data from Azure Monitor to take advantage of these visualizations.

Azure Monitor export tools

You can get data out of Azure Monitor into other tools by using the following methods:

- **Metrics:** Use the REST API for metrics to extract metric data from the Azure Monitor metrics database. The API supports filter expressions to refine the data retrieved. For more information, see Azure Monitor REST API reference.
- **Logs:** Use the REST API or the associated client libraries.
- Another option is the workspace data export.

To get started with the REST API for Azure Monitor, see Azure monitoring REST API walkthrough.

Kusto queries

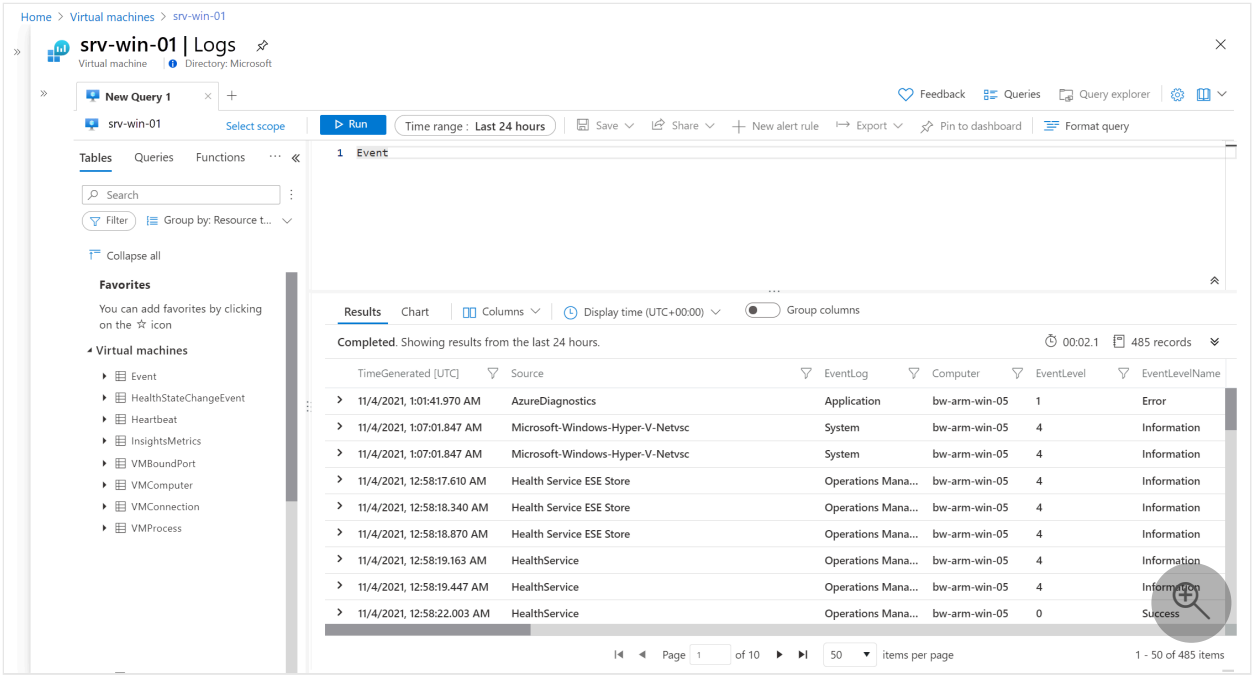
You can analyze monitoring data in the Azure Monitor Logs / Log Analytics store by using the Kusto query language (KQL).

> **Important**
>
> When you select **Logs** from the service's menu in the portal, Log Analytics opens with the query scope set to the current service. This scope means that log queries will only include data from that type of resource. If you want to run a query that includes data from other Azure services, select **Logs** from the **Azure Monitor** menu. See **Log query scope and time range in Azure Monitor Log Analytics** for details.

For a list of common queries for any service, see the Log Analytics queries interface.

To analyze log data that you collect from your VMs, you can use log queries in Log Analytics. Several built-in queries for VMs are available to use, or you can create your own queries. You can interactively work with the results of these queries, include them in a workbook to make them available to other users, or generate alerts based on their results.

To access built-in Kusto queries for your VM, select **Logs** in the **Monitoring** section of the left navigation on your VM's Azure portal page. On the **Logs** page, select the **Queries** tab, and then select the query to run.



Alerts

Azure Monitor alerts proactively notify you when specific conditions are found in your monitoring data. Alerts allow you to identify and address issues in your system before your customers notice them. For more information, see Azure Monitor alerts.

There are many sources of common alerts for Azure resources. For examples of common alerts for Azure resources, see Sample log alert queries. The Azure Monitor Baseline Alerts (AMBA) site provides a semi-automated method of implementing important platform metric alerts, dashboards, and guidelines. The site applies to a continually expanding subset of Azure services, including all services that are part of the Azure Landing Zone (ALZ).

The common alert schema standardizes the consumption of Azure Monitor alert notifications. For more information, see Common alert schema.

Types of alerts

You can alert on any metric or log data source in the Azure Monitor data platform. There are many different types of alerts depending on the services you're monitoring and the monitoring data you're collecting. Different types of alerts have various benefits and drawbacks. For more information, see Choose the right monitoring alert type.

The following list describes the types of Azure Monitor alerts you can create:

- Metric alerts evaluate resource metrics at regular intervals. Metrics can be platform metrics, custom metrics, logs from Azure Monitor converted to metrics, or Application Insights metrics. Metric alerts can also apply multiple conditions and dynamic thresholds.
- Log alerts allow users to use a Log Analytics query to evaluate resource logs at a predefined frequency.
- Activity log alerts trigger when a new activity log event occurs that matches defined conditions. Resource Health alerts and Service Health alerts are activity log alerts that report on your service and resource health.

Some Azure services also support smart detection alerts, Prometheus alerts, or recommended alert rules.

For some services, you can monitor at scale by applying the same metric alert rule to multiple resources of the same type that exist in the same Azure region. Individual notifications are sent for each monitored resource. For supported Azure services and clouds, see Monitor multiple resources with one alert rule.

You can create a single multi-resource alert rule that applies to all VMs in a particular resource group or subscription within the same region. See Create availability alert rule for Azure virtual machine (preview) for a tutorial using the availability metric.

Recommended alert rules

For some Azure services, you can enable recommended out-of-the-box alert rules.

The system compiles a list of recommended alert rules based on:

- The resource provider's knowledge of important signals and thresholds for monitoring the resource.
- Data that tells what customers commonly alert on for this resource.

> **Note**
>
> Recommended alert rules are available for:

- Virtual machines
- Azure Kubernetes Service (AKS) resources
- Log Analytics workspaces

Recommended alert rules for Azure VMs include the VM availability metric, which alerts when a VM stops running.

For more information, see Tutorial: Enable recommended alert rules for Azure virtual machine.

Common alert rules

To see common VM log alert rules in the Azure portal, go to the **Queries** pane in Log Analytics. For **Resource type**, enter **Virtual machines**, and for **Type**, enter **Alerts**.

For a list and discussion of common Virtual Machines alert rules, see Common alert rules.

Advisor recommendations

For some services, if critical conditions or imminent changes occur during resource operations, an alert displays on the service **Overview** page in the portal. You can find more information and recommended fixes for the alert in **Advisor recommendations** under **Monitoring** in the left menu. During normal operations, no advisor recommendations display.

For more information on Azure Advisor, see Azure Advisor overview.

Other VM monitoring options

Azure VMs has the following non-Azure Monitor monitoring options:

Boot diagnostics

Boot diagnostics is a debugging feature for Azure VMs that allows you to diagnose VM boot failures by collecting serial log information and screenshots of a VM as it boots up. When you create a VM in the Azure portal, boot diagnostics is enabled by default. For more information, see Azure boot diagnostics.

Troubleshoot VM performance issues with Performance Diagnostics

The Performance Diagnostics tool helps troubleshoot performance issues on Windows or Linux virtual machines by quickly diagnosing and providing insights on issues it currently finds on your machines. It supports two different modes:

- **Continuous diagnostics (preview)** collects data at five-second intervals and reports actionable insights about high resource usage every five minutes.
- **On-demand diagnostics** helps you troubleshoot an ongoing performance issue with more in-depth data, insights, and recommendations based on data collected at a single point in time.

For more information, go to Troubleshoot performance issues on Azure virtual machines using Performance Diagnostics.

Related content

- For a reference of the metrics, logs, and other important values for Virtual Machines, see Virtual Machines monitoring data reference.
- For general details about monitoring Azure resources, see Monitor Azure resources with Azure Monitor.
- For guidance based on the five pillars of the Azure Well-Architected Framework, see Best practices for monitoring virtual machines in Azure Monitor.
- To get started with VM insights, see Overview of VM insights.
- To learn how to collect and analyze VM host and client metrics and logs, see the training course Monitor your Azure virtual machines with Azure Monitor.
- For a complete guide to monitoring Azure and hybrid VMs, see the Monitor virtual machines deployment guide.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure Virtual Machines monitoring data reference

Article • 01/24/2025

This article contains all the monitoring reference information for this service.

See Monitor Azure Virtual Machines for details on the data you can collect for Azure Virtual Machines and how to use it.

Metrics

This section lists all the automatically collected platform metrics for this service. These metrics are also part of the global list of all platform metrics supported in Azure Monitor.

For information on metric retention, see Azure Monitor Metrics overview.

> **Important**
>
> Metrics for the guest operating system (guest OS) that runs in a virtual machine (VM) aren't listed here. Guest OS metrics must be collected through one or more agents that run on or as part of the guest operating system. Guest OS metrics include performance counters that track guest CPU percentage or memory usage, both of which are frequently used for autoscaling or alerting.
>
> Host OS metrics are available and listed in the following tables. Host OS metrics relate to the Hyper-V session that's hosting your guest OS session. For more information, see **Guest OS and host OS metrics**.

Supported metrics for Microsoft.Compute/virtualMachines

The following table lists the metrics available for the Microsoft.Compute/virtualMachines resource type.

- All columns might not be present in every table.
- Some columns might be beyond the viewing area of the page. Select **Expand table** to view all available columns.

Table headings

- **Category** - The metrics group or classification.
- **Metric** - The metric display name as it appears in the Azure portal.

- **Name in REST API** - The metric name as referred to in the REST API.
- **Unit** - Unit of measure.
- **Aggregation** - The default aggregation type. Valid values: Average (Avg), Minimum (Min), Maximum (Max), Total (Sum), Count.
- **Dimensions** - Dimensions available for the metric.
- **Time Grains** - Intervals at which the metric is sampled. For example, `PT1M` indicates that the metric is sampled every minute, `PT30M` every 30 minutes, `PT1H` every hour, and so on.
- **DS Export**- Whether the metric is exportable to Azure Monitor Logs via diagnostic settings. For information on exporting metrics, see Create diagnostic settings in Azure Monitor.

Expand table

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
Available Memory Bytes (Preview) Amount of physical memory, in bytes, immediately available for allocation to a process or for system use in the Virtual Machine	`Available Memory Bytes`	Bytes	Average	<none>	PT1M	Yes
CPU Credits Consumed Total number of credits consumed by the Virtual Machine. Only available on B-series burstable VMs	`CPU Credits Consumed`	Count	Average	<none>	PT1M	Yes
CPU Credits Remaining Total number	`CPU Credits Remaining`	Count	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
of credits available to burst. Only available on B-series burstable VMs						
Data Disk Bandwidth Consumed Percentage Percentage of data disk bandwidth consumed per minute. Only available on VM series that support premium storage.	`Data Disk Bandwidth Consumed Percentage`	Percent	Average	`LUN`	PT1M	Yes
Data Disk IOPS Consumed Percentage Percentage of data disk I/Os consumed per minute. Only available on VM series that support premium storage.	`Data Disk IOPS Consumed Percentage`	Percent	Average	`LUN`	PT1M	Yes
Data Disk Latency (Preview) Average time to complete each IO during monitoring period for Data Disk. Values are in milliseconds.	`Data Disk Latency`	Milliseconds	Average	`LUN`	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
Data Disk Max Burst Bandwidth Maximum bytes per second throughput Data Disk can achieve with bursting	`Data Disk Max Burst Bandwidth`	Count	Average	`LUN`	PT1M	Yes
Data Disk Max Burst IOPS Maximum IOPS Data Disk can achieve with bursting	`Data Disk Max Burst IOPS`	Count	Average	`LUN`	PT1M	Yes
Data Disk Queue Depth Data Disk Queue Depth(or Queue Length)	`Data Disk Queue Depth`	Count	Average	`LUN`	PT1M	Yes
Data Disk Read Bytes/Sec Bytes/Sec read from a single disk during monitoring period	`Data Disk Read Bytes/sec`	BytesPerSecond	Average	`LUN`	PT1M	Yes
Data Disk Read Operations/Sec Read IOPS from a single disk during monitoring period	`Data Disk Read Operations/Sec`	CountPerSecond	Average	`LUN`	PT1M	Yes
Data Disk Target Bandwidth	`Data Disk Target Bandwidth`	Count	Average	`LUN`	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
Baseline bytes per second throughput Data Disk can achieve without bursting						
Data Disk Target IOPS Baseline IOPS Data Disk can achieve without bursting	`Data Disk Target IOPS`	Count	Average	`LUN`	PT1M	Yes
Data Disk Used Burst BPS Credits Percentage Percentage of Data Disk burst bandwidth credits used so far	`Data Disk Used Burst BPS Credits Percentage`	Percent	Average	`LUN`	PT1M	Yes
Data Disk Used Burst IO Credits Percentage Percentage of Data Disk burst I/O credits used so far	`Data Disk Used Burst IO Credits Percentage`	Percent	Average	`LUN`	PT1M	Yes
Data Disk Write Bytes/Sec Bytes/Sec written to a single disk during monitoring period	`Data Disk Write Bytes/sec`	BytesPerSecond	Average	`LUN`	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
Data Disk Write Operations/Sec Write IOPS from a single disk during monitoring period	`Data Disk Write Operations/Sec`	CountPerSecond	Average	`LUN`	PT1M	Yes
Disk Read Bytes Bytes read from disk during monitoring period	`Disk Read Bytes`	Bytes	Total (Sum)	<none>	PT1M	Yes
Disk Read Operations/Sec Disk Read IOPS	`Disk Read Operations/Sec`	CountPerSecond	Average	<none>	PT1M	Yes
Disk Write Bytes Bytes written to disk during monitoring period	`Disk Write Bytes`	Bytes	Total (Sum)	<none>	PT1M	Yes
Disk Write Operations/Sec Disk Write IOPS	`Disk Write Operations/Sec`	CountPerSecond	Average	<none>	PT1M	Yes
Inbound Flows Inbound Flows are number of current flows in the inbound direction (traffic going into the VM)	`Inbound Flows`	Count	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
Inbound Flows Maximum Creation Rate The maximum creation rate of inbound flows (traffic going into the VM)	`Inbound Flows Maximum Creation Rate`	CountPerSecond	Average	<none>	PT1M	Yes
Network In Billable (Deprecated) The number of billable bytes received on all network interfaces by the Virtual Machine(s) (Incoming Traffic) (Deprecated)	`Network In`	Bytes	Total (Sum)	<none>	PT1M	Yes
Network In Total The number of bytes received on all network interfaces by the Virtual Machine(s) (Incoming Traffic)	`Network In Total`	Bytes	Total (Sum)	<none>	PT1M	Yes
Network Out Billable (Deprecated) The number of billable bytes out on all network interfaces by the Virtual Machine(s)	`Network Out`	Bytes	Total (Sum)	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
(Outgoing Traffic) (Deprecated)						
Network Out Total The number of bytes out on all network interfaces by the Virtual Machine(s) (Outgoing Traffic)	`Network Out Total`	Bytes	Total (Sum)	<none>	PT1M	Yes
OS Disk Bandwidth Consumed Percentage Percentage of operating system disk bandwidth consumed per minute. Only available on VM series that support premium storage.	`OS Disk Bandwidth Consumed Percentage`	Percent	Average	`LUN`	PT1M	Yes
OS Disk IOPS Consumed Percentage Percentage of operating system disk I/Os consumed per minute. Only available on VM series that support premium storage.	`OS Disk IOPS Consumed Percentage`	Percent	Average	`LUN`	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
OS Disk Latency (Preview) Average time to complete each IO during monitoring period for OS Disk. Values are in milliseconds.	`OS Disk Latency`	Milliseconds	Average	<none>	PT1M	Yes
OS Disk Max Burst Bandwidth Maximum bytes per second throughput OS Disk can achieve with bursting	`OS Disk Max Burst Bandwidth`	Count	Average	`LUN`	PT1M	Yes
OS Disk Max Burst IOPS Maximum IOPS OS Disk can achieve with bursting	`OS Disk Max Burst IOPS`	Count	Average	`LUN`	PT1M	Yes
OS Disk Queue Depth OS Disk Queue Depth(or Queue Length)	`OS Disk Queue Depth`	Count	Average	<none>	PT1M	Yes
OS Disk Read Bytes/Sec Bytes/Sec read from a single disk during monitoring period for OS disk	`OS Disk Read Bytes/sec`	BytesPerSecond	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
OS Disk Read Operations/Sec Read IOPS from a single disk during monitoring period for OS disk	`OS Disk Read Operations/Sec`	CountPerSecond	Average	<none>	PT1M	Yes
OS Disk Target Bandwidth Baseline bytes per second throughput OS Disk can achieve without bursting	`OS Disk Target Bandwidth`	Count	Average	`LUN`	PT1M	Yes
OS Disk Target IOPS Baseline IOPS OS Disk can achieve without bursting	`OS Disk Target IOPS`	Count	Average	`LUN`	PT1M	Yes
OS Disk Used Burst BPS Credits Percentage Percentage of OS Disk burst bandwidth credits used so far	`OS Disk Used Burst BPS Credits Percentage`	Percent	Average	`LUN`	PT1M	Yes
OS Disk Used Burst IO Credits Percentage Percentage of OS Disk burst I/O credits used so far	`OS Disk Used Burst IO Credits Percentage`	Percent	Average	`LUN`	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
OS Disk Write Bytes/Sec Bytes/Sec written to a single disk during monitoring period for OS disk	`OS Disk Write Bytes/sec`	BytesPerSecond	Average	<none>	PT1M	Yes
OS Disk Write Operations/Sec Write IOPS from a single disk during monitoring period for OS disk	`OS Disk Write Operations/Sec`	CountPerSecond	Average	<none>	PT1M	Yes
Outbound Flows Outbound Flows are number of current flows in the outbound direction (traffic going out of the VM)	`Outbound Flows`	Count	Average	<none>	PT1M	Yes
Outbound Flows Maximum Creation Rate The maximum creation rate of outbound flows (traffic going out of the VM)	`Outbound Flows Maximum Creation Rate`	CountPerSecond	Average	<none>	PT1M	Yes
Percentage CPU The percentage of allocated	`Percentage CPU`	Percent	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
compute units that are currently in use by the Virtual Machine(s)						
Premium Data Disk Cache Read Hit Premium Data Disk Cache Read Hit	`Premium Data Disk Cache Read Hit`	Percent	Average	`LUN`	PT1M	Yes
Premium Data Disk Cache Read Miss Premium Data Disk Cache Read Miss	`Premium Data Disk Cache Read Miss`	Percent	Average	`LUN`	PT1M	Yes
Premium OS Disk Cache Read Hit Premium OS Disk Cache Read Hit	`Premium OS Disk Cache Read Hit`	Percent	Average	<none>	PT1M	Yes
Premium OS Disk Cache Read Miss Premium OS Disk Cache Read Miss	`Premium OS Disk Cache Read Miss`	Percent	Average	<none>	PT1M	Yes
Temp Disk Latency (Preview) Average time to complete each IO during monitoring period for Temp Disk.	`Temp Disk Latency`	Milliseconds	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
Values are in milliseconds.						
Temp Disk Queue Depth Temp Disk Queue Depth(or Queue Length).	`Temp Disk Queue Depth`	Count	Average	<none>	PT1M	Yes
Temp Disk Read Bytes/Sec Bytes/Sec read from a single disk during monitoring period for Temp Disk.	`Temp Disk Read Bytes/sec`	BytesPerSecond	Average	<none>	PT1M	Yes
Temp Disk Read Operations/Sec Read IOPS from a single disk during monitoring period for Temp Disk.	`Temp Disk Read Operations/Sec`	CountPerSecond	Average	<none>	PT1M	Yes
Temp Disk Write Bytes/Sec Bytes/Sec written to a single disk during monitoring period for Temp Disk.	`Temp Disk Write Bytes/sec`	BytesPerSecond	Average	<none>	PT1M	Yes
Temp Disk Write Operations/Sec Write IOPS	`Temp Disk Write Operations/Sec`	CountPerSecond	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
from a single disk during monitoring period for Temp Disk.						
VM Cached Bandwidth Consumed Percentage Percentage of cached disk bandwidth consumed by the VM. Only available on VM series that support premium storage.	`VM Cached Bandwidth Consumed Percentage`	Percent	Average	<none>	PT1M	Yes
VM Cached IOPS Consumed Percentage Percentage of cached disk IOPS consumed by the VM. Only available on VM series that support premium storage.	`VM Cached IOPS Consumed Percentage`	Percent	Average	<none>	PT1M	Yes
VM Cached Used Burst BPS Credits Percentage Percentage of Cached Burst BPS Credits used by the VM.	`VM Local Used Burst BPS Credits Percentage`	Percent	Average, Minimum, Maximum	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
VM Cached Used Burst IO Credits Percentage Percentage of Cached Burst IO Credits used by the VM.	`VM Local Used Burst IO Credits Percentage`	Percent	Average, Minimum, Maximum	<none>	PT1M	Yes
VM Uncached Used Burst BPS Credits Percentage Percentage of Uncached Burst BPS Credits used by the VM.	`VM Remote Used Burst BPS Credits Percentage`	Percent	Average, Minimum, Maximum	<none>	PT1M	Yes
VM Uncached Used Burst IO Credits Percentage Percentage of Uncached Burst IO Credits used by the VM.	`VM Remote Used Burst IO Credits Percentage`	Percent	Average, Minimum, Maximum	<none>	PT1M	Yes
VM Uncached Bandwidth Consumed Percentage Percentage of uncached disk bandwidth consumed by the VM. Only available on VM series that support premium storage.	`VM Uncached Bandwidth Consumed Percentage`	Percent	Average	<none>	PT1M	Yes

Metric	Name in REST API	Unit	Aggregation	Dimensions	Time Grains	DS Export
VM Uncached IOPS Consumed Percentage Percentage of uncached disk IOPS consumed by the VM. Only available on VM series that support premium storage.	`VM Uncached IOPS Consumed Percentage`	Percent	Average	<none>	PT1M	Yes
VM Availability Metric (Preview) Measure of Availability of Virtual machines over time.	`VmAvailabilityMetric`	Count	Average, Minimum, Maximum	`Context`	PT1M	Yes

For an example that shows how to collect the *Percentage CPU* metric from a VM, see Get virtual machine usage metrics using the REST API.

VM availability metric (preview)

The VM availability metric is currently in public preview. This metric value indicates whether a machine is currently running and available. You can use the metric to trend availability over time and to alert if the machine is stopped. VM availability displays the following values.

The VM availability metric is computed based on an aggregate of different signals from the host.

To learn how to use the VM availability metric to monitor Azure Virtual Machine availability, see Use Azure Monitor to monitor Azure Virtual Machine availability.

Expand table

Value	Description
1	VM is running and available.
0	VM is unavailable. The VM could be stopped or rebooting. If you shut down a VM from within the VM, it emits this value.
Null (dashed line)	State of the VM is unknown. If you stop a VM from the Azure portal, CLI, or PowerShell, it immediately stops emitting the availability metric, and you see null values.

Context dimension informs whether VM availability was influenced due to Azure or user orchestrated activity. It can assume values of *Platform*, *Customer*, or *Unknown*.

Expand table

Display name	Description
Aggregation	*Average* (default aggregation): for prioritized investigations based on extent of downtime incurred. *Min*: immediately pinpoints all the times where the VM was unavailable. *Max*: immediately pinpoints all the instances where the VM was available. For more information on chart range, granularity, and data aggregation, see Azure Monitor metrics aggregation and display explained.
Data retention	Data for the VM availability metric is stored for 93 days to help trend analysis and historical lookback.
Pricing	Refer to the Pricing breakdown, specifically in the *Metrics* and *Alert Rules* sections.

Metric dimensions

For information about what metric dimensions are, see Multi-dimensional metrics.

The dimension Logical Unit Number (`LUN`) is associated with some of the preceding metrics.

Azure Monitor Logs tables

This section lists the Azure Monitor Logs tables relevant to this service, which are available for query by Log Analytics using Kusto queries. The tables contain resource log data and possibly more depending on what is collected and routed to them.

Expand table

Table	Categories	Data collection method	Supports basic log plan	Queries
ADAssessmentRecommendation Recommendations generated by AD assessments that are started through a scheduled task. When you schedule the assessment it runs by default every seven days and uploads the data into Azure Log Analytics.	workloads	Active Directory On-Demand Assessment	No	Yes
AzureActivity Entries from the Azure Activity log that provides insight into any subscription-level or management group level events that have occurred in Azure.	resources, audit, security	Export Activity log	No	Yes
CommonSecurityLog This table is for collecting events in the Common Event Format, that are most often sent from different security appliances such as Check Point, Palo Alto and more.	security	Common Event Format (CEF) via AMA connector for Microsoft Sentinel	No	Yes
ConfigurationChange View changes to in-guest configuration data such as Files Software Registry Keys Windows Services and Linux Daemons	management	Enable Change Tracking and Inventory	No	Yes
ConfigurationData View the last reported state for in-guest configuration data such as Files Software Registry Keys Windows Services and Linux Daemons	management	Enable Change Tracking and Inventory	No	Yes
ContainerLog Log lines collected from stdout and stderr streams for containers.	container, applications	Container Insights	No	Yes
DnsEvents	network	Stream and filter data from Windows DNS servers with Azure Monitor Agent	No	Yes
DnsInventory	network	Stream and filter data from	No	-

Table	Categories	Data collection method	Supports basic log plan	Queries
		Windows DNS servers with Azure Monitor Agent		
Event Events from Windows Event Log on Windows computers using Azure Monitor Agent Analytics agent.	virtualmachines	Collect events with Azure Monitor Agent	No	Yes
HealthStateChangeEvent Workload Monitor Health. This data represents state transitions of a health monitor.	undefined	VM Insights	No	-
Heartbeat Records logged by Azure Monitor Agent once per minute to report on agent health.	virtualmachines, container, management	Azure Monitor Agent	No	Yes
InsightsMetrics Table that stores metrics. 'Perf' table also stores many metrics and over time they all will converge to InsightsMetrics.	virtualmachines, container, resources	VM Insights, Container Insights	No	Yes
Perf Performance counters from Windows and Linux agents that provide insight into the performance of hardware components operating systems and applications.	virtualmachines, container	Collect performance counters from VMs with Azure Monitor Agent	No	Yes
ProtectionStatus Antimalware installation info and security health status of the machine:	security	Enable Azure Monitor Agent in Defender for Cloud	No	Yes
SQLAssessmentRecommendation Recommendations generated by SQL assessments that are started through a scheduled task. When you schedule the assessment it runs by default every seven days and uploads the data into Azure Log Analytics.	workloads	SQL Server On-Demand Assessment	No	Yes
SecurityBaseline	security	Enable Azure Monitor Agent	No	-

Table	Categories	Data collection method	Supports basic log plan	Queries
		in Defender for Cloud		
SecurityBaselineSummary	security	Enable Azure Monitor Agent in Defender for Cloud	No	-
SecurityEvent Security events collected from windows machines by Azure Security Center or Azure Sentinel.	security	Windows Security Events via AMA connector for Microsoft Sentinel	No	Yes
Syslog Syslog events on Linux computers using Azure Monitor Agent.	virtualmachines, security	Collect Syslog events with Azure Monitor Agent	No	Yes
Update Details for update schedule run. Includes information such as which updates where available and which were installed.	management, security	Enable Update Management	No	Yes
UpdateRunProgress Breaks down each run of your update schedule by the patches available at the time with details on the installation status of each patch.	management	Enable Update Management	No	Yes
UpdateSummary Summary for each update schedule run. Includes information such as how many updates weren't installed.	virtualmachines	Enable Update Management	No	Yes
VMBoundPort Traffic for open server ports on the monitored machine.	virtualmachines	VM Insights	No	-
VMComputer Inventory data for servers collected by the Service Map and VM insights solutions using the Dependency agent and Azure Monitor Agent.	virtualmachines	VM Insights	No	-
VMConnection Traffic for inbound and outbound	virtualmachines	VM Insights	No	-

Table	Categories	Data collection method	Supports basic log plan	Queries
connections to and from monitored computers.				
VMProcess Process data for servers collected by the Service Map and VM insights solutions using the Dependency agent and Azure Monitor Agent.	virtualmachines	VM Insights	No	-
W3CIISLog Internet Information Server (IIS) log on Windows computers using Azure Monitor Agent.	management, virtualmachines	Collect IIS logs with Azure Monitor Agent	No	Yes
WindowsFirewall	security	Enable Azure Monitor Agent in Defender for Cloud	No	-

Activity log

The linked table lists the operations that can be recorded in the activity log for this service. These operations are a subset of all the possible resource provider operations in the activity log.

For more information on the schema of activity log entries, see Activity Log schema.

The following table lists a few example operations that relate to creating VMs in the activity log. For a complete list of operations, see Microsoft.Compute resource provider operations.

Expand table

Operation	Description
Microsoft.Compute/virtualMachines/start/action	Starts the virtual machine
Microsoft.Compute/virtualMachines/restart/action	Deletes a managed cluster
Microsoft.Compute/virtualMachines/write	Creates a new virtual machine or updates an existing one
Microsoft.Compute/virtualMachines/deallocate/action	Powers off the virtual machine and releases the compute resources
Microsoft.Compute/virtualMachines/extensions/write	Creates a new virtual machine extension or updates an existing one

Operation	Description
Microsoft.Compute/virtualMachineScaleSets/write	Starts the instances of the virtual machine scale set

Related content

- See Monitor Virtual Machines for a description of monitoring Virtual Machines.
- See Monitor Azure resources with Azure Monitor for details on monitoring Azure resources.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Connect self-managed Prometheus to Azure Monitor managed service for Prometheus

Summarize this article for me

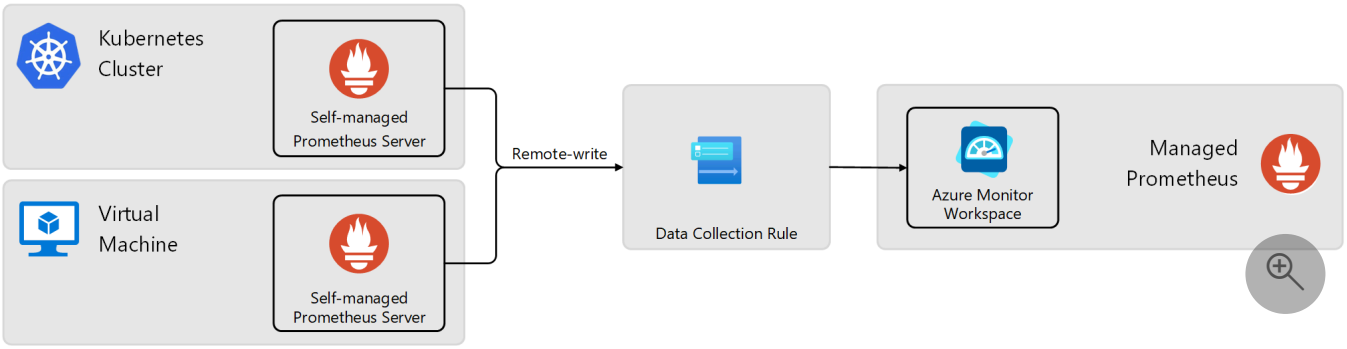
Azure Monitor managed service for Prometheus is intended to be a replacement for self-managed Prometheus so you don't need to manage a Prometheus server in your Kubernetes clusters. There may be scenarios though where you want to continue to use self-managed Prometheus in your Kubernetes clusters while also sending data to Managed Prometheus for long term data retention and to create a centralized view across your clusters. This may be a temporary solution while you migrate to Managed Prometheus or a long term solution if you have specific requirements for self-managed Prometheus.

> **Important**
>
> If you are currently using the Azure Monitor sidecar container to remote-write metrics to Azure Monitor Workspace, we recommend transitioning to a direct Prometheus configuration following the guidance in this article. This transition enables direct ingestion without requiring the sidecar component.

Architecture

Remote_write is a feature in Prometheus that allows you to send metrics from a local Prometheus instance to remote storage or to another Prometheus instance. Use this feature to send metrics from self-managed Prometheus running in your Kubernetes clusters or virtual machines to an Azure Monitor workspace used by Managed Prometheus.

The following diagram illustrates this configuration. A data collection rule (DCR) in Azure Monitor provides an endpoint for the self-managed Prometheus instance to send metrics and defines the Azure Monitor workspace where the data will be sent.



> **Important**
>
> Review key concepts about Azure Monitor managed service for Prometheus when you remote-write from self-hosted Prometheus: Migrate to Azure Monitor managed service for Prometheus.

Authentication types

The configuration requirements for remote-write depend on the authentication type used to connect to the Azure Monitor workspace. The following table describes the supported authentication types. The details for each configuration are described in the following sections.

Expand table

Type	Clusters supported
System-assigned managed identity	Azure Kubernetes service (AKS) Azure VM/VMSS
User-assigned managed identity	Azure Kubernetes service (AKS) Arc-enabled Kubernetes Azure VM/VMSS
Microsoft Entra ID	Azure Kubernetes service (AKS) Arc-enabled Kubernetes cluster Cluster running in another cloud or on-premises Azure VM/VMSS Arc-enabled servers VM running in another cloud or on-premises
Microsoft Entra workload identity	Azure Kubernetes service (AKS) Arc-enabled Kubernetes cluster

Prerequisites

Supported versions

- Prometheus version 2.45 or greater is required for user-assigned managed identity authentication.
- Prometheus version 2.48 or greater is required for Microsoft Entra application authentication.

- Prometheus version 3.50 or greater is required for system-assigned managed identity authentication.
- Prometheus version 3.60 or greater is required for Microsoft Entra workload identity authentication.

Azure Monitor workspace

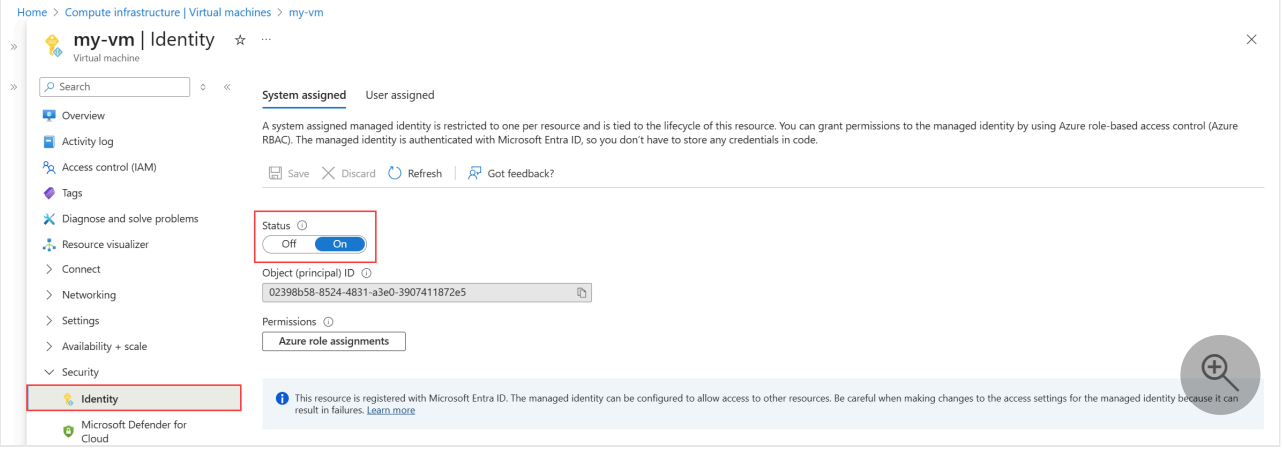
Your Azure Monitor workspace must be created before you can configure remote-write. This automatically enables Managed prometheus. If you don't already have one, see Manage an Azure Monitor workspace.

Create identity for authentication

Before you can configure remote-write, you must create the identity that you'll use to authenticate to the Azure Monitor workspace. The following sections describe how to create each type of identity if you aren't reusing an existing one.

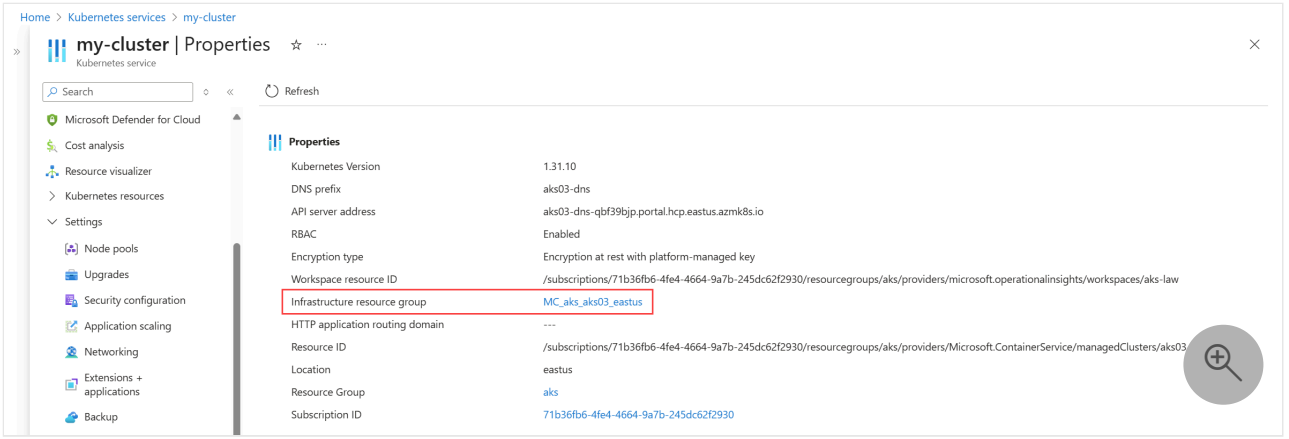
System-assigned Managed identity

You don't directly create a system-assigned managed identity but instead enable it for an Azure virtual machine or virtual machine scale set. For an Azure VM, you can enable the identity when you create the VM or enable it later from its **Identity** page in the Azure portal. For a VMSS, you must enable it after creation. For the different options to enable system-managed identity, see Configure managed identities on Azure virtual machines (VMs) and Configure managed identities for Azure resources on a virtual machine scale set.



For an AKS cluster, the managed identity must be assigned to virtual machine scale sets in the cluster. AKS creates a resource group that contains the virtual machine scale sets. Access this resource group from the **Properties** page in the cluster's menu in the Azure portal. Click on **Infrastructure Resource Group** to view list the resources in this resource

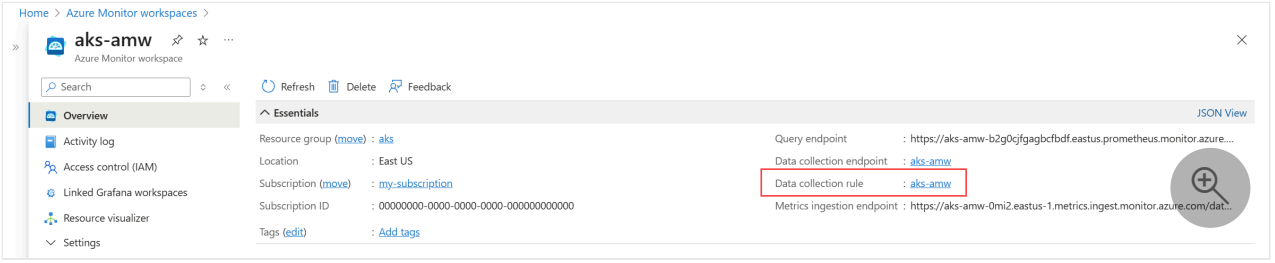
group. You must enable system managed identity for each virtual machine scale set in the resource group.



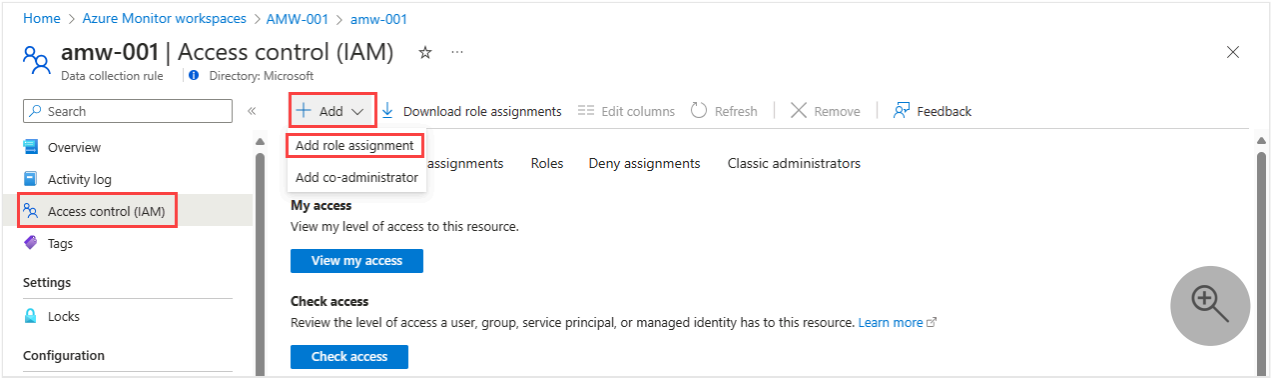
Assign roles

Once the identity that you're going to use is created, it needs to be given access to the data collection rule (DCR) associated with the Azure Monitor workspace that will receive the remote-write data. This DCR is automatically created when you create the workspace. You'll specify this identity in the remote-write configuration for the cluster or VM.

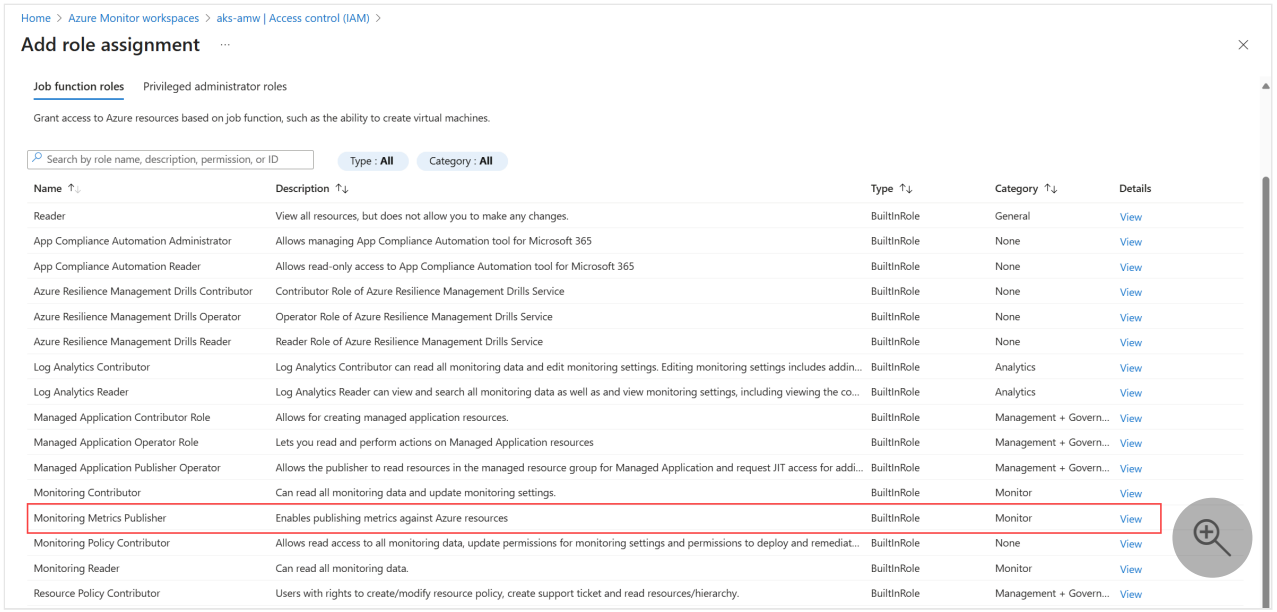
1. On the Azure Monitor workspace's overview pane, select the Data collection rule link. This opens the data collection rule (DCR) that is associated with the workspace.



2. On the page for the data collection rule, select **Access control (IAM)**. Select **Add** and then **Add role assignment**.

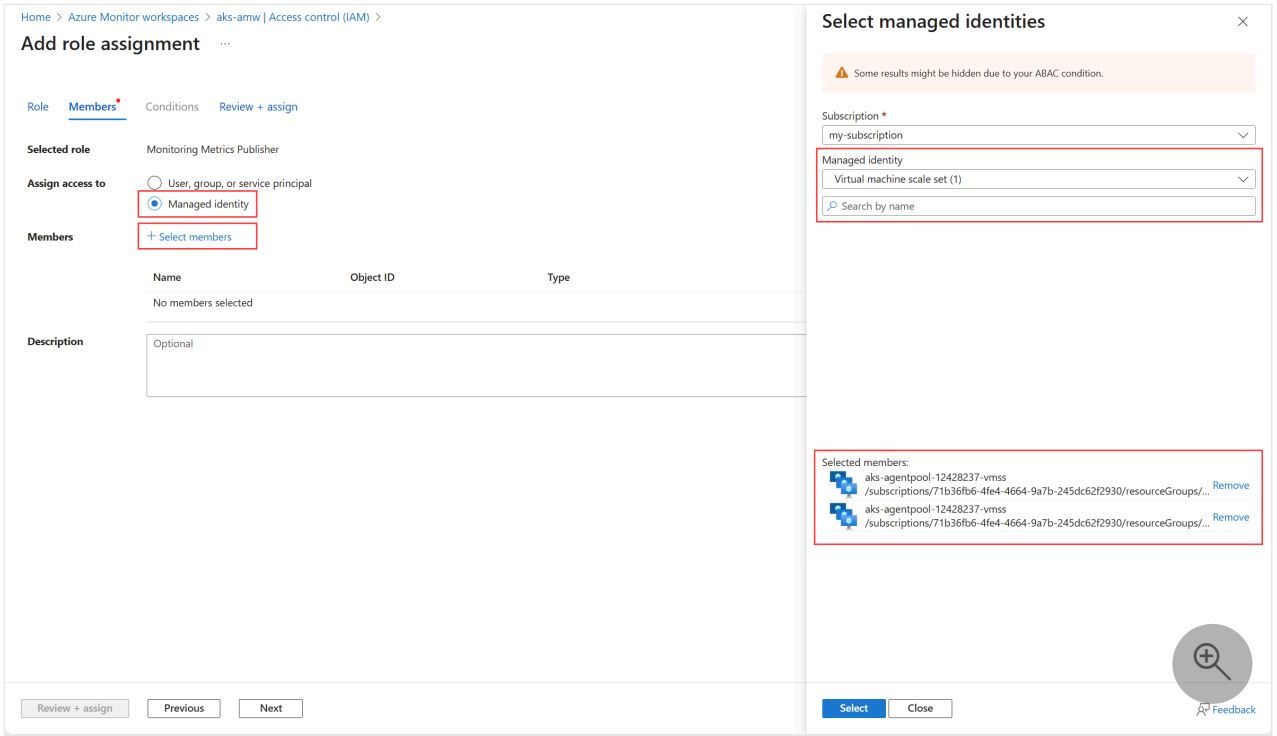


3. Select the **Monitoring Metrics Publisher** role, and then select **Next**.

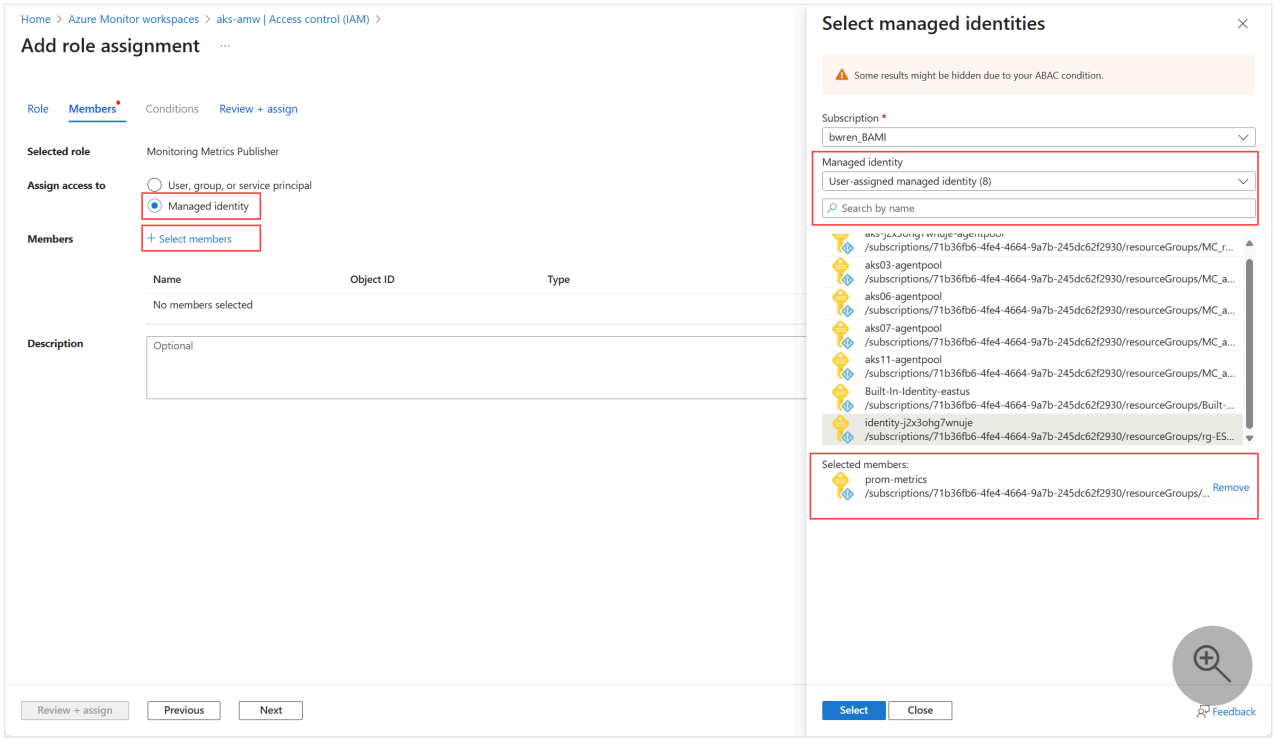


4. Select the identity to be assigned the role.

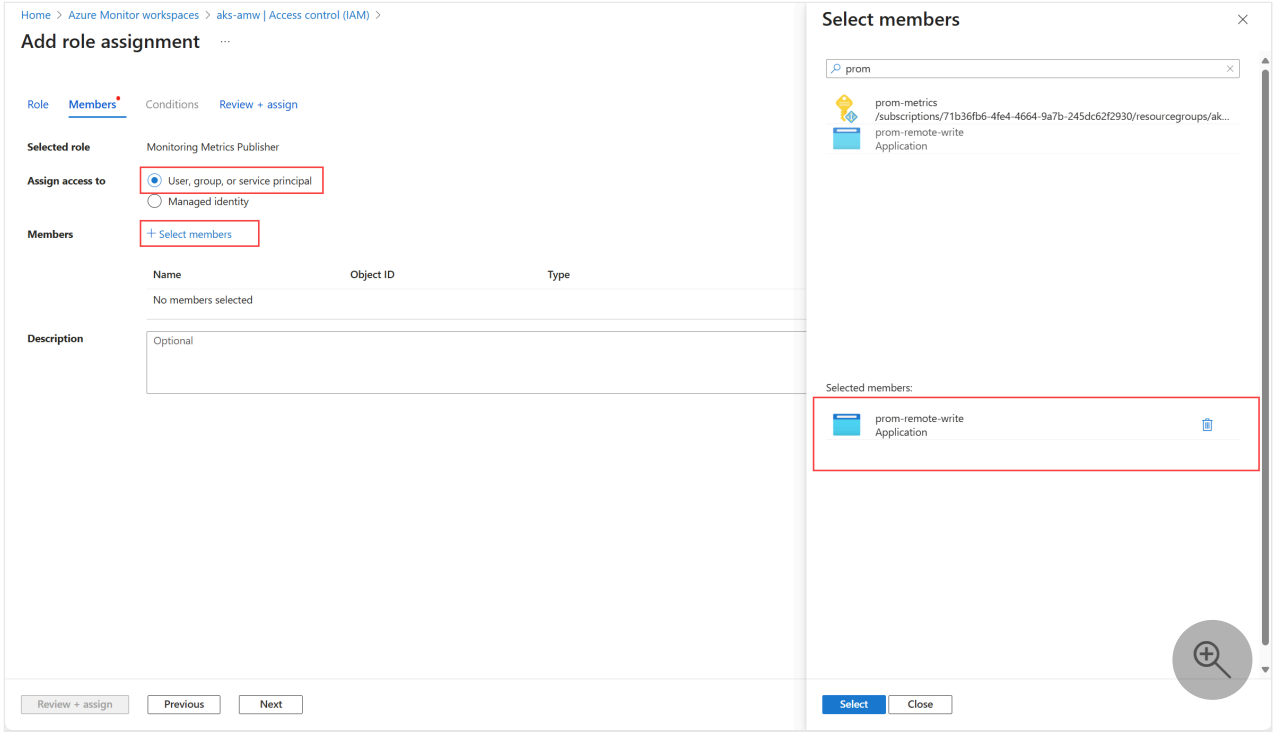
   a. For system-assigned managed identity, select **Managed identity** and then **Select members**. In the **Managed identity** dropdown, select the VM/VMSS or each of the VMSS in the AKS cluster.



   b. For user-assigned managed identity, select **User-assigned managed identity** section. Select the identity that you created.



c. For Entra ID, select **User, group, or service principal**, and then **Select members**. Select the application that you created, and then choose **Select**.

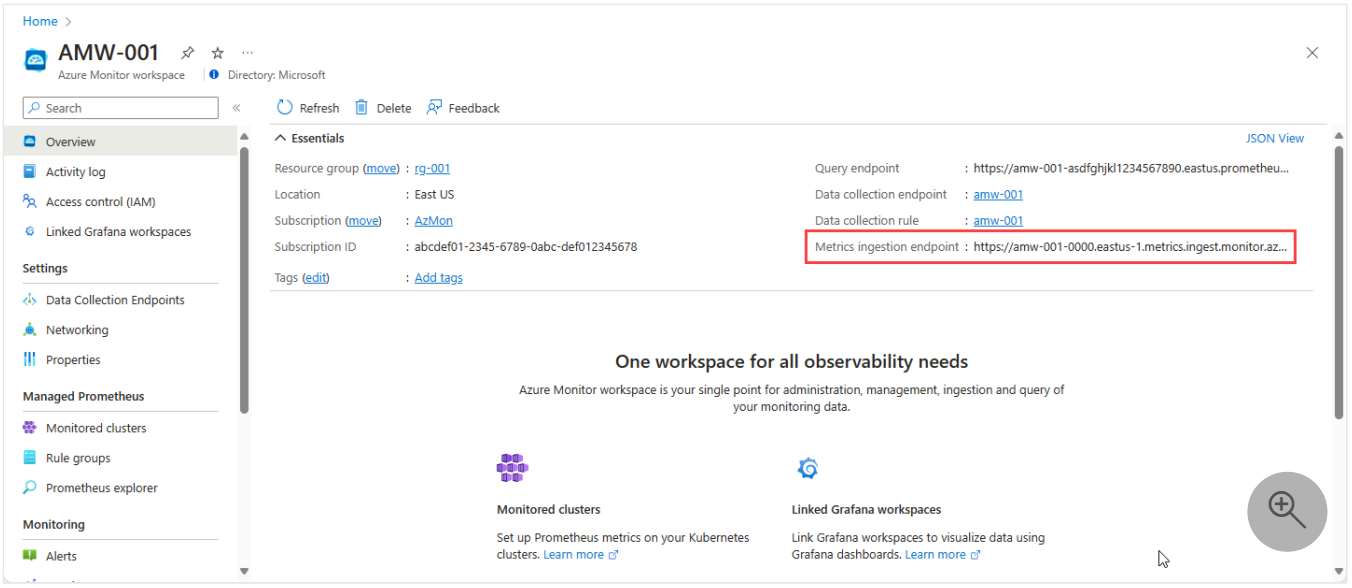


5. Select **Select** to confirm your selection and then **Review + assign** to complete the role assignment.

Configure remote-write in configuration file

Remote write is configured in the Prometheus configuration file `prometheus.yml` or in Prometheus Operator. The final step is to add remote write to the configuration file for your

self-managed Prometheus server. In addition to details for the identity that you created, you'll also need the metrics ingestion endpoint (URL) for the Azure Monitor workspace. Get this value from the **Overview** page for your Azure Monitor workspace in the Azure portal, and make a note of it.



Configure remote write with Prometheus config file

To send data to your Azure Monitor workspace, add the following section to the configuration file (`prometheus.yml`) of your self-managed Prometheus instance.

> **Note**
>
> For system-assigned managed identity, leave the client ID field blank (client_id: "").

Managed identity

Azure CLI

```
prometheus:
  prometheusSpec:
    remote_write:
    - url: "<metrics ingestion endpoint for your Azure Monitor workspace>"
      azuread:
        cloud: 'AzurePublic'  # Options are 'AzurePublic', 'AzureChina', or
'AzureGovernment'.
        managed_identity:
          client_id: "<client-id of the managed identity>"
```

Entra ID

Azure CLI

```
prometheus:
  prometheusSpec:
    remote_write:
    - url: "<metrics ingestion endpoint for your Azure Monitor workspace>"
      azuread:
        cloud: 'AzurePublic'  # Options are 'AzurePublic', 'AzureChina', or
'AzureGovernment'.
        oauth:
          client_id: "<client-id from the Entra app>"
          client_secret: "<client secret from the Entra app>"
          tenant_id: "<Azure subscription tenant Id>"
```

Workload identity

Azure CLI

```
prometheus:
  prometheusSpec:
    podMetadata:
      labels:
        azure.workload.identity/use: "true"
    remote_write:
    - url: "<metrics ingestion endpoint for your Azure Monitor workspace>"
      azuread:
        cloud: 'AzurePublic'  # Options are 'AzurePublic', 'AzureChina', or
'AzureGovernment'.
        workload_identity:
          client_id: "<client-id from the Entra app>"
          tenant_id: "<Azure subscription tenant Id>"
```

Apply the configuration file updates:

Virtual Machine

For a virtual machine, the configuration file will be `prometheus.yml` unless you specify a different one using `prometheus --config.file <path-to-config-file>` when starting the Prometheus server.

Kubernetes cluster

For a Kubernetes cluster, the configuration file is typically stored in a ConfigMap. Following is a sample ConfigMap that includes a remote-write configuration using managed identity for self-managed Prometheus running in a Kubernetes cluster.

yml

```
  GNU nano 6.4
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-server-conf # must match what your pod mounts
  namespace: monitoring  # adjust to your namespace
data:
  prometheus.yml: |-
   global:
     scrape_interval: 15s
     evaluation_interval: 15s
     external_labels:
       cluster: "aks11"

   scrape_configs:
     - job_name: "prometheus"
       static_configs:
         - targets: ["localhost:9090"]

   remote_write:
     - url: "https://aks-amw-0mi2.eastus-
1.metrics.ingest.monitor.azure.com/dataCollectionRules/dcr-
00000000000000000000000000000000/streams/Microsoft-
PrometheusMetrics/api/v1/write?api-version=2023-04-24"
       azuread:
         cloud: 'AzurePublic'
         managed_identity:
           client_id: "00001111-aaaa-2222-bbbb-3333cccc4444"
```

Use the following command to apply the configuration file updates.

Bash

```
kubectl apply -f <configmap-file-name>.yaml
```

Restart Prometheus to pick up the new configuration. If you are using a deployment, you can restart the pods by running the following command.

Bash

```
kubectl -n monitoring rollout restart deploy <prometheus-deployment-name>
```

> ⓘ **Note**
>
> After you edit the configuration file, restart Prometheus to apply the changes.

For information on tuning the remote write configuration, see Remote write tuning.

Troubleshoot

HTTP 403 error in the Prometheus log

It takes about 30 minutes for the assignment of the role to take effect. During this time, you may see an HTTP 403 error in the Prometheus log. Check that you have configured the managed identity or Microsoft Entra ID application correctly with the **Monitoring Metrics Publisher** role on the workspace's DCR. If the configuration is correct, wait 30 minutes for the role assignment to take effect.

No Kubernetes data is being collected

If data isn't being collected in Managed Prometheus, run the following command to find errors in the remote write container.

Azure CLI

```
kubectl --namespace <Namespace> describe pod <Prometheus-Pod-Name>
```

Container restarts repeatedly

A container regularly restarting is likely due to misconfiguration of the container. Run the following command to view the configuration values set for the container. Verify the configuration values especially `AZURE_CLIENT_ID` and `IDENTITY_TYPE`.

azureccli

```
kubectl get pod <Prometheus-Pod-Name> -o json | jq -c  '.spec.containers[] |
select( .name | contains("<Azure-Monitor-Side-Car-Container-Name>"))'
```

The output from this command has the following format:

```
{"env":[{"name":"INGESTION_URL","value":"https://my-azure-monitor-
workspace.eastus2-1.metrics.ingest.monitor.azure.com/dataCollectionRules/dcr-
00000000000000000/streams/Microsoft-PrometheusMetrics/api/v1/write?api-
version=2021-11-01-preview"},{"name":"LISTENING_PORT","value":"8081"},
{"name":"IDENTITY_TYPE","value":"userAssigned"},
{"name":"AZURE_CLIENT_ID","value":"00000000-0000-0000-0000-
00000000000"}],"image":"mcr.microsoft.com/azuremonitor/prometheus/promdev/prom-
remotewrite:prom-remotewrite-20221012.2","imagePullPolicy":"Always","name":"prom-
remotewrite","ports":[{"containerPort":8081,"name":"rw-
port","protocol":"TCP"}],"resources":{},"terminationMessagePath":"/dev/termination-
```

```
log","terminationMessagePolicy":"File","volumeMounts":
[{"mountPath":"/var/run/secrets/kubernetes.io/serviceaccount","name":"kube-api-
access-vbr9d","readOnly":true}]}
```


Data dropped due in high volume environments

The Data Collection Rule (DCR) and Data Collection Endpoint (DCE) for the Azure Monitor workspace are subject ingestion limits listed in Azure Monitor service limits. You're most subject to these limits when configuring remote write for multiple clusters sending data to the same endpoint.

Consider Remote write tuning to adjust configuration settings for better performance. If you still see data drops, consider creating additional DCRs and DCEs to distribute the ingestion load across multiple endpoints. This approach helps optimize performance and ensures efficient data handling. See Instructions on how to create custom Data collection endpoint(DCE) and custom Data collection rule(DCR) for an existing Azure monitor workspace(AMW) to ingest Prometheus metrics.

Next steps

- Learn more about Azure Monitor managed service for Prometheus.
- Collect Prometheus metrics from an AKS cluster

Last updated on 01/28/2026

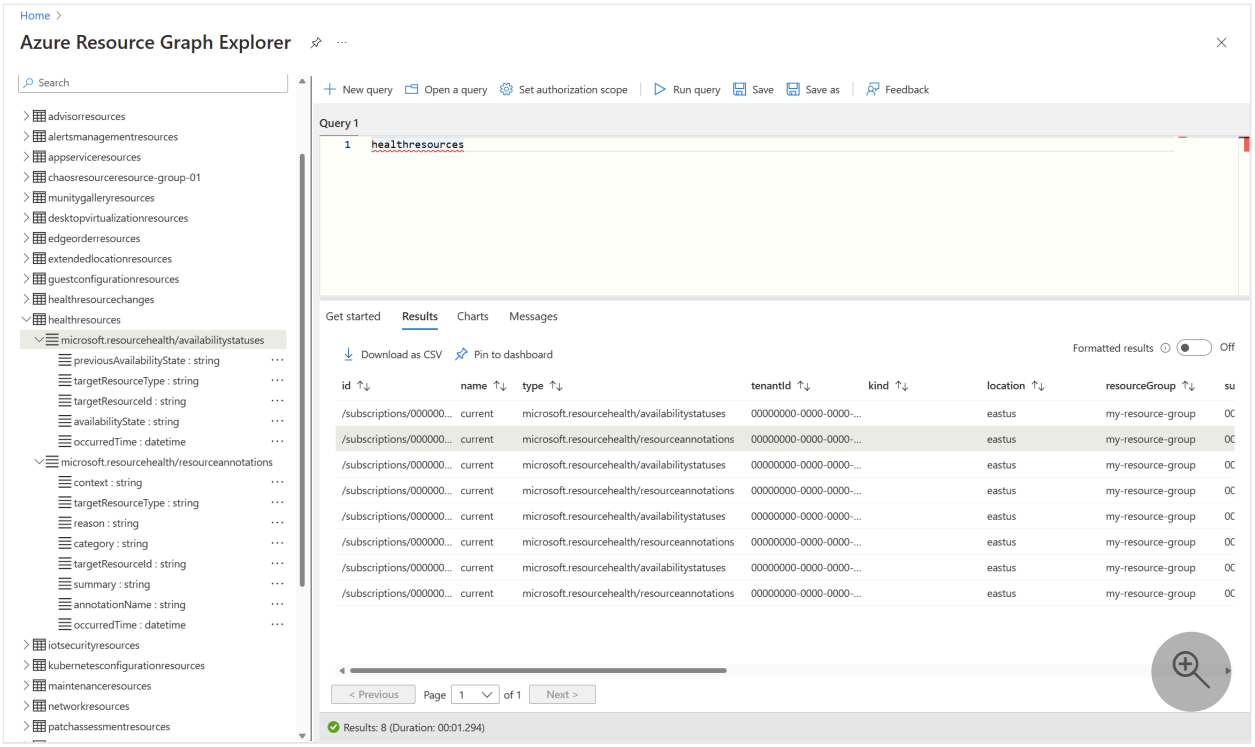
VM availability information in Azure Resource Graph

Article • 08/22/2024

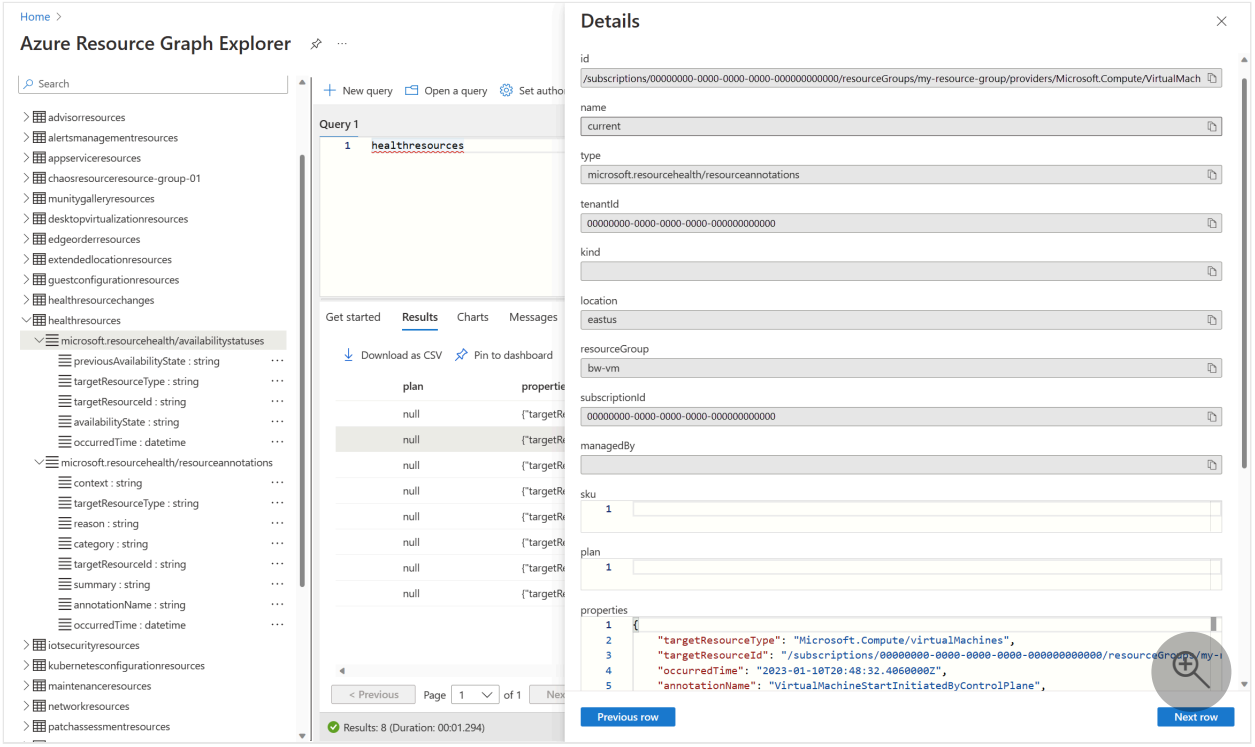
Azure Resource Graph is an Azure service that allows you to use the same KQL query language used in log queries to query your Azure resources at scale with complex filtering, grouping, and sorting by resource properties. You can use VM health annotations to Azure Resource Graph (ARG) for detailed failure attribution and downtime analysis including the following:

- Query the latest snapshot of VM availability together across all your Azure subscriptions.
- Assess the impact to business SLAs and trigger decisive mitigation actions, in response to disruptions and type of failure signature.
- Set up custom dashboards to supervise the comprehensive health of applications by joining VM availability information with additional resource metadata in Resource Graph.
- Track relevant changes in VM availability across a rolling 14 days window, by using the change tracking mechanism for conducting detailed investigations.

To get started with Resource Graph, open **Resource Graph Explorer** in the Azure portal. Select the **Table** tab and have a look at the microsoft.resourcehealth/availabilitystatuses and microsoft.resourcehealth/resourceannotations tables which are described below. Click on **healthresources** to create a simple query and then click **Run** to return the records.



To view the details for a record, scroll to the right and select **See details**.



There will be two types of events populated in the HealthResources table:

microsoft.resourcehealth/availabilitystatuses

This event denotes the latest availability status of a VM, based on the health checks performed by the underlying Azure platform. The availability states currently emitted for VMs are as follows:

- **Available**: The VM is up and running as expected.
- **Unavailable**: A disruption to the normal functioning of the VM has been detected.
- **Unknown**: The platform is unable to accurately detect the health of the VM. Check back in a few minutes.

The availability state is in the `properties` field of the record which includes the following properties:

Expand table

Field	Description
targetResourceType	Type of resource for which health data is flowing
targetResourceId	Resource ID
occurredTime	Timestamp when the latest availability state is emitted by the platform
previousAvailabilityState	Previous availability state of the VM
availabilityState	Current availability state of the VM

A sample `properties` value looks similar to the following:

JSON

```
{
    "targetResourceType": "Microsoft.Compute/virtualMachines",
    "targetResourceId":
"/subscriptions/<subscriptionId>/resourceGroups/<ResourceGroupName>/providers/Microsoft.Compute/virtualMachines/<VMName>",
    "occurredTime": "2022-10-11T11:13:59.9570000Z",
    "previousAvailabilityState": "Available",
    "availabilityState": "Unavailable"
}
```

microsoft.resourcehealth/resourceannotations

This event contextualizes any changes to VM availability, by detailing necessary failure attributes to help you investigate and mitigate the disruption as needed. The full list of VM health annotations are listed at [Resource Health virtual machine Health Annotations] (../service-health/resource-health-vm-annotation.md).

These annotations can be broadly classified into the following:

- **Downtime Annotations**: Emitted when the platform detects VM availability transitioning to Unavailable. Examples include host crashes or reboot operations.
- **Informational Annotations**: Emitted during control plane activities with no impact to VM availability. Examples include VM allocation, stop, delete, start. Usually, no additional customer action is required in response.
- **Degraded Annotations**: Emitted when VM availability is detected to be at risk. Examples include when failure prediction models predict a degraded hardware component that can cause the VM to reboot at any given time. You should redeploy by the deadline specified in the annotation message to avoid any unanticipated loss of data or downtime.

Expand table

Field	Description
targetResourceType	Type of resource for which health data is flowing
targetResourceId	Resource ID
occurredTime	Timestamp when the latest availability state is emitted by the platform
annotationName	Name of the Annotation emitted
reason	Brief overview of the availability impact observed by the customer
category	Denotes whether the platform activity triggering the annotation was either planned maintenance or unplanned repair. This field is not applicable to customer/VM-initiated events. Possible values: Planned | Unplanned | Not Applicable | Null
context	Denotes whether the activity triggering the annotation was due to an authorized user or process (customer initiated), due to the Azure platform (platform initiated), or due to activity in the guest OS that has resulted in availability impact (VM initiated). Possible values: Platform-Initiated | User-initiated | VM-initiated | Not Applicable | Null
summary	Statement detailing the cause for annotation emission, along with remediation steps that can be taken by users

See Azure Resource Graph sample queries by table for sample queries using this data.

Next steps

- Create alerts from collected data

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

VM watch: Enhancing VM health monitoring (preview)

Article • 04/29/2025

VM watch is a standardized, lightweight, and adaptable service offering for virtual machines (VMs) and virtual machine scale sets. It runs health checks within a VM at configurable intervals and sends the results via a uniform data model to Azure. The AI operations (AIOps) engines for production monitoring in Azure consume these health results for regression detection and prevention.

VM watch is delivered via the Application Health VM extension to provide ease of deployment and manageability for customers. In addition, VM watch is offered at no extra cost.

Monitoring specifics for VM watch

- **Ease of adoption**: VM watch is available through the Application Health VM extension.
- **Flexible deployment**: You can enable VM watch by using an Azure Resource Manager template (ARM template), PowerShell, or the Azure CLI.
- **Compatibility**: VM watch operates seamlessly in both Linux and Windows environments. It's suitable for individual VMs and virtual machine scale sets alike.
- **Resource governance**: VM watch provides efficient monitoring without affecting system performance. Resource caps on the CPU and memory utilization of the VM watch process help protect VMs.
- **Out-of-the-box readiness**: VM watch comes equipped with a suite of default tests that you can configure for your scenarios.

VM watch Memory Constraints

To maintain consistent performance across different virtual machine (VM) configurations, **VM watch** enforces memory usage limits according to the VM SKU's total available memory. The memory caps are adjusted dynamically based on the VM's memory tier.

Expand table

VM Memory Range	VM watch Memory Cap
Less than 8 GB	80MB
8 GB to 16 GB	200MB
Greater than 16 GB	400MB

Network

Expand table

Signal name	Type	Description
Outbound connectivity	Check	Verify the network outbound connectivity from the Azure VM.
DNS Resolution	Check	Verify if one or more DNS names can be resolved.
TCPSynRetransmits (Linux Only)	Metric	The number of times the system retransmits a TCP SYN and SYN/ACK packet before giving up on establishing a connection.
SegmentsRetransmitted	Metric	The number of transmitted TCP segments that contain one or more previously transmitted octets.
NormalizedSegmentsRetransmitted	Metric	**SegmentsRetransmitted** / (**SegmentsSent** + **SegmentsRetransmitted**)
ConnectionResets	Metric	The number of times that TCP connections made a direct transition to the `CLOSED` state from either the `ESTABLISHED` state or the `CLOSE_WAIT` state.
NormalizedConnectionResets	Metric	The percentage of connections that were reset during the last measurement interval.
FailedConnectionAttempts	Metric	The number of times that TCP connections made a direct transition to the `CLOSED` state from either the `SYN_SENT` state or the `SYN_RCVD` state.
NormalizedFailedConnectionAttempts	Metric	**FailedConnectionAttempts** / (**ActiveConnectionOpenings** + **PassiveConnectionOpenings**)
ActiveConnectionOpenings	Metric	The number of times that TCP connections made a direct transition to the `SYN_SENT` state from the `CLOSED` state.
PassiveConnectionOpenings	Metric	The number of times that TCP connections made a direct transition to the `SYN_RCVD` state from the `LISTEN` state.
CurrentConnections	Metric	The number of connections established.
SegmentsReceived	Metric	The number of segments received, including segments received in error.
SegmentsSent	Metric	The number of segments sent, including segments on current connections but excluding segments that

Signal name	Type	Description
		contain only retransmitted octets.

Disk

Expand table

Signal name	Type	Description
Azure Disk I/O	Check	Verify file creation, write, and read. Delete operations on each drive mounted to the VM.
FreeSpaceInBytes	Metric	The free disk space of the target mount point.
UsedSpaceInBytes	Metric	The used disk space of the target mount point.
CapacityInBytes	Metric	The disk space capacity of the target mount point.
UsedPercent	Metric	The percentage of used disk space for the target mount point.
WriteOps	Metric	The write operations per second for the target disk/partition.
ReadOps	Metric	The read operations per second for the target disk/partition.

CPU

Expand table

Signal name	Type	Description
ProcessCPUCoreUsage	Metric	An instantaneous measurement of the percentage of a single CPU core that the target process is using (100 = 100%, a whole core).
ProcessCPUMachineUsage	Metric	The percentage of the machine's total CPU that this process is using.
MachineTotalCpuUsage	Metric	The VM's total instantaneous CPU utilization.

Memory

Expand table

Signal name	Type	Description
ProcessRSSPercent	Metric	**Process RSS** / (**Machine Total Memory** * **100%**)

Signal name	Type	Description
ProcessPageFaults	Metric	The number of page faults since the process started.
MachineMemoryTotalInBytes	Metric	The VM's total Memory in Bytes.
MachineMemoryUsedPercent	Metric	**Machine Used Memory** / (**Machine Total Memory** * 100%)
TotalPageFaults	Metric	The total number of page faults for all running processes since they started.

Process

Expand table

Signal name	Type	Description
Process Creation	Check	Start a lightweight process to validate that process creation is possible.
Running Process(es)	Check	Verify if the target process or processes are running.
UpTime	Metric	How long the target process has been up and running since the last process startup.

IMDS

Expand table

Signal name	Type	Description
IMDS	Check	Verify that the user can reach an Azure Instance Metadata Service (IMDS) endpoint from within the VM. VM information is returned from the IMDS endpoint query.

Clock

Expand table

Signal name	Type	Description
Clock Skew	Check	Verify the clock skew between the remote Network Time Protocol (NTP) server and the Azure VM. For a Windows VM, fall back to check if the Windows Time service is synced with `w32tm` if the remote NTP server is inaccessible.

OS

Expand table

Signal name	Type	Description
System Errors	Metric	Collect the number of errors from the system-level event log (Windows only) where the SystemData <=2 (including LOG_ALWAYS, Critical, Error). The measurementTarget is defined as the Source_EventId of the EventLog using default Windows locale. Each collection is limited to more than 10 different measurement targets.

azblob

Expand table

Signal name	Type	Description
Azure Storage blob connectivity	Check	Verify the connectivity to the Azure Storage blob and download the blob by using MSI or a shared access signature (SAS) token.

Hardware

Expand table

Signal name	Type	Description
Hardware Health Monitor	EventLog	Collect hardware health info from the Windows event log. Currently, only disk-related critical events are collected, including events with ID 7, 500, 504, 505, 512, and 549.
Hardware Health Nvidia Smi	EventLog	Collect GPU stats including memory and GPU usage, temp and others by running nvidia-smi command (Linux Ubuntu only)

Related content

- VM watch Collectors Suite
- Install VM watch
- Configure VM watch
- Configure Event Hubs for VM watch

Install VM watch (preview)

Article • 01/13/2025

You can enable VM watch by using an Azure Resource Manager template (ARM template), PowerShell, or the Azure CLI on Azure virtual machines (VMs) and Azure virtual machine scale sets. You can enable VM watch on both Linux and Windows virtual machines. VM watch is delivered through the Application Health VM extension for ease of adoption.

The code in this article details the steps to install the Application Health VM extension and enable VM watch. Note that the code segments require user input. Any labels within angle brackets (`<>`) in the code need to be replaced with values that are specific to your installation. Here's a list of parameters with instructions on what to replace them with.

Expand table

Parameter	Description
`<your subscription id>`	The Azure subscription ID in which you want to install VM watch.
`<your vm name>`	The name of the virtual machine to which the extension is being installed.
`<your resource group name>`	The name of the resource group in your Azure subscription that your VM will be assigned to.
`<your location>`	The Azure region in which your VM is installed.
`<your extension name`	The name that will be assigned to the Application Health VM extension that you're installing.
`<application health extension type>`	Specifies whether the Windows or Linux Application Health extension will be installed.
`<your vm scale set name>`	The name of the virtual machine scale set in which you want to install VM watch.

Prerequisites

1. Register the feature

Register for adopting VM watch by running the following commands via the Azure CLI:

```
```

```
az feature register --name VMWatchPreview --namespace Microsoft.Compute --
subscription <your subscription id>
az provider register --namespace Microsoft.Compute --subscription <your
subscription id>
```

Validate feature registration

Validate that you successfully registered for the VM watch feature by running the following command:

```
az feature show --namespace Microsoft.Compute --name VMWatchPreview --
subscription <your subscription id>
```

2. Ensure that a VM is installed

For information on how to create a VM and/or virtual machine scale set, see the quickstart guide for Windows and the quickstart guide for Linux.

> **ⓘ Important**
>
> If the Application Health extension is already installed on the VM, ensure that the settings `autoUpgradeMinorVersion` and `enableAutomaticUpgrade` are set to `true`.

Install VM watch on an Azure virtual machine

> **ⓘ Important**
>
> The code segment is identical for both Windows and Linux, except for the value of the parameter `<application health extension type>` passed in to the extension type. Replace `<application health extension type>` with `"ApplicationHealthLinux"` for Linux installations and `"ApplicationHealthWindows"` for Windows installations.

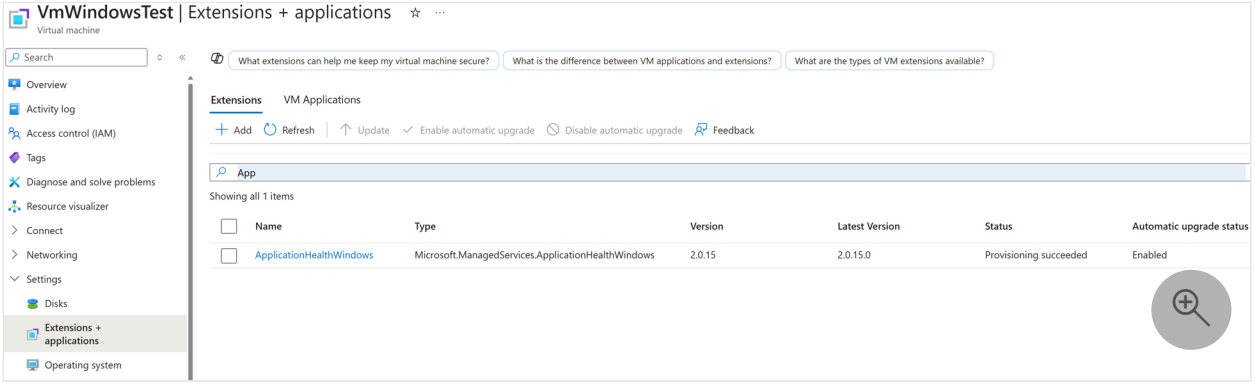
CLI

```
az vm extension set --resource-group <your resource group> --vm-name
<your vm name> --name <application health extension type> --publisher
Microsoft.ManagedServices --version 2.0 --settings '{"vmWatchSettings":
{"enabled": true}}' --enable-auto-upgrade true
```

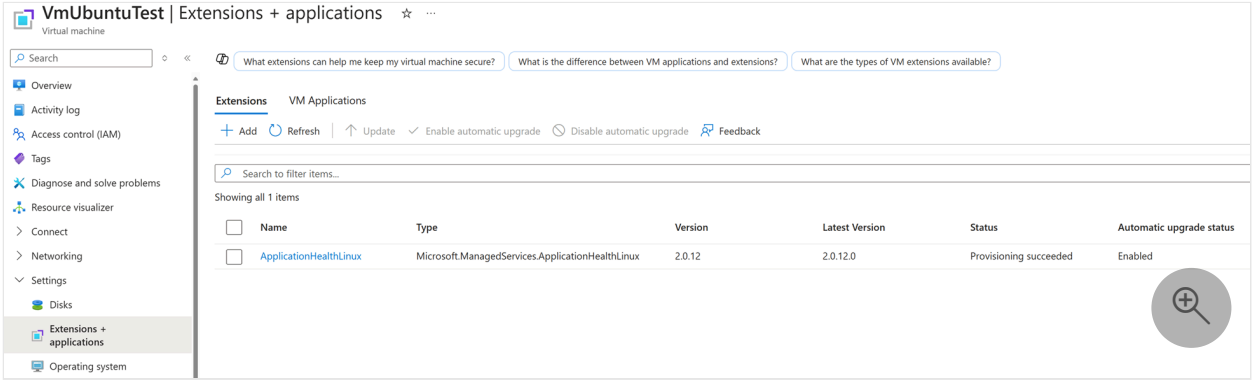
Validate that the Application Health VM extension is installed on the Azure VM

Go to the Azure portal and confirm that the Application Health VM extension was successfully installed.

The following screenshot shows a Windows installation.



The following screenshot shows a Linux installation.



To confirm that VM watch was enabled on this VM, go back to the overview page and select the JSON view for the VM. Ensure that the configuration exists in the JSON.

```
"settings": {
    "vmWatchSettings": {
        "enabled": true
```

```
    }
}
```

Install VM watch on an Azure virtual machine scale set

> **Important**
>
> The code segment is identical for both Windows and Linux, except for the value of the parameter `<application health extension type>` passed in to the extension type. Replace `<application health extension type>` with `"ApplicationHealthLinux"` for Linux installations and `"ApplicationHealthWindows"` for Windows installations.

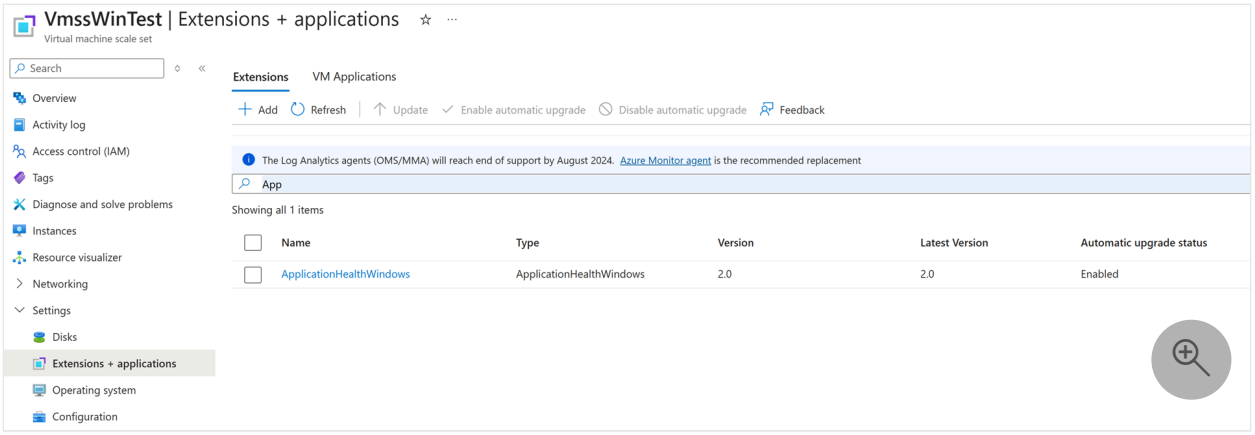
CLI

```
az vmss extension set --resource-group '<your resource group name>' --
vmss-name '<your vm scale set name>' --name <application health
extension type> --publisher Microsoft.ManagedServices --version 2.0 --
settings '{"vmWatchSettings": {"enabled": true}}' --enable-auto-upgrade
true
```

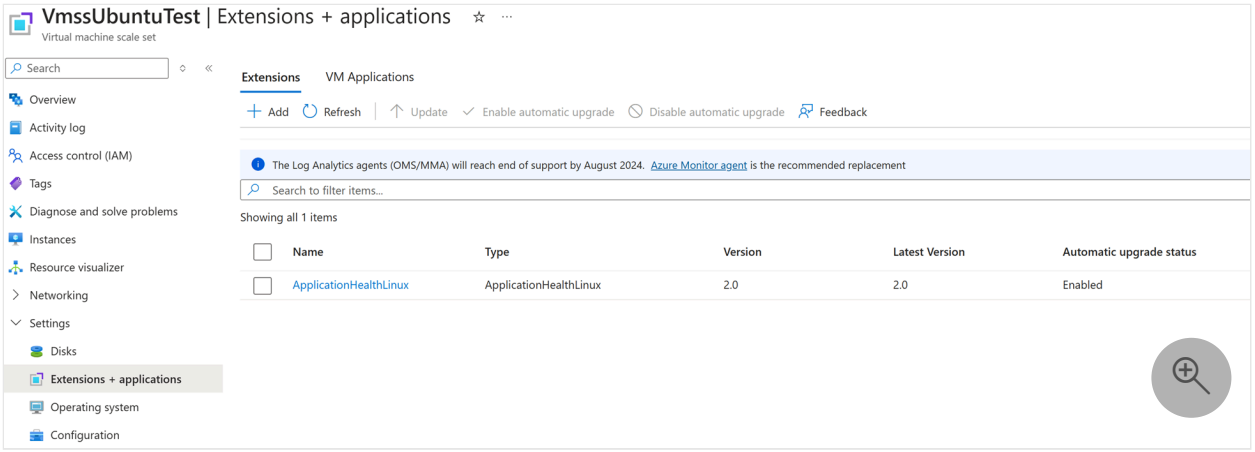
Validate that the Application Health VM extension is installed in the virtual machine scale set

Go to the Azure portal and confirm that the Application Health VM extension was successfully installed.

The following screenshot shows a Windows installation.



The following screenshot shows a Linux installation.



To confirm that VM watch was enabled on this scale set, go back to the overview page and select the JSON view for the scale set. Ensure that the configuration exists in the JSON.

```
"settings": {
    "vmWatchSettings": {
        "enabled": true
    }
}
```

Related content

- Application Health extension
- Azure CLI
- Azure portal

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

VM watch Collectors Suite

Article • 03/03/2025

VM watch collectors are designed to gather VM health data on various resources like disk and network, by running health checks within the VM. This suite of collectors aid in identifying issues, monitoring performance trends, and optimizing resources to enhance the overall user experience.

This article provides a summary of all available collectors in VM watch, along with the corresponding checks, metrics, logs, and parameter configurations. For detailed descriptions of each check, metric, and log, refer to the VM watch overview page.

Prerequisites

This article assumes that you're familiar with:

- VM watch checks, metrics, and logs
- Install VM watch to virtual machines and scale sets

Note

Expand table

Name	Description
Collector	Logical grouping of similar tests where you can collect checks, metrics, and logs to determine the health of a particular resource
Signals	What is emitted to reflect the health status of VMs. The three types of signals emitted are checks, metrics, and logs
Group	Indicates whether the collectors are part of the core or optional group. Core group collectors are enabled by default, while optional group collectors can be enabled or disabled based on your requirements
Tags	Used to categorize and filter checks, metrics, and logs
Eligibility	Determines whether a collector is eligible to be executed based on the environment attributes you specify
Default Behavior	Standard setting and action that would be followed if no custom configurations are provided.
Overwritable Parameters	Associated parameters that can be customized to override the default configuration

Groups, tags and corresponding checks, metrics, and event logs

Expand table

Collector Name	Group	Tags	Checks	Metrics	Event Logs
outbound_connectivity	Core	Network	• outbound_connectivity		
dns	Core	Network	• dns		
tcp_stats	Core	Network		• SegmentsRetransmitted • TCPSynRetransmits (Linux only) • NormalizedSegmentsRetransmitted • ConnectionResets • NormalizedConnectionResets • FailedConnectionAttempts • NormalizedFailedConnectionAttempts • ActiveConnectionOpenings • PassiveConnectionOpenings • CurrentConnections • SegmentsReceived • SegmentsSent	
clock_skew	Core	Clock	• clockskew		

Collector Name	Group	Tags	Checks	Metrics	Event Logs
disk_io	Core	Disk	• disk_io	• UsedSpaceInBytes • FreeSpaceInBytes • CapacityInBytes • UsedPercent	
disk_iops	Core	Disk		• WriteOps • ReadOps	
imds	Core	IMDS	• imds		
process	Core	Process	• process		
process_memory	Core	Process		• ProcessRSSPercent • ProcessPageFaults • MachineMemoryTotalInBytes • MachineMemoryUsedPercent • TotalPageFaults	
process_cpu	Core	Process		• ProcessCPUCoreUsage • ProcessCPUMachineUsage • MachineTotalCpuUsage	
process_monitor	Optional	Process	• process_monitor	• UpTime	
system_error	Core	OS		• SystemErrors	
az_storage_blob	Optional	AzBlob	• az_storage_blob		
hardware_health_monitor	Optional	Hardware			• hardware_health_monitor
hardware_health_nvidia_smi	Optional	Hardware			• hardware_health_nvidia_sm

Eligibility, default behavior, and overwritable parameters

Expand table

Collector Name	Eligibility	Default Behavior	Overwritable Parameters
outbound_connectivity	Eligible if EnvironmentAttribute "OutboundConnectivityDisabled" isn't set or set to "false"	This collector is executed every 60s. In each execution, it sends an http GET request to `http://www.msftconnecttest.com/connecttest.txt` with a time-out of 5s. If the request fails, it retries at most two more times with and interval of 10s. The verification is marked as "Failed" if all the retries fail.	• OUTBOUND_CONNECTIVITY_INTERVAL interval of the Collector. Default: 60s • OUTBOUND_CONNECTIVITY_URLS: the Collector sends http GET requests to. U string using `,` as separator. Default: `http://www.msftconnecttest.com/connec` • OUTBOUND_CONNECTIVITY_TIMEOUT_ the http GET request time-out in millise • OUTBOUND_CONNECTIVITY_TOTAL_AT number of attempts to send http reque fails. Default: 3 • OUTBOUND_CONNECTIVITY_RETRY_IN the retry interval in seconds if the previ Default: 10
dns	Eligible if EnvironmentAttribute "OutboundConnectivityDisabled" isn't set or set to "false"	This Collector is executed every 180s. In each execution, it tries to resolve the DNS name `www.msftconnecttest.com` . The verification is marked as "Failed" if the DNS name can't be resolved.	• DNS_INTERVAL: the execution interval c Default: 180s • DNS_NAMES: the domain names to be by `,`. Default: `www.msftconnecttest.com`
tcp_stats	Always eligible	This collector is executed every 180s. In each execution, it collects the TCP statistics of the last 180s.	• TCP_STATS_INTERVAL: the execution int Default: 180s

Collector Name	Eligibility	Default Behavior	Overwritable Parameters
clock_skew	Eligible if EnvironmentAttribute "OutboundConnectivityDisabled" isn't set or set to "false"	This collector is executed every 180s. In each execution, it retrieves the clock offset between the remote NTP server `time.windows.com` and the VM. The verification is marked as "Failed" if the clock skew is larger than 5.0 seconds. In Windows VM, if connecting to remote NTP server fails, it fallbacks to check Windows Time Service with w32tm command. The verification is marked as "Failed" if the w32tm command returns "Leap Indicator: 3(not synchronized)".	• CLOCK_SKEW_INTERVAL: the execution Collector. Default: 180s • CLOCK_SKEW_NTP_SERVER: the remote calculate clock skew. Default: time.wind • CLOCK_SKEW_TIME_SKEW_THRESHOLD threshold in seconds of clock offset to r as "Failed". Default: 5.0
disk_io	Always eligible if mount points aren't specified. If mount points are explicitly specified, only eligible when data disks are attached to the VM	This collector is executed every 180s. In each execution, it verifies the disk io availability in each available mount point by creating a folder, creating a file, writing bytes to it, deleting it and delete the folder. Then it collects the disk usage info including used space, free space, total capacity and used percentage from each mount point.	• DISK_IO_INTERVAL: the execution interv Default: 180s • DISK_IO_MOUNT_POINTS: the mount p No default value • DISK_IO_IGNORE_FS_LIST: the file syster ignored separated by `,`. Default: tmpfs,devtmpfs,devfs,iso9660,overlay,au • DISK_IO_FILENAME: the name of the file file read/write. Default: vmwatch-{times
disk_iops	Always eligible	This collector is executed every 180s. In each execution, it collects the disk read and write operations per second metrics from each available disk device.	• DISK_IOPS_INTERVAL: the execution int Default: 180s • DISK_IOPS_DEVICES: the device names default value • DISK_IOPS_IGNORE_DEVICE_REGEX: the name that should be ignored. Default: l
imds	Always eligible	This collector is executed every 180s. In each execution, it queries the IMDS endpoint `http://169.254.169.254/metadata/instance/compute` and verifies the response body contains the information (SubscriptionId, ResourceGroup, VMId, ResourceId) of the VM. The query time-out is 10s. If the query fails, it retries at most another three more times with an interval of 15s, 30s, and 45s.	• IMDS_INTERVAL: the execution interval Default: 180s • IMDS_ENDPOINT: the URL of the IMDS Default: `http://169.254.169.254/metadat` • IMDS_TIMEOUT_IN_SECONDS: the time each query. Default: 10 • IMDS_QUERY_TOTAL_ATTEMPTS: the to attempts to send http request if the pre Default: 4 • IMDS_RETRY_INTERVAL_IN_SEONDS: th seconds if the previous http request fail
process	Always eligible	This collector is executed every 180s. In each execution, it creates and executes command `${SYTEM_DIR}\system32\cmd.exe /c echo hello` in Windows machine and `/bin/sh -c echo hello` in Linux machine. The time-out of process execution is 10s.	• PROCESS_INTERVAL: the execution inte Default: 180s • PROCESS_TIMEOUT: the time-out of pro Default: 10s
process_memory	Always eligible	This collector is executed every 180s. In each execution, it selects the top three processes with the most memory usage and reports the ProcessRSSPercent, ProcessPageFaults, MachineMemoryTotalInBytes, MachineMemoryUsedPercent, and TotalPageFaults.	• PROCESS_MEMORY_INTERVAL: the exec Collector. Default: 180s
process_cpu	Always eligible	This collector is executed every 180s. In each execution, it selects the top three processes with the most CPU usage and reports the ProcessCoreUsage, ProcessMachineUsage, and MachineTotalCpuUsage.	• PROCESS_CPU_INTERVAL: the executior Collector. Default: 180s
process_monitor	Always eligible	Not executed. If explicitly enabled by the user, this collector verifies if the selected process is running and collect its running time in seconds.	• PROCESS_MONITOR_INTERVAL: the exe Collector. Default: 180s • PROCESS_MONITOR_PROCESS_NAMES Expression of process names to be mon `,`. No default value
system_error	Eligible in Windows machine	The Collector is executed every three mins. In each execution, it subscribes to the "System" channel of Windows EventLog and queries events with level defined in SystemData <=2 (including	• SYSTEM_ERROR_MEASUREMENT_TARG total different measurementTargets in e Default: 10

Collector Name	Eligibility	Default Behavior	Overwritable Parameters
		LOG_ALWAYS, Critical, Error). The measurementTarget is defined as Source_EventId of the EventLog using default Windows locale. A cap of no more than 10 different measurementTargets is applied in each collection.	
az_storage_blob	Eligible if EnvironmentAttribute "OutboundConnectivityDisabled" isn't set or set to "false"	Not executed. If explicitly enabled by the user, this collector verifies if the VM can have access to the selected Azure Storage Blob by using either Managed Identity or SAS token.	• AZ_STORAGE_BLOB_INTERVAL: the exec Collector. Default: 180s • AZ_STORAGE_ACCOUNT_NAME: the Az name. No default value • AZ_STORAGE_CONTAINER_NAME: the A Container name. No default value • AZ_STORAGE_BLOB_NAME: the Azure S No default value • AZ_STORAGE_BLOB_DOMAIN_NAME: th domain name. No default value • AZ_STORAGE_SAS_TOKEN_BASE64: the Azure Storage SAS token. No default va • AZ_STORAGE_USE_MANAGED_IDENTIT identity will be used for authentication. • AZ_STORAGE_MANAGED_IDENTITY_CLI identity client ID for authentication. No
hardware_health_monitor	Eligible in Windows machine	Not executed. If explicitly enabled by the user, this collector collects hardware health info from Windows event log, currently only disk related critical events are collected, including events with ID 7, 500, 504, 505, 512 and 549.	• HARDWARE_HEALTH_MONITOR_INTER interval of the Collector. Default: 180s
hardware_health_nvidia_smi	Eligible in Linux Ubuntu machine	Not executed. If explicitly enabled by the user, this collector collects hardware health info from Windows event log, currently only disk related critical events are collected, including events with ID 7, 500, 504, 505, 512 and 549.	• HARDWARE_HEALTH_NVIDIA_SMI_INTE interval of the Collector. Default: 60s • HARDWARE_HEALTH_NVIDIA_SMI_INTE running /usr/bin/nvidia-smi command.

Next steps

- Configure VM watch
- Configure Event Hubs for VM watch
- Install VM watch
- VM watch overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Configure VM watch

Article • 02/28/2025

VM watch signals are logically grouped into Collectors Suite, which can be categorized into two groups: core and optional. By default, only core group collectors are enabled with default configurations. However, these default settings can be easily overwritten from `vmWatchSettings` using either ARM template, Azure CLI, or PowerShell.

This article describes how to configure VM watch to suit your specific requirements.

Prerequisites

This article assumes that you're familiar with:

- VM watch checks, metrics, and logs
- Installing VM watch to virtual machines and scale sets
- VM watch Collectors Suite

Access `vmWatchSettings` on Azure virtual machines

> **ⓘ Important**
>
> The code segment is identical for both Windows and Linux except for the value of the parameter `<application health extension type>` passed into the Extension Type. Replace `<application health extension type>` with `ApplicationHealthLinux` for Linux and `ApplicationHealthWindows` for Windows installations.

ARM Template

1. Navigate to the Overview page on Azure portal and click on the JSON view for the VM to find the code segment below.
2. Copy the code segment to an IDE such as Visual Studio Code and make customizations as needed

```
{
   "settings": {
      "vmWatchSettings": {
         "enabled": true
      }
```

```
    }
}
```

Customize VM watch configurations

VM watch signals can be customized by configuring the `vmWatchSettings` properties to meet specific requirements. The following table lists the properties for `vmWatchSettings`.

vmWatchSettings Properties

Expand table

Name	Type	Description	Is Required
enabled	`bool`	This allows you to enable or disable VM watch	true
signalFilters	`object`	This filters the enabled / disabled signals, either by tag or collector name.	false
parameterOverrides	`object`	This specifies the parameters that can be overwritten for each signal execution. The full list of overwritable parameters can be found in the VM watch Collectors Suite page.	false
environmentAttributes	`object`	This specifies any environment attributes that help decide if a test is eligible to execute or not.	false

> **Important**
>
> For the full list of collectors, associated signals, tags, overwritable parameters, and environment attributes, visit VM watch Collectors Suite page

Switch on/off VM watch

VM watch can be switched on / off by configuring the `enabled` property, as shown in the code segment.

```
{
   "vmWatchSettings": {
      "enabled": true
```

```
    }
}
```

> **Note**

Expand table

Name	Description
true	This setting enables VM watch
false	This setting disables VM watch

Enable/Disable signal execution

By default, only the core group signals are enabled. However, the `signalFilters` property can be used to control and configure the signals to be executed. This property includes the following subfields.

Expand table

Subfields	Description
enabledTags	This enables the signals in the optional group specified with tags provided by the user
disabledTags	This disables the signals in the core and optional groups specified with tags provided by the user
enabledOptionalSignals	This enables signals specified in optional group. Provide collector name(s) as parameter
disabledSignals	This disables the signals specified in the core and optional groups. Provide collector name(s) as parameter

For instance, to enable signals in the optional group containing `Network` tag and disable signals containing `Disk` tag, specify such tags under the `enabledTags` and `disabledTags` as shown:

```
{
   "vmWatchSettings": {
      "enabled": true,
      "signalFilters": {
```

```
         "enabledTags": [
            "Network"
         ],
         "disabledTags": [
            "Disk"
         ]
      }
   }
}
```

Similarly, to enable an optional group signal with name `hardware_health_monitor`, and disable signals with name `process` and `dns`, specify such names under the `enabledOptionalSignals` and `disabledSignals` as shown:

```
{
   "vmWatchSettings": {
      "enabled": true,
      "signalFilters": {
         "enabledOptionalSignals": [
            "hardware_health_monitor"
         ],
         "disabledSignals": [
            "process",
            "dns"
         ]
      }
   }
}
```

Configure signal execution frequency

The signal execution frequency can be customized by adjusting the `parameterOverrides` property. For instance, to set the `outbound_connectivity` test execution frequency to 120 seconds, specify the following configuration:

```
{
   "vmWatchSettings": {
      "enabled": true,
      "parameterOverrides": {
         "OUTBOUND_CONNECTIVITY_INTERVAL": "120s"
      }
   }
}
```

Override default signal execution parameters

Signal execution parameters can be overwritten by setting the `parameterOverrides` property. For instance, to set `disk_io` signal mount point to `/mnt`, the following configuration can be specified:

```
{
   "vmWatchSettings": {
      "enabled": true,
      "parameterOverrides": {
         "DISK_IO_MOUNT_POINTS": "/mnt"
      }
   }
}
```

Environment attribute enrichments

In addition to tags, VM watch also checks the eligibility of the signals before execution. The `environmentAttributes` can be specified to help VM watch determine the eligibility of each signal for execution. For instance, if outbound traffic has been disabled on a VM, this information can be provided to VM watch. This ensures that any outbound network-related signal execution will be marked as ineligible.

```
{
   "vmWatchSettings": {
      "enabled": true,
      "environmentAttributes": {
         "OutboundConnectivityDisabled": true
      }
   }
}
```

Next steps

- VM watch Collectors Suite
- Configure Event Hubs for VM watch
- Install VM watch
- VM watch overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Configure Event Hubs for VM watch

Article • 03/03/2025

VM watch can send signal data to a preconfigured Event Hub.

This article provides instructions on configuring Event Hubs to access signals collected by VM watch

Prerequisites

This article assumes that you're familiar with:

- Azure Event Hubs
- VM watch checks, metrics, and logs
- Install VM watch to virtual machines and scale sets

Enable Event Hubs Output

1: Prepare Event Hubs for VM watch

- Deploy an Event Hub
- Authorize access to the Azure Event Hub

> **ⓘ Important**
>
> VM watch supports managed identity, SAS token, and connection string as authentication methods. When multiple authentication methods are provided, managed identity is prioritized as the highest, while the connection string is assigned the lowest priority.

2: Enable Event Hubs output for VM watch

For each Event Hub authentication method, you need to combine both the common and authentication specific parameter settings. Instructions are given for each authentication scenario. For virtual machines and virtual machine scale sets, specify the following settings within `vmWatchSettings` in the JSON configurations. See Configure VM watch for instructions on how to access `vmWatchSettings` using ARM template, Azure CLI, or PowerShell.

Common parameters for Event Hubs output

For all authentication methods, the following parameter set applies:

Expand table

Parameter	Is required	Description
`EVENT_HUB_OUTPUT_NAMESPACE`	Yes	Event hub name space name, without the domain name `".servicebus.windows.net"`
`EVENT_HUB_OUTPUT_NAME`	Yes	Event hub name within the given namespace
`EVENT_HUB_OUTPUT_DOMAIN_NAME`	No	Event hub domain name. Default value `"servicebus.windows.net"`
`EVENT_HUB_OUTPUT_CLOSE_TIMEOUT`	No	Client close time-out. Default is 30s
`EVENT_HUB_OUTPUT_PARTITION_ID`	No	Metric tag or field name to use for the event partition key. Default is null
`EVENT_HUB_OUTPUT_MAX_MESSAGE_SIZE`	No	The maximum batch message size in bytes. Setting this parameter to 0 means using the default size from the Azure Event Hubs Client library (1,000,000 bytes). Default is 0
`SEND_INTERNAL_TELEMETRY_TO_EVENT_HUB`	No	To receive VM watch internal metrics (startup and heartbeat events), set this value to "true." Default is "false"

Authentication specific parameters for Event Hubs output

Managed Identity

Expand table

Parameter	Is required	Description
`EVENT_HUB_OUTPUT_USE_MANAGED_IDENTITY`	No	Set this value to "true." Default is "false"
`EVENT_HUB_OUTPUT_MANAGED_IDENTITY_CLIENT_ID`	No	If you're using a specific managed identity to authenticate, specify this value

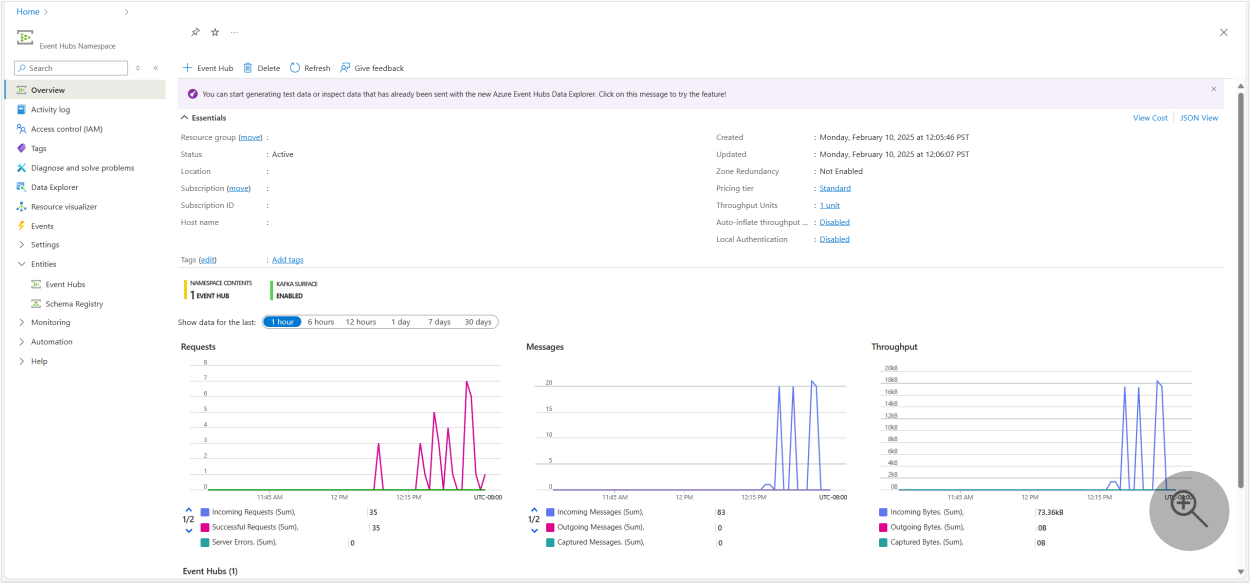
For example, the following VM watch JSON configuration sets the environment variables `EVENT_HUB_OUTPUT_NAMESPACE`, `EVENT_HUB_OUTPUT_NAME`, and `EVENT_HUB_OUTPUT_USE_MANAGED_IDENTITY`. This allows Event Hubs to use managed identity as the authentication method without needing to specify a managed identity client ID.

```
{
  "vmWatchSettings": {
    "enabled": true,
    "parameterOverrides": {
      "EVENT_HUB_OUTPUT_NAMESPACE": "<example event hub namespace>",
      "EVENT_HUB_OUTPUT_NAME": "<example event hub name>",
      "EVENT_HUB_OUTPUT_USE_MANAGED_IDENTITY": "true"
    }
  }
}
```

3: Examine events in Event Hubs

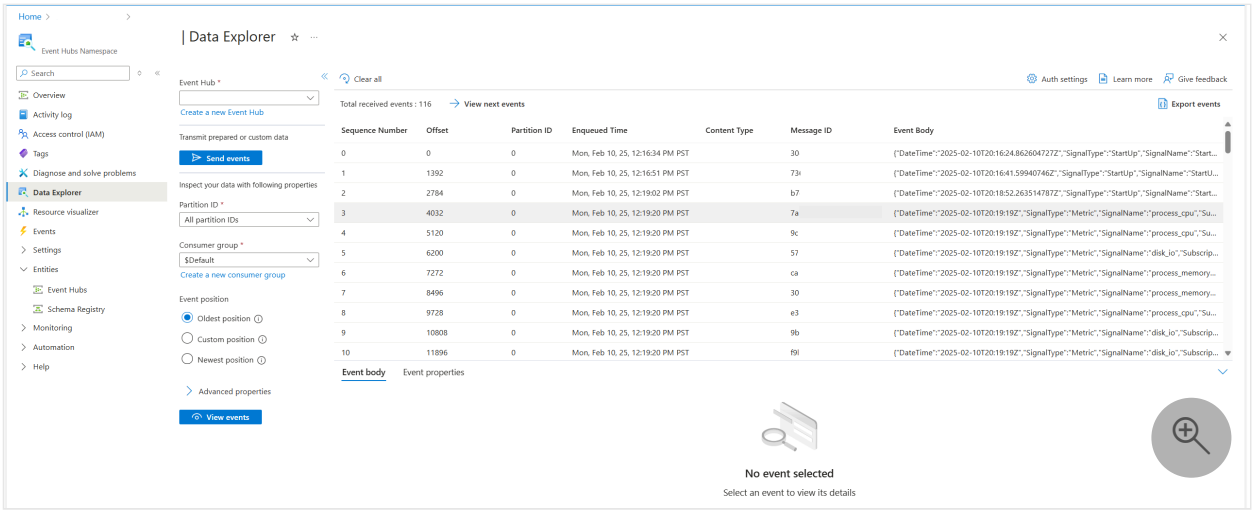
Once VM watch settings are successfully configured to use Event Hubs as the output, VM watch restarts. Events start flowing into Event Hubs within a few minutes. You can use the Azure portal to observe the incoming messages.

The following screenshot shows data flowing into the Event Hub



Also, you can use the Event Hubs Data Explorer feature to view incoming event and content.

The following screenshot shows Event Hubs Data Explorer



Event Hubs Events schema

Each Event Hub event has the following schema:

Expand table

Field Name	Data Type	Description
`DateTime`	time	The time the signal was emitted
`SignalType`	string	The type of the signal, which can be either "StartUp," "Heartbeat," "Check," "Metric," or "EventLog"
`SignalName`	string	The name of the signal
`SubscriptionId`	string	The VM subscription ID
`ResourceGroup`	string	The VM resource group name
`ResourceId`	string	The Azure resource URI of the VM
`VmId`	string	The unique ID of the VM queried from IMDS endpoint within the VM
`Vmss`	string	The virtual machine scale set name, if applicable
`Offer`	string	The Azure VM offer
`VmSize`	string	The VM size
`MeasurementTarget`	string	The target that the signal is measuring at. `Name` field and `MeasurementTarget` are used together for raw signal aggregation
`SignalValue`	json	The value of this signal, where the schema depends on the SignalType

Field Name	Data Type	Description
`Version`	string	The version of the Event Hub output event schema

Debug Event Hubs connection issues

If there are no events in Event Hubs after several minutes, check the VM watch logs in the following directories on the virtual machine or virtual machine scale set to diagnose the issue:

Linux

```
/var/log/azure/Microsoft.ManagedServices.ApplicationHealthLinux/vmwatch.log
```

Next steps

- Configure VM watch
- VM watch Collectors Suite
- Install VM watch
- VM watch overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Project Flash - Advancing Azure Virtual Machine availability monitoring

Article • 08/22/2024

Flash, as the project is internally known, derives its name from our steadfast commitment to building a robust, reliable, and rapid mechanism for customers to monitor virtual machine (VM) health. Our primary objective is to ensure customers can reliably access actionable and precise telemetry, promptly receive alerts on changes, and periodically monitor data at scale. We also place strong emphasis on developing a centralized and coherent experience that customers can conveniently use to meet their unique observability requirements. It's our mission to ensure you can:

- **Consume accurate and actionable data** on VM availability disruptions (for example, VM reboots and restarts, application freezes due to network driver updates, and 30-second host OS updates), along with precise failure details (for example, platform versus user-initiated, reboot versus freeze, planned versus unplanned).
- **Analyze and alert on trends in VM availability** for quick debugging and month-over-month reporting.
- **Periodically monitor data at scale** and build custom dashboards to stay updated on the latest availability states of all resources.
- **Receive automated root cause analyses (RCAs)** detailing impacted VMs, downtime cause and duration, consequent fixes, and similar—all to enable targeted investigations and post-mortem analyses.
- **Receive instantaneous notifications** on critical changes in VM availability to quickly trigger remediation actions and prevent end-user impact.
- **Dynamically tailor and automate platform recovery policies** , based on ever-changing workload sensitivities and failover needs.

Flash solutions

The Flash initiative is dedicated to developing solutions over the years that cater to the diverse monitoring needs of our customers. To help you determine the most suitable Flash monitoring solution(s) for your specific requirements, refer to the following table:

Expand table

Solution	Description
Azure Resource Graph (General Availability)	For investigations at scale, centralized resource repository and history lookup, large customers want to periodically consume resource availability telemetry across all their workloads, at once, using Azure Resource Graph (ARG).
Event Grid system topic (Public Preview)	To trigger time-sensitive and critical mitigations (redeploy, restart VM actions) for prevention of end-user impact, customers (for example, Pearl Abyss, Krafton) want to receive alerts within seconds of critical changes in resource availability via Event Handlers in Event Grid.
Azure Monitor (Public Preview)	To track trends, aggregate platform metrics (CPU, disk etc.), and set up precise threshold-based alerts, customers want to consume an out-of-box VM Availability metric via Azure Monitor.
Resource Health (General Availability)	To perform instantaneous and convenient Portal UI health checks per-resource customers can quickly view the RHC blade on the portal. They can also access a 30-day historical view of health checks for that resource for quick and easy troubleshooting.

Holistic VM availability monitoring

For a holistic approach to monitoring VM availability, including scenarios of routine maintenance, live migration, service healing, and VM degradation, we recommend you utilize both scheduled events (SE) and Flash health events.

Scheduled events are designed to offer an early warning, giving up to 15-minute advance notice prior to maintenance activities. This lead time enables you to make informed decisions regarding upcoming downtime, allowing you to either avoid or prepare for it. You have the flexibility to either acknowledge these events or delay actions during this 15-minute period, depending on your readiness for the upcoming maintenance.

On the other hand, Flash Health events are focused on real-time tracking of ongoing and completed availability disruptions, including VM degradation. This feature empowers you to effectively monitor and manage downtime, supporting automated mitigation, investigations, and post-mortem analysis.

To get started on your observability journey, you can explore the suite of Azure products to which we emit high-quality VM availability data. These products include resource health, activity logs, Azure resource graph, Azure monitor metrics and Azure Event Grid system topic.

Next steps

To learn more about the solutions offered, proceed to corresponding solution article:

- Use Azure Resource Graph to monitor Azure Virtual Machine availability
- Use Event Grid system topics to monitor Azure Virtual Machine availability
- Use Azure Monitor to monitor Azure Virtual Machine availability
- Use Azure Resource Health to monitor Azure Virtual Machine availability

For a general overview of how to monitor Azure Virtual Machines, see Monitor Azure virtual machines and the Monitoring Azure virtual machines reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Project Flash - Use Azure Resource Graph to monitor Azure Virtual Machine availability

Article • 08/22/2024

Azure Resource Graph is one solution offered by Flash. Flash is the internal name for a project dedicated to building a robust, reliable, and rapid mechanism for customers to monitor virtual machine (VM) health.

This article covers the use of Azure Resource Graph to monitor Azure Virtual Machine availability. For a general overview of Flash solutions, see the Flash overview.

For documentation specific to the other solutions offered by Flash, choose from the following articles:

- Use Event Grid system topics to monitor Azure Virtual Machine availability
- Use Azure Monitor to monitor Azure Virtual Machine availability
- Use Azure Resource Health to monitor Azure Virtual Machine availability

Azure Resource Graph - HealthResources

This feature is currently generally available. It's useful for conducting large-scale investigations. It offers a highly user-friendly experience for information retrieval with its use of kusto query language (KQL). It can also serve as a central hub for resource information and allows easy retrieval of historical data.

In addition to already flowing VM availability states, we published VM availability annotations to Azure Resource Graph (ARG) for detailed failure attribution and downtime analysis, along with enabling a 14-day change tracking mechanism to trace historical changes in VM availability for quick debugging. With these new additions, we're excited to announce the general availability of VM availability information in the HealthResources dataset in ARG! With this offering users can:

- Efficiently query the latest snapshot of VM availability across all Azure subscriptions at once and at low latencies for periodic and fleetwide monitoring.
- Accurately assess the impact to fleetwide business SLAs and quickly trigger decisive mitigation actions, in response to disruptions and type of failure signature.
- Set up custom dashboards to supervise the comprehensive health of applications by joining VM availability information with resource metadata present in ARG.

- Track relevant changes in VM availability across a rolling 14-day window, by using the change-tracking mechanism for conducting detailed investigations.

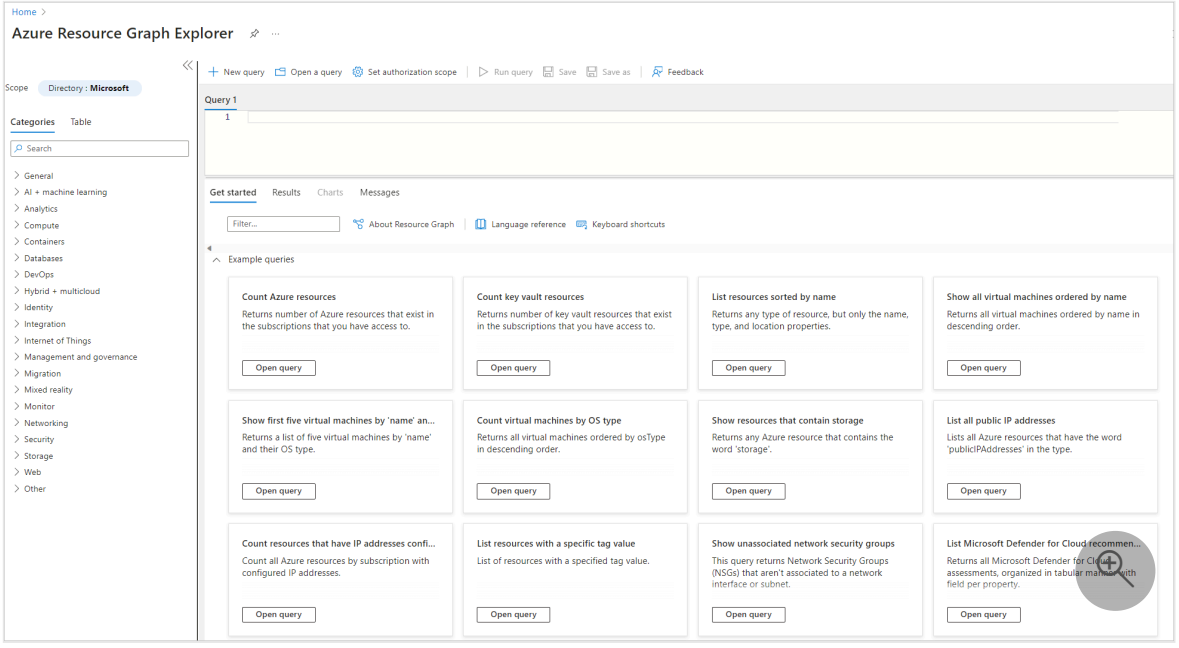
Sample queries

- Azure Resource Graph sample queries for Azure Service Health - Azure Service Health | Microsoft Learn
- VM availability information in Azure Resource Graph - Azure Virtual Machines | Microsoft Learn
- List of sample Azure Resource Graph queries by table - Azure Resource Graph | Microsoft Learn

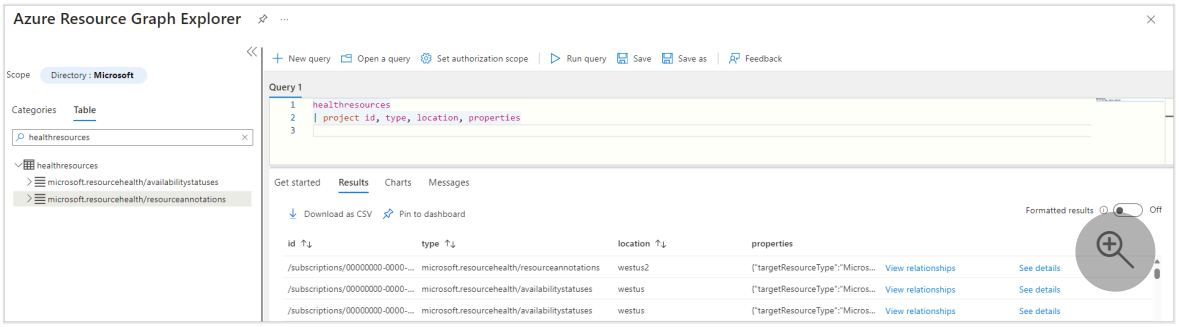
Get started

Users can query ARG via PowerShell, REST API, Azure CLI, or even the Azure portal. The following steps detail how data can be accessed from Azure portal.

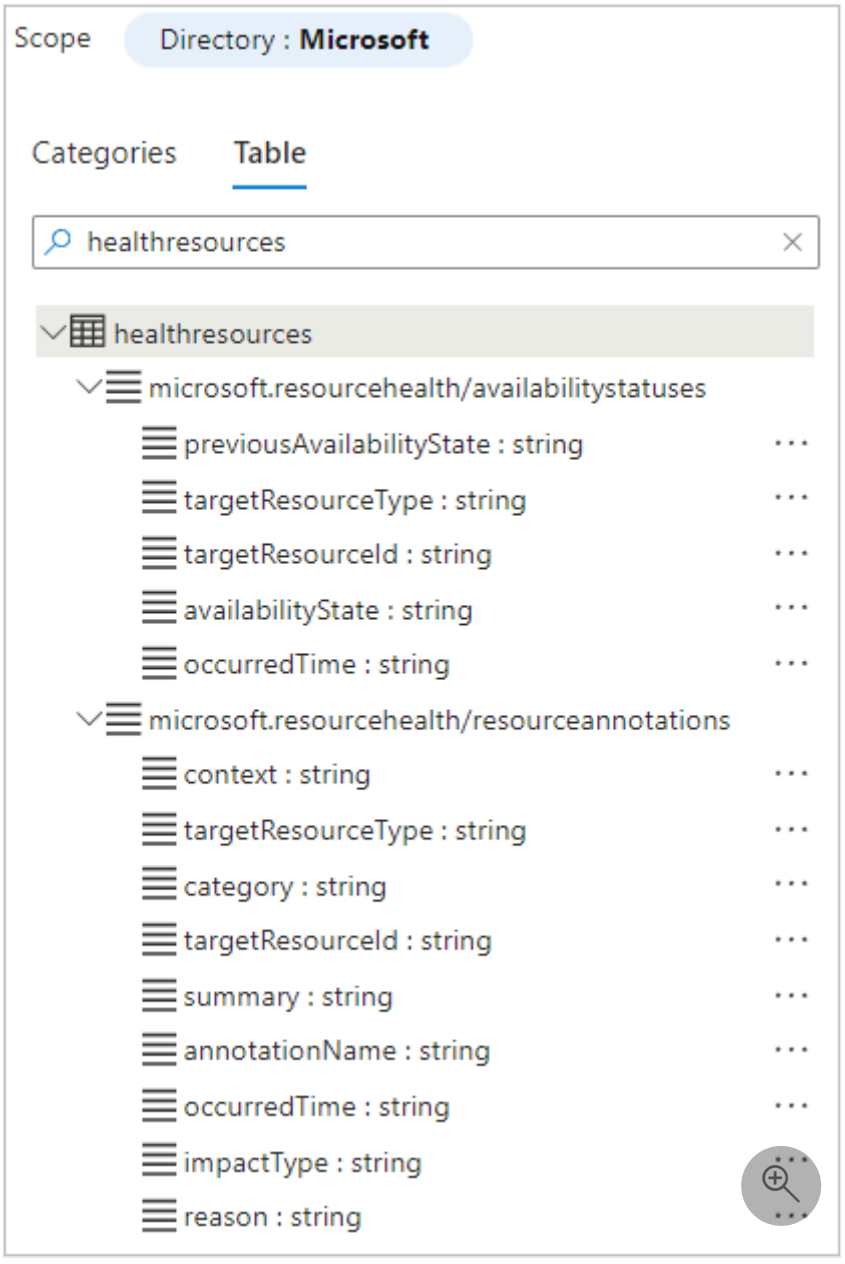
1. Once on the Azure portal, navigate to Resource Graph Explorer.



2. Select the Table tab and (single) click on the HealthResources table to retrieve the latest snapshot of VM availability information (availability state and health annotations).



There are two types of events populated in the HealthResources table:



- resourcehealth/availabilitystatuses

This event denotes the latest availability status of a VM, based on the health checks performed by the underlying Azure platform. The availability states we currently emit for VMs are:

- Available: The VM is up and running as expected.

- Unavailable: We detected disruptions to the normal functioning of the VM, and therefore, applications won't run as expected.
- Unknown: The platform is unable to accurately detect the health of the VM. Users can usually check back in a few minutes for an updated state.

To poll the latest VM availability state, refer to the properties field, which contains the following details:

Sample

```
{
 "targetResourceType": "Microsoft.Compute/virtualMachines",
 "previousAvailabilityState": "Available",
 "targetResourceId":
"/subscriptions//resourceGroups//providers/Microsoft.Compute/virtualMachines
/",
 "occurredTime": "2022-10-11T11:13:59.9570000Z",
 "availabilityState": "Unavailable"
 }
```

Property description

Expand table

Property	Description	Corresponding resource health category (RHC)
targetResourceType	Type of resource for which health data is flowing	resourceType
targetResourceId	Resource ID	resourceId
occurredTime	Timestamp when the platform emits the latest availability state	eventTimestamp
previousAvailabilityState	Previous availability state of the VM	previousHealthStatus
availabilityState	Current availability state of the VM	currentHealthStatus

See the HealthResources section of the samples queries documentation for a list of starter queries to further explore this data.

- resourcehealth/resourceannotations (NEWLY ADDED)

This event contextualizes any changes to VM availability, by detailing necessary failure attributes to help users investigate and mitigate the disruption as needed. See the full list of VM availability annotations emitted by the platform. These annotations can be broadly classified into three buckets:

- Downtime Annotations: These annotations are emitted when the platform detects VM availability transitioning to Unavailable. (For example, during unexpected host crashes, rebootful repair operations).
- Informational Annotations: These annotations are emitted during control plane activities with no impact to VM availability. (Such as VM allocation/Stop/Delete/Start). Usually, no further customer action is required in response.
- Degraded Annotations: These annotations are emitted when VM availability is detected to be at risk. (For example, when failure prediction models predict a degraded hardware component that can cause the VM to reboot at any given time). We strongly urge users to redeploy by the deadline specified in the annotation message, to avoid any unanticipated loss of data or downtime. You may receive an alert in Azure virtual machine scale sets Resource Health or Activity log in one of the following scenarios:
  - VMs in the Azure virtual machine scale sets are in the process of being stopped, deallocated, deleted, or started.
  - You performed scaling in or out operations on the virtual machine scale sets.
  - The alert indicates that the aggregated platform health of the virtual machine scale sets is in a transient state of "Degraded."

To poll the associated VM availability annotations for a resource, if any, refer to the properties field, which contains the following details:

Sample


```
{
 "targetResourceType": "Microsoft.Compute/virtualMachines",
"targetResourceId":
"/subscriptions//resourceGroups//providers/Microsoft.Compute/virtualMachines
/",
 "annotationName": "VirtualMachineHostRebootedForRepair",
 "occurredTime": "2022-09-25T20:21:37.5280000Z",
 "category": "Unplanned",
 "summary": "We're sorry, your virtual machine isn't available because an
unexpected failure on the host server. Azure has begun the auto-recovery
process and is currently rebooting the host server. No further action is
required from you at this time. The virtual machine will be back online
after the reboot completes.",
```

```
  "context": "Platform Initiated",
  "reason": "Unexpected host failure"
  }
```

Property description

Expand table

Property	Description	Corresponding RHC
targetResourceType	Type of resource for which health data is flowing	resourceType
targetResourceId	Resource ID	resourceId
occurredTime	Timestamp when the latest availability state is emitted by the platform	eventTimestamp
annotationName	Name of the Annotation emitted	eventName
reason	Brief overview of the availability impact observed by the customer	title
category	Denotes whether the platform activity that triggered the annotation was either planned maintenance or unplanned repair. This field isn't applicable to customer/VM-initiated events. Possible values: Planned, Unplanned, Not Applicable, Null	category
context	Denotes whether the activity that triggered the annotation was due to an authorized user or process (customer-initiated), the Azure platform (platform-initiated), or activity in the guest OS that resulted in availability impact (VM initiated). Possible values: Platform-initiated, User-initiated, VM-initiated, Not Applicable, Null	context
summary	Statement detailing the cause for annotation emission, along with remediation steps that users can take	summary

See the HealthResources section of the samples queries documentation for a list of starter queries to further explore this data.

We have multiple enhancements planned for the annotation metadata that is surfaced in the HealthResources dataset. These enrichments give users access to richer failure attributes to decisively prepare a response to a disruption. In parallel, we aim to extend the duration of historical lookback to a minimum of 30 days so users can comprehensively track past changes in VM availability.

Next steps

To learn more about the solutions offered, proceed to corresponding solution article:

- Use Event Grid system topics to monitor Azure Virtual Machine availability
- Use Azure Monitor to monitor Azure Virtual Machine availability
- Use Azure Resource Health to monitor Azure Virtual Machine availability

For a general overview of how to monitor Azure Virtual Machines, see Monitor Azure virtual machines and the Monitoring Azure virtual machines reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Project Flash - Use Azure Event Grid to monitor Azure Virtual Machine availability

Article • 08/22/2024

Azure Event Grid is one solution offered by Flash. Flash is the internal name for a project dedicated to building a robust, reliable, and rapid mechanism for customers to monitor virtual machine (VM) health.

This article covers the use of Azure Event Grid system topics to monitor Azure Virtual Machine availability. For a general overview of Flash solutions, see the Flash overview.

For documentation specific to the other solutions offered by Flash, choose from the following articles:

- Use Azure Monitor to monitor Azure Virtual Machine availability
- Use Azure Resource Health to monitor Azure Virtual Machine availability
- Use Azure Resource Graph to monitor Azure Virtual Machine availability

Azure Event Grid system topic - HealthResources

To ensure seamless operation of business-critical applications, it's crucial to have real time awareness of any event that might adversely impact VM availability. This awareness enables you to swiftly take remedial actions to shield end-users from any disruption. To support you in your daily operations, we're delighted to announce the public preview of the HealthResources Event Grid system topic with newly added VM availability annotations!

This system topic provides in-depth VM health data, giving you immediate insights into changes in VM availability states along with the necessary context. You can receive events on single-instance VMs and Virtual Machine Scale Set VMs for the Azure subscription for which this topic was created. Data is published to this topic by Azure Resource Notifications (ARN), our state-of-the-art publisher-subscriber service, equipped with robust Role-Based Access Control (RBAC) and advanced filtering capabilities. This empowers you to effortlessly subscribe to an Event Grid system topic and seamlessly direct relevant events utilizing the advanced filtering capabilities provided by Event Grid, to downstream tools in real-time. This enables you to respond and mitigate issues instantly.

Get started

- Step 1: Users start by creating a systemtopic within the Azure subscription for which they want to receive notifications.
- Step 2: Users then proceed to create an event subscription within the system topic in Step 1. During this step, they specify the endpoint (such as Event Hubs or Azure Monitor Alerts) to which the events are routed. Users can also configure event filters to narrow down the scope of delivered events.

As you start subscribing to events from the HealthResources system topic, consider the following best practices:

- Choose an appropriate destination or event handler based on the anticipated scale and size of events.
- For fan-in scenarios where notifications from multiple system topics need to be consolidated, event hubs are highly recommended as a destination. This practice is especially useful for real-time processing scenarios to maintain data freshness and for periodic processing for analytics, with configurable retention periods.
- NEW: Customers can now subscribe to Health Resources events and send them to Azure Monitor alerts as new destination. For step-by-step guide, see Subscribe to Health Resources events and send them to Azure monitor alerts.

We have plans to transition the preview into a fully fledged general availability feature. As part of the preview, we emit events scoped to changes in VM availability states with the following sample schema:

Sample


```
{
 "id": "4c70abbc-4aeb-4cac-b0eb-ccf06c7cd102",
 "topic": "/subscriptions/,
 "subject":
"/subscriptions//resourceGroups//providers/Microsoft.Compute/virtualMachines
//providers/Microsoft.ResourceHealth/AvailabilityStatuses/current",
 "data": {
 "resourceInfo": {

"id":"/subscriptions//resourceGroups//providers/Microsoft.Compute/virtualMac
hines//providers/Microsoft.ResourceHealth/AvailabilityStatuses/current",
 "properties": {

"targetResourceId":"/subscriptions//resourceGroups//providers/Microsoft.Comp
ute/virtualMachines/"
 "targetResourceType": "Microsoft.Compute/virtualMachines",
```

```
 "occurredTime": "2022-09-25T20:21:37.5280000Z"
 "previousAvailabilityState": "Available",
 "availabilityState": "Unavailable"
 }
 },
 "apiVersion": "2020-09-01"
 },
 "eventType":
"Microsoft.ResourceNotifications.HealthResources.AvailabilityStatusesChanged
",
 "dataVersion": "1",
 "metadataVersion": "1",
 "eventTime": "2022-09-25T20:21:37.5280000Z"
 }
```


The properties field is fully consistent with the `microsoft.resourcehealth/availabilitystatuses` event in ARG. The Event Grid solution offers near-real-time alerting capabilities on the data present in ARG.

Next steps

To learn more about the solutions offered, proceed to corresponding solution article:

- Use Azure Monitor to monitor Azure Virtual Machine availability
- Use Azure Resource Health to monitor Azure Virtual Machine availability
- Use Azure Resource Graph to monitor Azure Virtual Machine availability

For a general overview of how to monitor Azure Virtual Machines, see Monitor Azure virtual machines and the Monitoring Azure virtual machines reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Project Flash - Use Azure Monitor to monitor Azure Virtual Machine availability

Article • 10/18/2024

Azure Monitor is one solution offered by Flash. Flash is the internal name for a project dedicated to building a robust, reliable, and rapid mechanism for customers to monitor virtual machine (VM) health.

This article covers the use of the Azure Monitor VM availability metric to monitor Azure Virtual Machine availability. For a general overview of Flash solutions, see the Flash overview.

For documentation specific to the other solutions offered by Flash, choose from the following articles:

- Use Azure Resource Health to monitor Azure Virtual Machine availability
- Use Azure Resource Graph to monitor Azure Virtual Machine availability
- Use Event Grid system topics to monitor Azure Virtual Machine availability

Azure monitor - VM availability metric

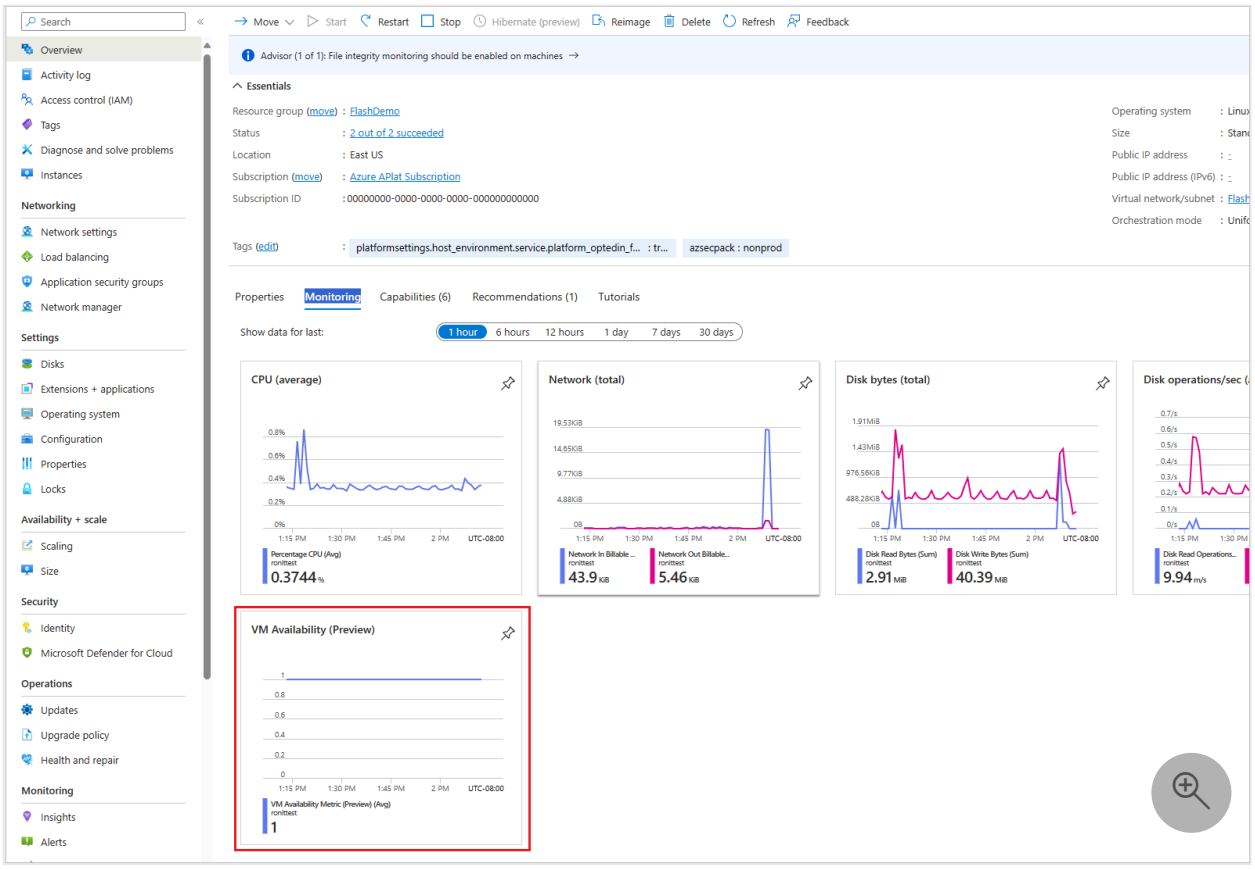
Currently in public preview. The VM availability metric is well-suited for tracking trends, aggregating platform metrics (such as CPU and disk usage), and configuring precise threshold-based alerts. Customers can utilize this out-of-the-box VM availability metric in Azure Monitor. This metric displays the trend of VM availability over time, so users can:

- Set up threshold-based metric alerts on dipping VM availability to quickly trigger appropriate mitigation actions.
- Correlate the VM availability metric with existing platform metrics like memory, network, or disk for deeper insights into concerning changes that impact the overall performance of workloads.
- Easily interact with and chart metric data during any relevant time window on Metrics Explorer, for quick and easy debugging.
- Route metrics to downstream tooling like Grafana dashboards, for constructing custom visualizations and dashboards.

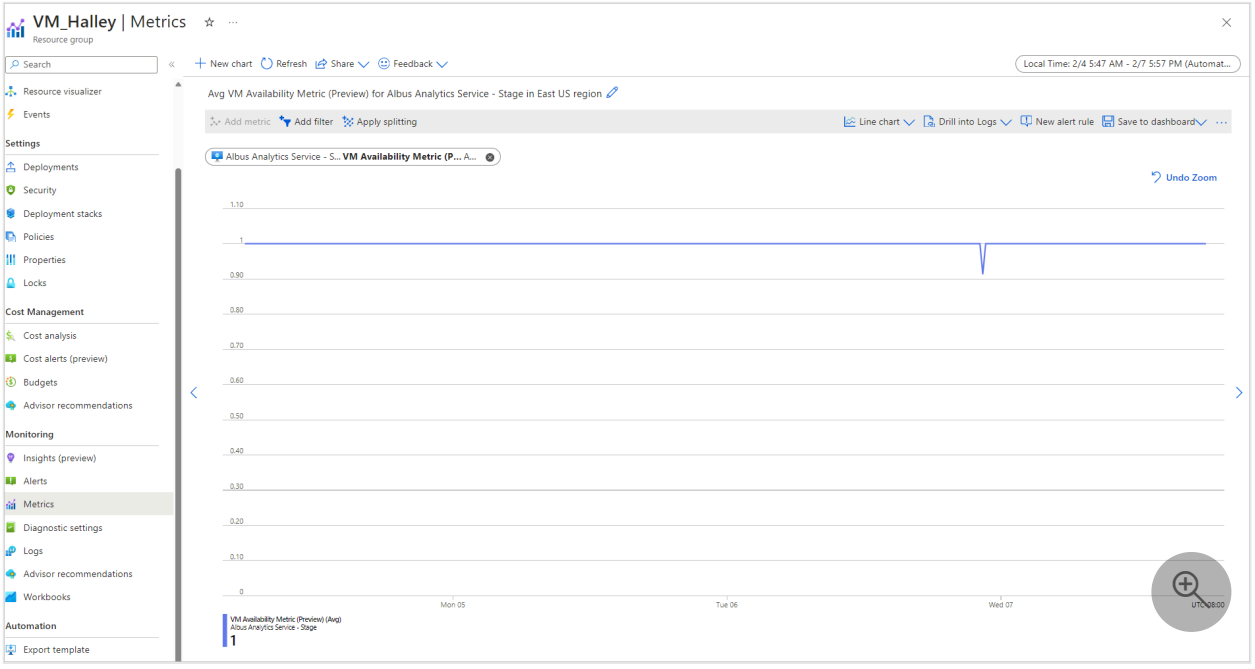
Get started

Users can either consume the metric programmatically via the Azure Monitor REST API or directly from the Azure portal. The following steps highlight metric consumption from the Azure portal.

Once on the Azure portal, navigate to the VM overview blade. The new metric is displayed as VM Availability (Preview), along with other platform metrics under the Monitoring tab.

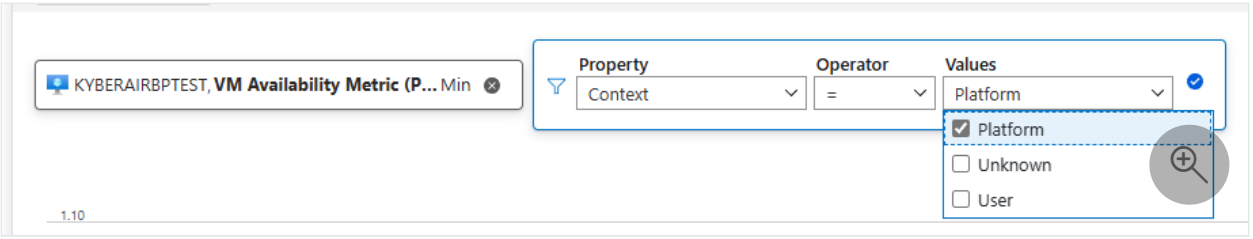


To navigate to the metrics explorer for further analysis, select the VM availability metric chart on the overview page.



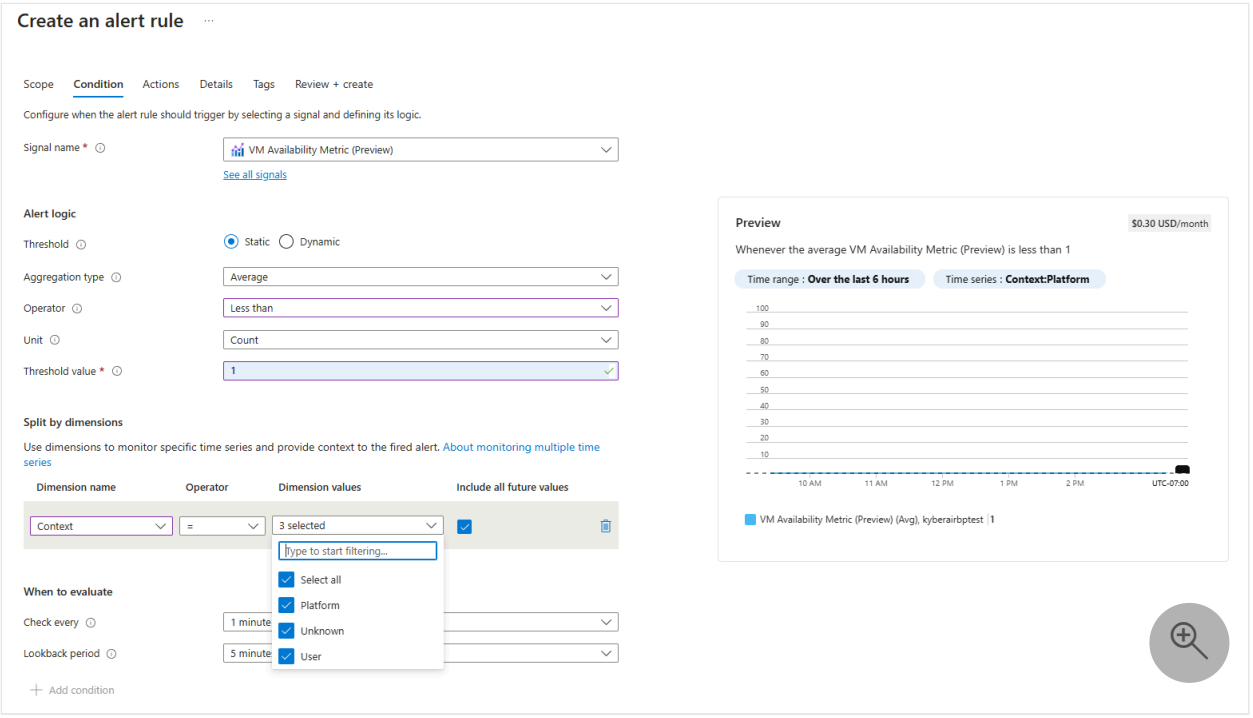
For a description of the metric's display values, see VM availability metric.

Users can split the VM availability by the 'Context' property.



> **Note**
>
> The context for Service Healing and Live Migration is currently unknown by default.

Users can create an alert rule based on dimension values. Under the Condition tab, choose VM Availability Metric as the Signal name. In the Split by dimensions section, enter Context as the dimension name and select the corresponding dimension values.



Useful links

- How to filter events for Azure Event Grid - Azure Event Grid | Microsoft Learn
- Event filtering for Azure Event Grid - Azure Event Grid | Microsoft Learn

Next steps

To learn more about the solutions offered, proceed to corresponding solution article:

- Use Azure Resource Health to monitor Azure Virtual Machine availability

- Use Azure Resource Graph to monitor Azure Virtual Machine availability
- Use Event Grid system topics to monitor Azure Virtual Machine availability

For a general overview of how to monitor Azure Virtual Machines, see Monitor Azure virtual machines and the Monitoring Azure virtual machines reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Project Flash - Use Azure Resource Health to monitor Azure Virtual Machine availability

Article • 08/22/2024

Azure Resource Health is one solution offered by Flash. Flash is the internal name for a project dedicated to building a robust, reliable, and rapid mechanism for customers to monitor virtual machine (VM) health.

This article covers the use of Azure Resource Health to monitor Azure Virtual Machine availability. For a general overview of Flash solutions, see the Flash overview.

For documentation specific to the other solutions offered by Flash, choose from the following articles:

- Use Azure Resource Graph to monitor Azure Virtual Machine availability
- Use Event Grid system topics to monitor Azure Virtual Machine availability
- Use Azure Monitor to monitor Azure Virtual Machine availability

Azure Resource Health

It offers immediate and user-friendly health checks for individual resources through the portal. Customers can quickly access the resource health blade on the portal and also review a 30-day historical record of health checks, making it an excellent tool for fast and straightforward troubleshooting. The existing Azure Resource Health feature helps you to diagnose and get support for service problems that affect your Azure resources. It reports on the current and past health of your resources, showing any time ranges that each of your resources have been unavailable.

But we know that our customers and partners are interested in understanding what causes underlying technical issues, and in improving how they can receive communications about any issues—to feed into monitoring processes, to explain hiccups to other stakeholders, and ultimately to inform business decisions.

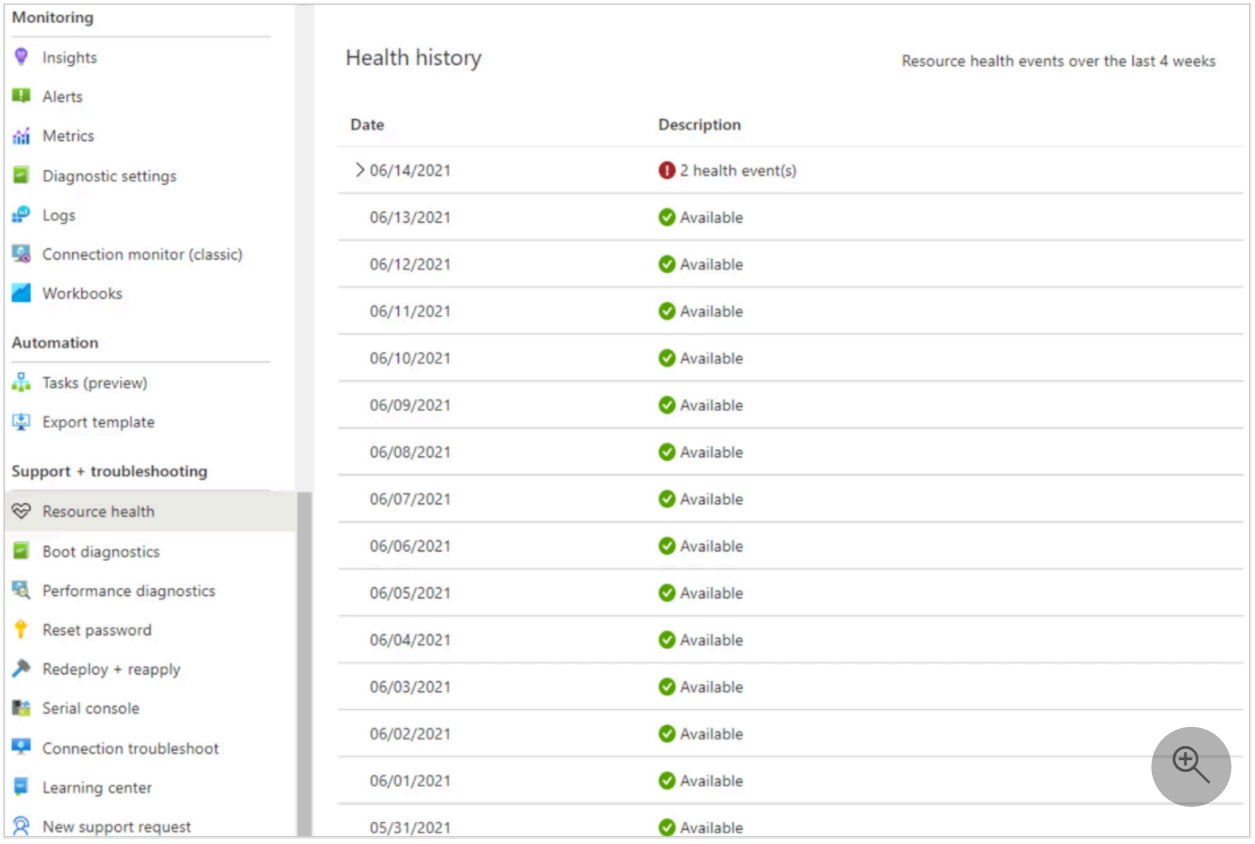
Root causes for VM issues in Azure Resource Health

We recently shipped an improvement to the resource health experience that will enhance the information we share with customers about VM failures and provide further context on the root cause that led to the issue. Now, in addition to getting a fast

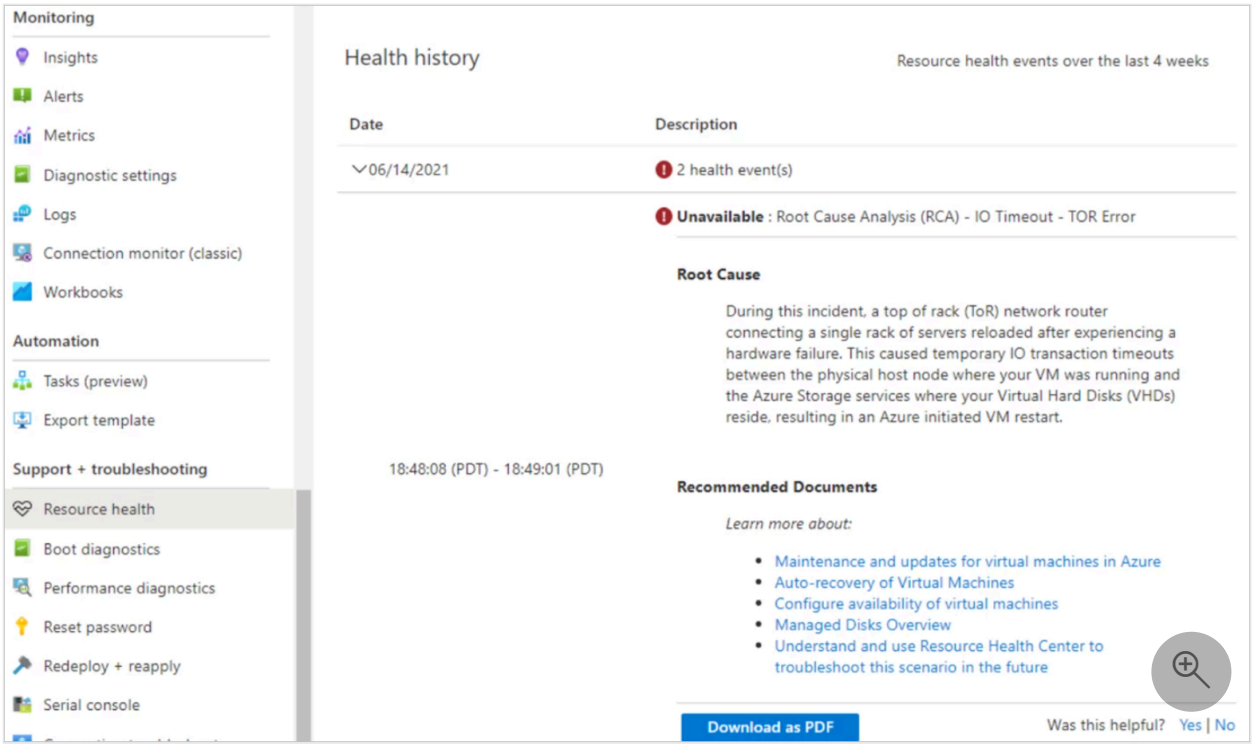
notification when a VM's availability is impacted, customers can expect a root cause to be added at a later point once our automated Root Cause Analysis (RCA) system identifies the failing Azure platform component that led to the VM failure. Let's walk through an example to see how this works process in practice:

At time T1, a server rack goes offline due to a networking issue, causing VMs on the rack to lose connectivity. Recent reliability improvements related to network architecture will be shared in a future Advancing Reliability blog post—watch this space!

At time T2, Azure's internal monitoring recognizes that it's unable to reach VMs on the rack and begins to mitigate by redeploying the impacted VMs to a new rack. During this time, an annotation is sent to resource health notifying customers that their VM is currently impacted and unavailable.



At time T3, platform telemetry from the top of rack switch, the host machine, and internal monitoring systems are correlated together in our RCA engine to derive the root cause of the failure. Once computed, the RCA is then published back into resource health along with relevant architectural resiliency recommendations that customers can implement to minimize the probability of impact in the future.



While the initial downtime notification functionality is several years old, the publishing of a root cause statement is a new addition. Now, let's dive into the details of how we derive these root causes.

Root Cause Analysis engine

Let's take a closer look at the prior example and walk through the details of how the RCA engine works and the technology behind it. At the core of our RCA engine for VMs is Azure Data Explorer (ADX), a big data service optimized for high volume log telemetry analytics. Azure Data Explorer enables the ability to easily parse terabytes of log telemetry from devices and services that comprise the Azure platform, join them together, and interpret the correlated information streams to derive a root cause for different failure scenarios. This ends up being a multistep data engineering process:

Phase 1: Detecting downtime

The first phase in root cause analysis is to define the trigger under which the analysis is executed. For Virtual Machines, we want to determine root causes whenever a VM unexpectedly reboots, so the trigger is a VM transitioning from an up state to a down state. Identifying these transitions from platform telemetry is straightforward in most scenarios, but more complicated around certain kinds of infrastructure failure where platform telemetry might get lost due to device failure or power loss. To handle these classes of failures, other techniques are required—like tracking data loss as a possible indication of a VM availability transition. Azure Data Explorer excels at this time of series analysis, and a more detailed look at techniques around this process can be found in the

Microsoft Tech Community: Calculating downtime using Window functions and Time Series functions in Azure Data Explorer.

Phase 2: Correlation analysis

Once a trigger event is defined (in this case, a VM transitioning to an unhealthy state) the next phase is correlation analysis. In this step, we use the presence of the trigger event to correlate telemetry from points across the Azure platform, like:

- Azure host: the physical blade hosting VMs.
- TOR: the top of rack network switch.
- Azure Storage: the service that hosts Virtual Disks for Azure Virtual Machines.

Each of these systems has their own telemetry feeds that need to get parsed and correlated with the VM downtime trigger event. This process is done through understanding the dependency graph for a VM and the underlying systems that can cause a VM to fail, and then joining all these dependent systems' health telemetry together, filtered on events that occurred close to the time of the VM transition. Azure Data Explorer's intuitive and powerful query language helps by offering documented patterns like time window join for correlating temporal telemetry streams together. At the end of this correlation process, we have a dataset that represents VM downtime transitions with correlated platform telemetry from all the dependent systems that could cause or could have information useful in determining what led to the VM failure.

Phase 3: Root cause attribution

The next step in the process is attribution. Now that we've collected all the relevant data together in a single dataset, attribution rules get applied to interpret the information and translate it into a customer-facing root cause statement. If you go back to our original example of a TOR failure, after our correlation analysis we might have many interesting pieces of information to interpret. For example, systems monitoring the Azure hosts might have logs indicating they lost connectivity to the hosts during this time. We might also have signals related to virtual disk connectivity problems, and explicit signals from the TOR device about the failure. All these pieces of information are now scanned over, and the explicit TOR failure signal is prioritized over the other signals as the root cause. This prioritization process, and the rules behind it, are constructed with domain experts and modified as the Azure platform evolves. Machine learning and anomaly detection mechanisms sit on top of these attributed root causes, to help identify opportunities to improve these classification rules and detect pattern changes in the rate of these failures to feed back into safe deployment pipelines.

Phase 4: RCA publishing

The last step is publishing root causes to Azure Resource Health, where they become visible to customers. Publishing is done by a simple Azure Functions application, which periodically queries the processed root cause data in Azure Data Explorer, and emits the results to the resource health backend. Because information streams can come in with various data delays, RCAs can occasionally be updated in this process to reflect better sources of information having arrived leading to a more specific root cause that what was originally published.

Going forward

Identifying and communicating the root cause of any issues to our customers and partners is just the beginning. Our customers may need to take these RCAs and share them with their customers and coworkers. We want to build on the work here to make it easier to identify and track resource RCAs, and easily share them out. To accomplish that, we're working on backend changes to generate unique per-resource and per-downtime tracking IDs that we can expose to you, so that you can easily match downtimes to their RCAs. We're also working on new features to make it easier to email RCAs out, and eventually subscribe to RCAs for your VMs. This feature will make it possible to sign up for RCAs directly in your inbox after an unavailability event with no further action needed on your part.

Next steps

To learn more about the solutions offered, proceed to corresponding solution article:

- Use Azure Resource Graph to monitor Azure Virtual Machine availability
- Use Event Grid system topics to monitor Azure Virtual Machine availability
- Use Azure Monitor to monitor Azure Virtual Machine availability

For a general overview of how to monitor Azure Virtual Machines, see Monitor Azure virtual machines and the Monitoring Azure virtual machines reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

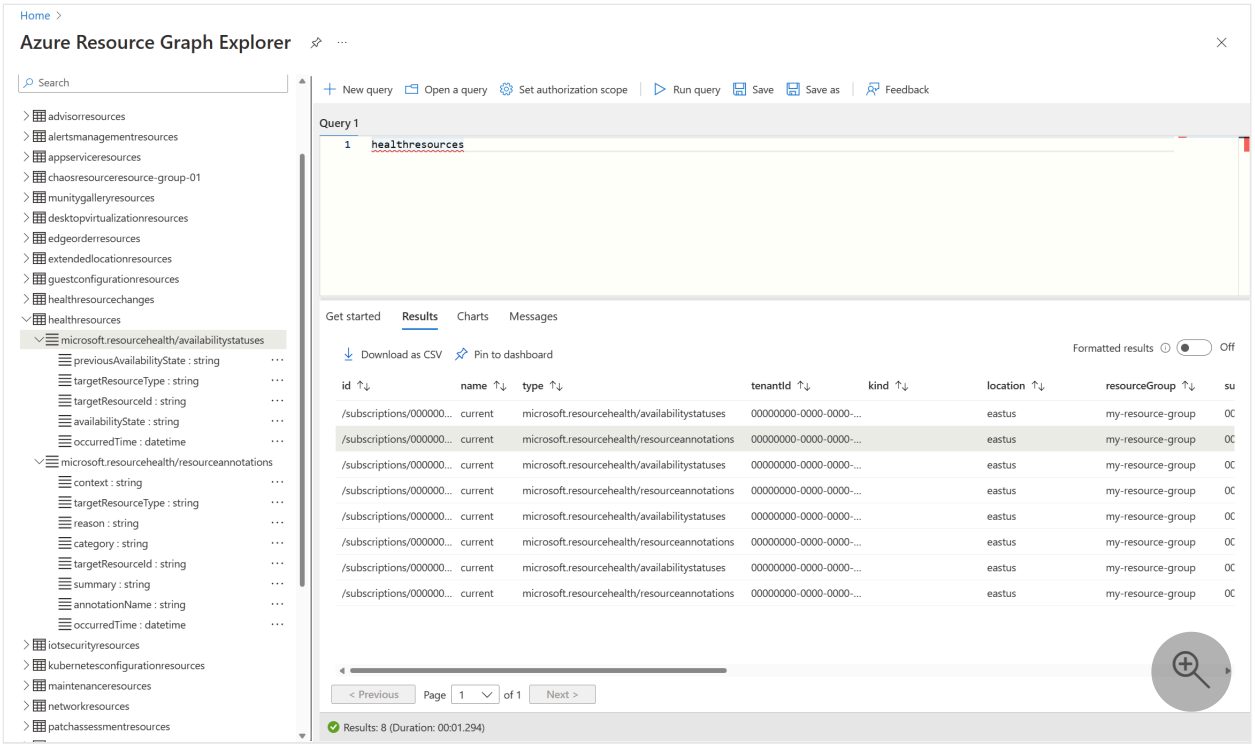
VM availability information in Azure Resource Graph

Article • 08/22/2024

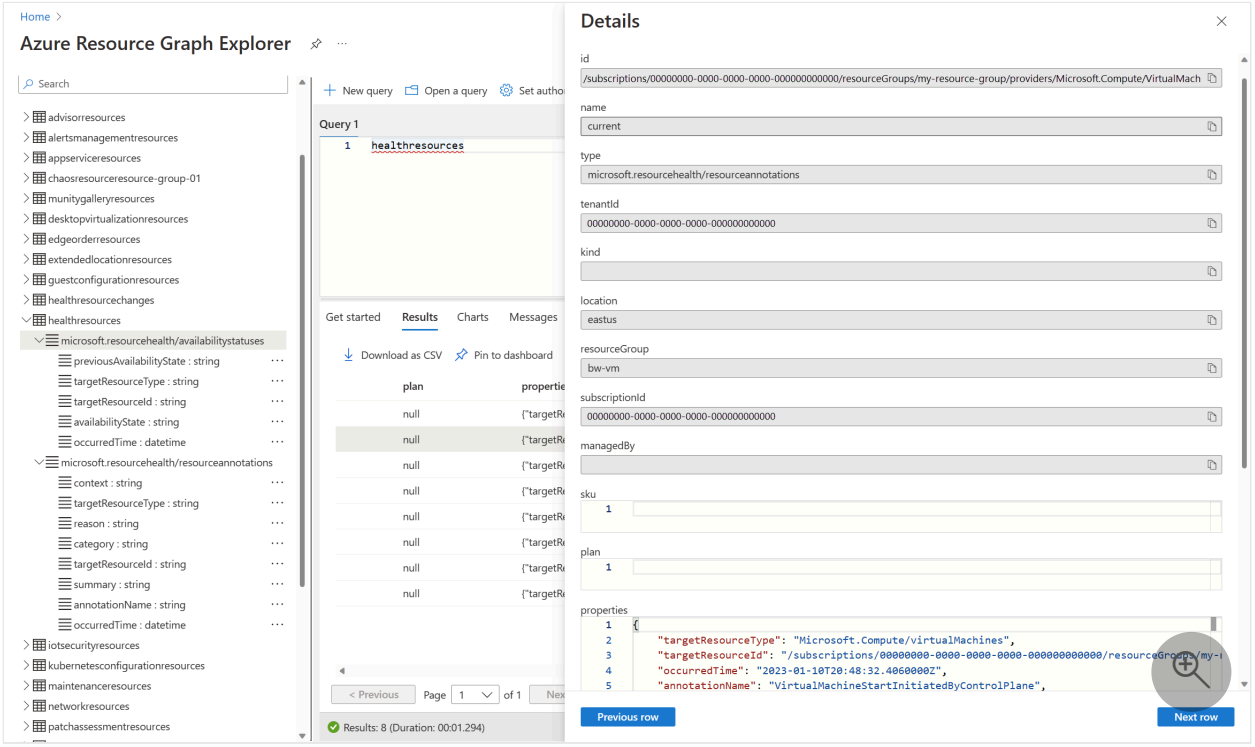
Azure Resource Graph is an Azure service that allows you to use the same KQL query language used in log queries to query your Azure resources at scale with complex filtering, grouping, and sorting by resource properties. You can use VM health annotations to Azure Resource Graph (ARG) for detailed failure attribution and downtime analysis including the following:

- Query the latest snapshot of VM availability together across all your Azure subscriptions.
- Assess the impact to business SLAs and trigger decisive mitigation actions, in response to disruptions and type of failure signature.
- Set up custom dashboards to supervise the comprehensive health of applications by joining VM availability information with additional resource metadata in Resource Graph.
- Track relevant changes in VM availability across a rolling 14 days window, by using the change tracking mechanism for conducting detailed investigations.

To get started with Resource Graph, open **Resource Graph Explorer** in the Azure portal. Select the **Table** tab and have a look at the microsoft.resourcehealth/availabilitystatuses and microsoft.resourcehealth/resourceannotations tables which are described below. Click on **healthresources** to create a simple query and then click **Run** to return the records.



To view the details for a record, scroll to the right and select **See details**.



There will be two types of events populated in the HealthResources table:

microsoft.resourcehealth/availabilitystatuses

This event denotes the latest availability status of a VM, based on the health checks performed by the underlying Azure platform. The availability states currently emitted for VMs are as follows:

- **Available**: The VM is up and running as expected.
- **Unavailable**: A disruption to the normal functioning of the VM has been detected.
- **Unknown**: The platform is unable to accurately detect the health of the VM. Check back in a few minutes.

The availability state is in the `properties` field of the record which includes the following properties:

Expand table

Field	Description
targetResourceType	Type of resource for which health data is flowing
targetResourceId	Resource ID
occurredTime	Timestamp when the latest availability state is emitted by the platform
previousAvailabilityState	Previous availability state of the VM
availabilityState	Current availability state of the VM

A sample `properties` value looks similar to the following:

JSON

```
{
    "targetResourceType": "Microsoft.Compute/virtualMachines",
    "targetResourceId":
"/subscriptions/<subscriptionId>/resourceGroups/<ResourceGroupName>/providers/Microsoft.Compute/virtualMachines/<VMName>",
    "occurredTime": "2022-10-11T11:13:59.9570000Z",
    "previousAvailabilityState": "Available",
    "availabilityState": "Unavailable"
}
```

microsoft.resourcehealth/resourceannotations

This event contextualizes any changes to VM availability, by detailing necessary failure attributes to help you investigate and mitigate the disruption as needed. The full list of VM health annotations are listed at [Resource Health virtual machine Health Annotations] (../service-health/resource-health-vm-annotation.md).

These annotations can be broadly classified into the following:

- **Downtime Annotations**: Emitted when the platform detects VM availability transitioning to Unavailable. Examples include host crashes or reboot operations.
- **Informational Annotations**: Emitted during control plane activities with no impact to VM availability. Examples include VM allocation, stop, delete, start. Usually, no additional customer action is required in response.
- **Degraded Annotations**: Emitted when VM availability is detected to be at risk. Examples include when failure prediction models predict a degraded hardware component that can cause the VM to reboot at any given time. You should redeploy by the deadline specified in the annotation message to avoid any unanticipated loss of data or downtime.

Expand table

Field	Description
targetResourceType	Type of resource for which health data is flowing
targetResourceId	Resource ID
occurredTime	Timestamp when the latest availability state is emitted by the platform
annotationName	Name of the Annotation emitted
reason	Brief overview of the availability impact observed by the customer
category	Denotes whether the platform activity triggering the annotation was either planned maintenance or unplanned repair. This field is not applicable to customer/VM-initiated events. Possible values: Planned | Unplanned | Not Applicable | Null
context	Denotes whether the activity triggering the annotation was due to an authorized user or process (customer initiated), due to the Azure platform (platform initiated), or due to activity in the guest OS that has resulted in availability impact (VM initiated). Possible values: Platform-Initiated | User-initiated | VM-initiated | Not Applicable | Null
summary	Statement detailing the cause for annotation emission, along with remediation steps that can be taken by users

See Azure Resource Graph sample queries by table for sample queries using this data.

Next steps

- Create alerts from collected data

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Understanding Azure virtual machine usage

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

By analyzing your Azure usage data, powerful consumption insights can be gained – insights that can enable better cost management and allocation throughout your organization. This document provides a deep dive into your Azure Compute consumption details. For more details on general Azure usage, navigate to Understanding your bill.

Download your usage details

To begin, download your usage details. The table provides the definition and example values of usage for Virtual Machines deployed via the Azure Resource Manager. This document doesn't contain detailed information for VMs deployed via our classic model.

Expand table

Field	Meaning	Example Values
Usage Date	The date when the resource was used	`11/23/2017`
Meter ID	Identifies the top-level service for which this usage belongs to	`Virtual Machines`
Meter Sub-Category	The billed meter identifier. For Compute Hour usage, there's a meter for each VM Size + OS (Windows, Non-Windows) + Region. For Premium software usage, there's a meter for each software type. Most premium software images have different meters for each core size. For more information, visit the Compute Pricing Page	`2005544f-659d-49c9-9094-8e0aea1be3a5`
Meter Name	This value is specific for each service in	`Compute Hours`

Field	Meaning	Example Values
	Azure. For compute, it is always "Compute Hours".	
Meter Region	Identifies the location of the datacenter for certain services that are priced based on datacenter location.	`JA East`
Unit	Identifies the unit that the service is charged in. Compute resources are billed per hour.	`Hours`
Consumed	The amount of the resource that has been consumed for that day. For Compute, we bill for each minute the VM ran for a given hour (up to six decimals of accuracy).	`1, 0.5`
Resource Location	Identifies the datacenter where the resource is running.	`JA East`
Consumed Service	The Azure platform service that you used.	`Microsoft.Compute`
Resource Group	The resource group in which the deployed resource is running in. For more information, see Azure Resource Manager overview.	`MyRG`

Instance ID	The identifier for the resource. The identifier contains the name you specify for the resource when it was created. For VMs, the Instance ID contains the SubscriptionId, ResourceGroupName, and VM Name (or scale set name for scale set usage).	`/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/MyRG/providers/Microsoft.Compute/virtualMachines/MyVM1`
		or
		`/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/MyRG/providers/Microsoft.Compute/virtualMachineScaleSets/MyVMSS1`

Field	Meaning	Example Values
Tags	Tag you assign to the resource. Use tags to group billing records. Learn how to tag your Virtual Machines using the CLI or PowerShell This is available for Resource Manager VMs only.	`{"myDepartment":"RD","myUser":"myName"}`

More Info	Service-specific metadata. For VMs, we populate the following data in the additional info field:
	
	Image Type- specific image that you ran. Find the full list of supported strings under Image Types.
	
	Service Type: the size that you deployed.
	
	VM Name: name of your VM. This field is only populated for scale set VMs. If you need your VM Name for scale set VMs, you can find that in the Instance ID string.
	
	UsageType: Specifies the type of usage.
	
	ComputeHR is the Compute Hour usage for the underlying VM, like Standard_D1_v2.
	
	ComputeHR_SW is the premium software charge if the VM is using premium software.
		Virtual Machines
		`{"ImageType":"Canonical","ServiceType":"Standard_DS1_v2","VM Name":"", "UsageType":"ComputeHR"}`
		
		Virtual Machine Scale Sets
		`{"ImageType":"Canonical","ServiceType":"Standard_DS1_v2","VM Name":"myVM1", "UsageType":"ComputeHR"}`
		
		Premium Software
		`{"ImageType":"","ServiceType":"Standard_DS1_v2","VM Name":"", "UsageType":"ComputeHR_SW"}`

Image Type

For some images in the Azure gallery, the image type is populated in the Additional Info field. This enables users to understand and track what they have deployed on their Virtual Machine. The following values that are populated in this field based on the image you have deployed:

- BitRock
- Canonical FreeBSD
- Open Logic
- Oracle
- SLES for SAP
- SQL Server 14 Preview on Windows Server 2012 R2 Preview
- SUSE
- SUSE Premium
- StorSimple Cloud Appliance
- Red Hat
- Red Hat for SAP Business Applications
- Red Hat for SAP HANA
- Windows Client BYOL
- Windows Server BYOL
- Windows Server Preview

Service Type

The service type field in the Additional Info field corresponds to the exact VM size you deployed. Premium storage VMs (SSD-based) and nonpremium storage VMs (HDD-based) are priced the same. If you deploy an SSD-based size, like Standard_DS2_v2, you see the non-SSD size (`Standard\_D2\_v2 VM`) in the Meter Sub-Category column and the SSD-size (`Standard\_DS2\_v2`) in the Additional Info field.

Region Names

The region name populated in the Resource Location field in the usage details varies from the region name used in the Azure Resource Manager. Here's a mapping between the region values:

Expand table

Resource Manager Region Name	Resource Location in Usage Details
australiaeast	AU East
australiasoutheast	AU Southeast
brazilsouth	BR South
CanadaCentral	CA Central
CanadaEast	CA East
CentralIndia	IN Central

Resource Manager Region Name	Resource Location in Usage Details
centralus	Central US
chinaeast	China East
chinanorth	China North
eastasia	East Asia
eastus	East US
eastus2	East US 2
GermanyCentral	DE Central
GermanyNortheast	DE Northeast
japaneast	JA East
japanwest	JA West
KoreaCentral	KR Central
KoreaSouth	KR South
northcentralus	North Central US
northeurope	North Europe
southcentralus	South Central US
southeastasia	Southeast Asia
SouthIndia	IN South
UKNorth	US North
uksouth	UK South
UKSouth2	UK South 2
ukwest	UK West
USDoDCentral	US DoD Central
USDoDEast	US DoD East
USGovArizona	USGov Arizona
usgoviowa	USGov Iowa
USGovTexas	USGov Texas
usgovvirginia	USGov Virginia
westcentralus	US West Central
westeurope	West Europe
WestIndia	IN West

Resource Manager Region Name	Resource Location in Usage Details
westus	West US
westus2	US West 2

Virtual machine usage FAQ

What resources are charged when deploying a VM?

VMs acquire costs for the VM itself, any premium software running on the VM, the storage account\managed disk associated with the VM, and the networking bandwidth transfers from the VM.

How can I tell if a VM is using Azure Hybrid Benefit in the Usage CSV?

If you deploy using the Azure Hybrid Benefit, you're charged the Non-Windows VM rate since you're bringing your own license to the cloud. In your bill, you can distinguish which Resource Manager VMs are running Azure Hybrid Benefit because they have either “Windows_Server BYOL” or “Windows_Client BYOL” in the ImageType column.

How are Basic vs. Standard VM Types differentiated in the Usage CSV?

Both Basic and Standard A-Series VMs are offered. If you deploy a Basic VM, in the Meter Sub Category, it has the string “Basic.” If you deploy a Standard A-Series VM, then the VM size appears as “A1 VM” since Standard is the default. To learn more about the differences between Basic and Standard, see the Pricing Page.

What are ExtraSmall, Small, Medium, Large, and ExtraLarge sizes?

ExtraSmall - ExtraLarge are the legacy names for Standard_A0 – Standard_A4. In classic VM usage records, you might see this convention used if you have deployed these sizes.

What is the difference between Meter Region and Resource Location?

The Meter Region is associated with the meter. For some Azure services who use one price for all regions, the Meter Region field could be blank. However, since VMs have dedicated prices per region for Virtual Machines, this field is populated. Similarly, the Resource Location for Virtual Machines is the location where the VM is deployed. The Azure regions in both fields are the same, although they might have a different string convention for the region name.

Why is the ImageType value blank in the Additional Info field?

The ImageType field is only populated for a subset of images. If you didn't deploy one of the listed images, the ImageType is blank.

Why is the VMName blank in the Additional Info?

The VMName is only populated in the Additional Info field for VMs in a scale set. The InstanceID field contains the VM name for nonscale set VMs.

What does ComputeHR mean in the UsageType field in the Additional Info?

ComputeHR stands for 'Compute Hour', which represents the usage event for the underlying infrastructure cost. If the UsageType is ComputeHR_SW, the usage event represents the premium software charge for the VM.

How do I know if I'm charged for premium software?

When exploring which VM Image best fits your needs, be sure to check out Azure Marketplace. The image has the software plan rate. If you see “Free” for the rate, there's no extra cost for the software.

What is the difference between 'Microsoft.ClassicCompute' and 'Microsoft.Compute' in the Consumed service?

Microsoft.ClassicCompute represents classic resources deployed via the Azure Service Manager. If you deploy via the Resource Manager, then Microsoft.Compute is populated in the consumed service. Learn more about the Azure Deployment models.

Why is the InstanceID field blank for my Virtual Machine usage?

If you deploy via the classic deployment model, the InstanceID string isn't available.

Why are the tags for my VMs not flowing to the usage details?

Tags flow to the Usage CSV for Resource Manager VMs only. Classic resource tags aren't available in the usage details.

How can the consumed quantity be more than 24 hours one day?

In the Classic model, billing for resources is aggregated at the Cloud Service level. If you have more than one VM in a Cloud Service that uses the same billing meter, your usage is aggregated together. VMs deployed via Resource Manager are billed at the VM level, so this aggregation won't apply.

Why is pricing not available for DS/FS/GS/LS sizes on the pricing page?

Premium storage capable VMs are billed at the same rate as nonpremium storage capable VMs. Only your storage costs differ. Visit the storage pricing page for more information.

Next steps

To learn more about your usage details, see Understand your bill for Microsoft Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Use tags to organize your Azure resources and management hierarchy

09/15/2025

Tags are metadata elements that you apply to your Azure resources. They are key-value pairs that help you identify resources based on settings that are relevant to your organization. If you want to track the deployment environment for your resources, add a key named `Environment`. To identify the resources deployed to production, give them a value of `Production`. The full key-value pair is `Environment = Production`.

This article describes the conditions and limitations for using tags.

Tag usage and recommendations

You can apply tags to your Azure resources, resource groups, and subscriptions but not to management groups.

For recommendations on how to implement a tagging strategy, see Resource naming and tagging decision guide.

Resource tags support all cost-accruing services. To ensure that cost-accruing services are provisioned with a tag, use one of the tag policies.

> **⚠ Warning**
>
> Tags are stored as plain text. Do not add sensitive values to tags. Sensitive values could be exposed through many methods, including cost reports, commands that return existing tag definitions, deployment histories, exported templates, and monitoring logs.

> **⚠ Warning**
>
> Be careful when using non-English language in your tags. It can cause decoding progress failure while loading your virtual machine's metadata from IMDS (Instance Metadata Service).

> **ⓘ Important**
>
> Tag names are case-insensitive for operations. An operation updates or retrieves a tag with a tag name, regardless of the casing. However, the resource provider might keep the

casing you provide for the tag name. You see that casing in cost reports.

Tag values are case-sensitive.

> **Note**
>
> This article provides steps about how to delete personal data from the device or service and can be used to support your obligations under the GDPR. For general information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Required access

There are two ways to get the required access to tag resources:

- You can have write access to the `Microsoft.Resources/tags` resource type. This access lets you tag any resource, even if you don't have access to the resource itself. The Tag Contributor role grants this access. For example, this role can't apply tags to resources or resource groups through the Azure portal. However, it can apply tags to subscriptions through the Azure portal. It supports all tag operations through Azure PowerShell and REST API.
- You can have write access to the resource itself. The Contributor role grants the required access to apply tags to any entity. To apply tags to only one resource type, use the Contributor role for that resource. To apply tags to virtual machines, use the Virtual Machine Contributor role.

Inherit tags

Resources don't inherit the tags you apply to a resource group or a subscription. To apply tags from a subscription or resource group to the resources, see Assign policy definitions for tag compliance.

You can group costs for an Azure resource by using the **cm-resource-parent** tag. This tag lets you review tagged costs in Microsoft Cost Management without having to use filters. The key for this tag is `cm-resource-parent` and its value is the resource ID of the Azure resource you want to group costs by. For example, to group costs by an Azure Virtual Desktop host pool, provide the resource ID of the host pool. For more information, see Group related resources in the Resources view.

Tags and billing

Use tags to group your billing data. For example, if you're running multiple virtual machines for different organizations, use the tags to group usage by cost center. You can also use tags to categorize costs by runtime environment, including the billing usage for virtual machines running in the production environment.

To retrieve information about tags, download the usage file from the Azure portal. For more information, see Download or view your Azure billing invoice. For services that support tags with billing, the tags appear in the **Tags** column.

For REST API operations, see the Azure Billing REST API overview.

Unique tags pagination

When you call the Unique tags API, there's a limit to the size of each API response page. A tag that has a large set of unique values requires the API to fetch the next page to retrieve the remaining set of values. When the results are divided over multiple pages, the API response shows the tag key again to indicate that the values are still under this key.

This behavior can cause some tools like the Azure portal to show the tag key twice.

Limitations

The following limitations apply to tags:

- Not all resource types support tags. To determine if you can apply a tag to a resource type, see Tag support for Azure resources.
- Each resource type might have specific requirements when working with tags. For example, you can only update tags on virtual machine extensions when the virtual machine is running. If you receive an error message while trying to update a tag, follow the instructions in the message.
- Each resource, resource group, and subscription can have a maximum of 50 tag name-value pairs. If you need to apply more tags than the maximum allowed number, use a JSON string for the tag value. The JSON string can contain many of the values that you apply to a single tag name. A resource group or subscription can contain many resources that each have 50 tag name-value pairs.
- The tag name has a limit of 512 characters and the tag value has a limit of 256 characters. For storage accounts, the tag name has a limit of 128 characters and the tag value has a

limit of 256 characters.

- Classic resources such as Cloud Services don't support tags.
- Azure IP Groups and Azure Firewall policies don't support PATCH operations. Therefore, PATCH API method operations can't update tags through the Azure portal. Instead, use the update commands for those resources. For example, you can update tags for an IP group with the az network ip-group update command.
- Tag names can't contain these characters: `<`, `>`, `%`, `&`, `\`, `?`, `/`

  > **Note**
  >
  > - Azure DNS zones don't support the use of spaces or parentheses in the tag or a tag that starts with a number. Azure DNS tag names don't support special and Unicode characters. The value can contain all characters.
  > - Traffic Manager doesn't support the use of spaces, `#`, or `:` in the tag name. The tag name can't start with a number.
  > - Azure Front Door doesn't support the use of `#` or `:` in the tag name.
  > - The following Azure resources only support 15 tags:
  >   - Azure Automation
  >   - Azure Content Delivery Network
  >   - Azure Public DNS (Zone and A records)
  >   - Azure Private DNS (Zone and A records)
  >   - Azure Log Analytics saved search

Next steps

For more information on how to work with tags, see:

- Azure portal
- Azure CLI
- Azure PowerShell
- Python
- ARM templates
- Bicep

Azure Instance Metadata Service

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

The Azure Instance Metadata Service (IMDS) provides information about currently running virtual machine instances. You can use it to manage and configure your virtual machines. This information includes the SKU, storage, network configurations, and upcoming maintenance events. For a complete list of the data available, see the Endpoint Categories Summary.

IMDS is available for running instances of virtual machines (VMs) and scale set instances. All endpoints support VMs created and managed by using Azure Resource Manager. Only the Attested category and Network portion of the Instance category support VMs created by using the classic deployment model. The Attested endpoint does so only to a limited extent.

IMDS is a REST API that's available at a well-known, non-routable IP address (`169.254.169.254`). You can only access it from within the VM. Communication between the VM and IMDS never leaves the host. Have your HTTP clients bypass web proxies within the VM when querying IMDS.

Usage

Access Azure Instance Metadata Service

To access IMDS, create a VM from Azure Resource Manager or the Azure portal, and use the following samples. For more examples, see Azure Instance Metadata Samples.

Here's sample code to retrieve all metadata for an instance. To access a specific data source, see Endpoint Categories for an overview of all available features.

Request

> **Important**
>
> This example bypasses proxies. You **must** bypass proxies when querying IMDS. See Proxies for additional information.

> **Note**
>
> IMDS requests must be sent using the VM's primary NIC and primary IP, and DHCP must be enabled.

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/instance?api-version=2025-04-07" | ConvertTo-Json -Depth 64
```

`-NoProxy` requires PowerShell V6 or greater. See our samples repository for examples with older PowerShell versions.

Response

> **Note**
>
> The response is a JSON string. The following example response is formatted for readability.

Windows

JSON


```
{
    "compute": {
        "azEnvironment": "AZUREPUBLICCLOUD",
        "additionalCapabilities": {
            "hibernationEnabled": "true"
        },
        "hostGroup": {
          "id": "testHostGroupId"
        },
        "extendedLocation": {
            "type": "edgeZone",
            "name": "microsoftlosangeles"
        },
        "evictionPolicy": "",
        "isHostCompatibilityLayerVm": "true",
        "licenseType":  "Windows_Client",
        "location": "westus",
        "name": "examplevmname",
        "offer": "WindowsServer",
        "osProfile": {
            "adminUsername": "admin",
            "computerName": "examplevmname",
            "disablePasswordAuthentication": "true"
```

```
        },
        "osType": "Windows",
        "placementGroupId": "f67c14ab-e92c-408c-ae2d-da15866ec79a",
        "plan": {
            "name": "planName",
            "product": "planProduct",
            "publisher": "planPublisher"
        },
        "platformFaultDomain": "36",
        "platformSubFaultDomain": "",
        "platformUpdateDomain": "42",
        "priority": "Regular",
        "publicKeys": [{
                "keyData": "ssh-rsa 0",
                "path": "/home/user/.ssh/authorized_keys0"
            },
            {
                "keyData": "ssh-rsa 1",
                "path": "/home/user/.ssh/authorized_keys1"
            }
        ],
        "publisher": "RDFE-Test-Microsoft-Windows-Server-Group",
        "resourceGroupName": "macikgo-test-may-23",
        "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/virtualMachines/examplevmname",
        "securityProfile": {
            "secureBootEnabled": "true",
            "virtualTpmEnabled": "false",
            "encryptionAtHost": "true",
            "securityType": "TrustedLaunch"
        },
        "sku": "2019-Datacenter",
        "storageProfile": {
            "dataDisks": [{
                "bytesPerSecondThrottle": "979202048",
                "caching": "None",
                "createOption": "Empty",
                "diskCapacityBytes": "274877906944",
                "diskSizeGB": "1024",
                "image": {
                  "uri": ""
                },
                "isSharedDisk": "false",
                "isUltraDisk": "true",
                "lun": "0",
                "managedDisk": {
                  "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampledatadiskname",
                  "storageAccountType": "StandardSSD_LRS"
                },
                "name": "exampledatadiskname",
                "opsPerSecondThrottle": "65280",
                "vhd": {
```

```
                "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        }],
        "imageReference": {
            "id": "",
            "offer": "WindowsServer",
            "publisher": "MicrosoftWindowsServer",
            "sku": "2019-Datacenter",
            "version": "latest",
            "communityGalleryImageId":
"/CommunityGalleries/testgallery/Images/1804Gen2/Versions/latest",
            "sharedGalleryImageId":
"/SharedGalleries/1P/Images/gen2/Versions/latest",
            "exactVersion": "1.1686127202.30113"
        },
        "osDisk": {
            "caching": "ReadWrite",
            "createOption": "FromImage",
            "diskSizeGB": "30",
            "diffDiskSettings": {
                "option": "Local"
            },
            "encryptionSettings": {
              "enabled": "false",
              "diskEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-source-
guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/test-kv"
                },
                "secretUrl": "https://test-
disk.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxx"
              },
              "keyEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-key-
guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/test-kv"
                },
                "keyUrl": "https://test-key.vault.azure.net/secrets/xxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
              }
            },
            "image": {
                "uri": ""
            },
            "managedDisk": {
                "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampleosdiskname",
                "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampleosdiskname",
            "osType": "Windows",
            "vhd": {
```

```
                    "uri": ""
                },
                "writeAcceleratorEnabled": "false"
            },
            "resourceDisk": {
                "size": "4096"
            }
        },
        "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
        "tags": "baz:bash;foo:bar",
        "userData": "Zm9vYmFy",
        "version": "15.05.22",
        "virtualMachineScaleSet": {
            "id": "/subscriptions/xxxxxxxx-xxxxx-xxx-xxx-
xxxx/resourceGroups/resource-group-
name/providers/Microsoft.Compute/virtualMachineScaleSets/virtual-machine-scale-
set-name"
        },
        "vmId": "02aab8a4-74ef-476e-8182-f6d2ba4166a6",
        "vmScaleSetName": "crpteste9vflji9",
        "vmSize": "Standard_A3",
        "zone": ""
    },
    "network": {
        "interface": [{
            "ipv4": {
               "ipAddress": [{
                    "privateIpAddress": "10.144.133.132",
                    "publicIpAddress": ""
                }],
                "subnet": [{
                    "address": "10.144.133.128",
                    "prefix": "26"
                }]
            },
            "ipv6": {
                "ipAddress": [
                 ]
            },
            "macAddress": "0011AAFFBB22"
        }]
    }
}
```


Security and authentication

The Instance Metadata Service is only accessible from within a running virtual machine instance on a non-routable IP address. VMs can only interact with their own metadata/functionality. The API is HTTP only and never leaves the host.

In order to ensure that requests are directly intended for IMDS and prevent unintended or unwanted redirection of requests, requests:

- **Must** contain the header `Metadata: true`
- Must **not** contain an `X-Forwarded-For` header

Any request that doesn't meet **both** of these requirements are rejected by the service.

> **Important**
>
> IMDS is **not** a channel for sensitive data. The API is unauthenticated and open to all processes on the VM. Information exposed through this service should be considered as shared information to all applications running inside the VM.

If it isn't necessary for every process on the VM to access IMDS endpoint, you can set local firewall rules to limit the access. For example, if only a known system service needs to access instance metadata service, you can set a firewall rule on IMDS endpoint, only allowing the specific process(es) to access, or denying access for the rest of the processes.

Proxies

IMDS is **not** intended to be used behind a proxy and doing so is unsupported. Most HTTP clients provide an option for you to disable proxies on your requests, and this functionality must be utilized when communicating with IMDS. Consult your client's documentation for details.

> **Important**
>
> Even if you don't know of any proxy configuration in your environment, **you still must override any default client proxy settings**. Proxy configurations can be automatically discovered, and failing to bypass such configurations exposes you to outage risks should the machine's configuration be changed in the future.

Rate limiting

In general, requests to IMDS are limited to 5 requests per second (on a per VM basis). Requests exceeding this threshold will be rejected with 429 responses. Requests to the Managed Identity category are limited to 20 requests per second and 5 concurrent requests (on a per VM basis).

HTTP verbs

The following HTTP verbs are currently supported:

Expand table

Verb	Description
`GET`	Retrieve the requested resource

Parameters

Endpoints may support required and/or optional parameters. See Schema and the documentation for the specific endpoint in question for details.

Query parameters

IMDS endpoints support HTTP query string parameters. For example:

URL

```
http://169.254.169.254/metadata/instance/compute?api-version=2025-04-07&format=json
```

Specifies the parameters:

Expand table

Name	Value
`api-version`	`2025-04-07`
`format`	`json`

Requests with duplicate query parameter names will be rejected.

Route parameters

For some endpoints that return larger json blobs, we support appending route parameters to the request endpoint to filter down to a subset of the response:

URL

```
http://169.254.169.254/metadata/<endpoint>/[<filter parameter>/...]?<query
parameters>
```

The parameters correspond to the indexes/keys that would be used to walk down the json object were you interacting with a parsed representation.

For example, `/metadata/instance` returns the json object:

JSON


```
{
    "compute": { ... },
    "network": {
        "interface": [
            {
                "ipv4": {
                   "ipAddress": [{
                       "privateIpAddress": "10.144.133.132",
                       "publicIpAddress": ""
                    }],
                    "subnet": [{
                        "address": "10.144.133.128",
                        "prefix": "26"
                    }]
                },
                "ipv6": {
                    "ipAddress": [{
                        "privateIpAddress":
"b4bc:8fce:f33b:4990:cced:d94e:ab4f:6ea0"
                    }]
                },
                "macAddress": "0011AAFFBB22"
            },
            ...
        ]
    }
}
```


If we want to filter the response down to just the compute property, we would send the request:

URL

```
http://169.254.169.254/metadata/instance/compute?api-version=<version>
```

Similarly, if we want to filter to a nested property or specific array element we keep appending keys:

URL

```
http://169.254.169.254/metadata/instance/network/interface/0?api-version=<version>
```

would filter to the first element from the `Network.interface` property and return:

JSON


```
{
    "ipv4": {
       "ipAddress": [{
            "privateIpAddress": "10.144.133.132",
            "publicIpAddress": ""
        }],
        "subnet": [{
            "address": "10.144.133.128",
            "prefix": "26"
        }]
    },
    "ipv6": {
        "ipAddress": [{
            "privateIpAddress": "b4bc:8fce:f33b:4990:cced:d94e:ab4f:6ea0"
        }]
    },
    "macAddress": "0011AAFFBB22"
}
```


> **Note**
>
> When filtering to a leaf node, `format=json` doesn't work. For these queries `format=text` needs to be explicitly specified since the default format is json.

Schema

Data format

By default, IMDS returns data in JSON format (`Content-Type: application/json`). However, endpoints that support response filtering (see Route Parameters) also support the format `text`.

To access a non-default response format, specify the requested format as a query string parameter in the request. For example:

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance?api-version=2025-04-07&format=text"
```

In json responses, all primitives will be of type `string`, and missing or inapplicable values are always included but will be set to an empty string.

Versioning

IMDS is versioned and specifying the API version in the HTTP request is mandatory. The only exception to this requirement is the versions endpoint, which can be used to dynamically retrieve the available API versions.

As newer versions are added, older versions can still be accessed for compatibility if your scripts have dependencies on specific data formats.

When you don't specify a version, you get an error with a list of the newest supported versions:

JSON

```
{
    "error": "Bad request. api-version is invalid or was not specified in the
request. For more information refer to aka.ms/azureimds",
    "newest-versions": [
        "2025-04-07",
        "2024-07-17",
        "2024-03-15"
    ]
}
```

Supported API versions

- 2025-04-07
- 2024-07-17
- 2024-03-15
- 2023-11-15
- 2023-07-01
- 2021-12-13
- 2021-11-15
- 2021-11-01
- 2021-10-01
- 2021-08-01
- 2021-05-01
- 2021-03-01

- 2021-02-01
- 2021-01-01
- 2020-12-01
- 2020-10-01
- 2020-09-01
- 2020-07-15
- 2020-06-01
- 2019-11-01
- 2019-08-15
- 2019-08-01
- 2019-06-04
- 2019-06-01
- 2019-04-30
- 2019-03-11
- 2019-02-01
- 2018-10-01
- 2018-04-02
- 2018-02-01
- 2017-12-01
- 2017-10-01
- 2017-08-01
- 2017-04-02
- 2017-03-01

Swagger

A full Swagger definition for IMDS is available at: https://github.com/Azure/azure-rest-api-specs/blob/main/specification/imds/data-plane/InstanceMetadataService/readme.md

Regional availability

The service is **generally available** in all Azure clouds.

Root endpoint

The root endpoint is `http://169.254.169.254/metadata`.

Endpoint categories

The IMDS API contains multiple endpoint categories representing different data sources, each of which contains one or more endpoints. See each category for details.

Expand table

Category root	Description	Version introduced
`/metadata/attested`	See Attested Data	2018-10-01
`/metadata/identity`	See Managed Identity via IMDS	2018-02-01
`/metadata/instance`	See Instance Metadata	2017-04-02
`/metadata/loadbalancer`	See Retrieve Load Balancer metadata via IMDS	2020-10-01
`/metadata/scheduledevents`	See Scheduled Events via IMDS	2017-08-01
`/metadata/versions`	See Versions	N/A

Versions

List API versions

Returns the set of supported API versions.

```
GET /metadata/versions
```

Parameters

None (this endpoint is unversioned).

Response

JSON

```
{
  "apiVersions": [
    "2017-03-01",
    "2017-04-02",
    ...
  ]
}
```

Instance metadata

Get VM metadata

Exposes the important metadata for the VM instance, including compute, network, and storage.

```
GET /metadata/instance
```

Parameters

Expand table

Name	Required/Optional	Description
`api-version`	Required	The version used to service the request.
`format`	Optional*	The format (`json` or `text`) of the response. *Note: May be required when using request parameters

This endpoint supports response filtering via route parameters.

Response

Windows

JSON


```
{
    "compute": {
        "azEnvironment": "AZUREPUBLICCLOUD",
        "additionalCapabilities": {
            "hibernationEnabled": "true"
        },
        "hostGroup": {
          "id": "testHostGroupId"
        },
        "extendedLocation": {
            "type": "edgeZone",
            "name": "microsoftlosangeles"
        },
```

```
        "evictionPolicy": "",
        "isHostCompatibilityLayerVm": "true",
        "licenseType":  "Windows_Client",
        "location": "westus",
        "name": "examplevmname",
        "offer": "WindowsServer",
        "osProfile": {
            "adminUsername": "admin",
            "computerName": "examplevmname",
            "disablePasswordAuthentication": "true"
        },
        "osType": "Windows",
        "placementGroupId": "f67c14ab-e92c-408c-ae2d-da15866ec79a",
        "plan": {
            "name": "planName",
            "product": "planProduct",
            "publisher": "planPublisher"
        },
        "platformFaultDomain": "36",
        "platformSubFaultDomain": "",
        "platformUpdateDomain": "42",
        "priority": "Regular",
        "publicKeys": [{
                "keyData": "ssh-rsa 0",
                "path": "/home/user/.ssh/authorized_keys0"
            },
            {
                "keyData": "ssh-rsa 1",
                "path": "/home/user/.ssh/authorized_keys1"
            }
        ],
        "publisher": "RDFE-Test-Microsoft-Windows-Server-Group",
        "resourceGroupName": "macikgo-test-may-23",
        "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/virtualMachines/examplevmname",
        "securityProfile": {
            "secureBootEnabled": "true",
            "virtualTpmEnabled": "false",
            "encryptionAtHost": "true",
            "securityType": "TrustedLaunch"
        },
        "sku": "2019-Datacenter",
        "storageProfile": {
            "dataDisks": [{
                "bytesPerSecondThrottle": "979202048",
                "caching": "None",
                "createOption": "Empty",
                "diskCapacityBytes": "274877906944",
                "diskSizeGB": "1024",
                "image": {
                  "uri": ""
                },
                "isSharedDisk": "false",
                "isUltraDisk": "true",
```

```
                "lun": "0",
                "managedDisk": {
                  "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampledatadiskname",
                  "storageAccountType": "StandardSSD_LRS"
                },
                "name": "exampledatadiskname",
                "opsPerSecondThrottle": "65280",
                "vhd": {
                  "uri": ""
                },
                "writeAcceleratorEnabled": "false"
            }],
            "imageReference": {
                "id": "",
                "offer": "WindowsServer",
                "publisher": "MicrosoftWindowsServer",
                "sku": "2019-Datacenter",
                "version": "latest",
                "communityGalleryImageId":
"/CommunityGalleries/testgallery/Images/1804Gen2/Versions/latest",
                "sharedGalleryImageId":
"/SharedGalleries/1P/Images/gen2/Versions/latest",
                "exactVersion": "1.1686127202.30113"
            },
            "osDisk": {
                "caching": "ReadWrite",
                "createOption": "FromImage",
                "diskSizeGB": "30",
                "diffDiskSettings": {
                    "option": "Local"
                },
                "encryptionSettings": {
                  "enabled": "false",
                  "diskEncryptionKey": {
                    "sourceVault": {
                      "id": "/subscriptions/test-source-
guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/test-kv"
                    },
                    "secretUrl": "https://test-
disk.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxx"
                  },
                  "keyEncryptionKey": {
                    "sourceVault": {
                      "id": "/subscriptions/test-key-
guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/test-kv"
                    },
                    "keyUrl": "https://test-key.vault.azure.net/secrets/xxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
                  }
                },
                "image": {
                    "uri": ""
```

```
                },
                "managedDisk": {
                    "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampleosdiskname",
                    "storageAccountType": "StandardSSD_LRS"
                },
                "name": "exampleosdiskname",
                "osType": "Windows",
                "vhd": {
                    "uri": ""
                },
                "writeAcceleratorEnabled": "false"
            },
            "resourceDisk": {
                "size": "4096"
            }
        },
        "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx",
        "tags": "baz:bash;foo:bar",
        "userData": "Zm9vYmFy",
        "version": "15.05.22",
        "virtualMachineScaleSet": {
            "id": "/subscriptions/xxxxxxxx-xxxxx-xxx-xxx-
xxxx/resourceGroups/resource-group-
name/providers/Microsoft.Compute/virtualMachineScaleSets/virtual-machine-scale-
set-name"
        },
        "vmId": "02aab8a4-74ef-476e-8182-f6d2ba4166a6",
        "vmScaleSetName": "crpteste9vflji9",
        "vmSize": "Standard_A3",
        "zone": ""
    },
    "network": {
        "interface": [{
            "ipv4": {
               "ipAddress": [{
                    "privateIpAddress": "10.144.133.132",
                    "publicIpAddress": ""
                }],
                "subnet": [{
                    "address": "10.144.133.128",
                    "prefix": "26"
                }]
            },
            "ipv6": {
                "ipAddress": [
                 ]
            },
            "macAddress": "0011AAFFBB22"
        }]
    }
}
```

Schema breakdown:

Compute

Expand table

Data	Description	Version introduced
`azEnvironment`	Azure Environment where the VM is running in	2018-10-01
`additionalCapabilities.hibernationEnabled`	Identifies if hibernation is enabled on the VM	2021-11-01
`customData`	This feature is deprecated and disabled in IMDS. It has been superseded by `userData`	2019-02-01
`evictionPolicy`	Sets how a Spot VM will be evicted.	2020-12-01
`extendedLocation.type`	Type of the extended location of the VM.	2021-03-01
`extendedLocation.name`	Name of the extended location of the VM	2021-03-01
`host.id`	Name of the host of the VM. Note that a VM will either have a host or a hostGroup but not both.	2021-11-15
`hostGroup.id`	Name of the hostGroup of the VM. Note that a VM will either have a host or a hostGroup but not both.	2021-11-15
`isHostCompatibilityLayerVm`	Identifies if the VM runs on the Host Compatibility Layer	2020-06-01
`licenseType`	Type of license for Azure Hybrid Benefit. This is only present for AHB-enabled VMs	2020-09-01
`location`	Azure Region the VM is running in	2017-04-02
`name`	Name of the VM	2017-04-02
`offer`	Offer information for the VM image and is only present for images deployed from Azure image gallery	2017-04-02
`osProfile.adminUsername`	Specifies the name of the admin account	2020-07-15
`osProfile.computerName`	Specifies the name of the computer	2020-07-15
`osProfile.disablePasswordAuthentication`	Specifies if password authentication is	2020-10-01

Data	Description	Version introduced
	disabled. This is only present for Linux VMs	
`osType`	Linux or Windows	2017-04-02
`physicalZone`	Physical zone of the VM	2023-11-15
`placementGroupId`	Placement Group of your scale set	2017-08-01
`plan`	Plan containing name, product, and publisher for a VM if it's an Azure Marketplace Image	2018-04-02
`platformUpdateDomain`	Update domain the VM is running in	2017-04-02
`platformFaultDomain`	Fault domain the VM is running in	2017-04-02
`platformSubFaultDomain`	Sub fault domain the VM is running in, if applicable.	2021-10-01
`priority`	Priority of the VM. Refer to Spot VMs for more information	2020-12-01
`provider`	Provider of the VM	2018-10-01
`publicKeys`	Collection of Public Keys assigned to the VM and paths	2018-04-02
`publisher`	Publisher of the VM image	2017-04-02
`resourceGroupName`	Resource group for your Virtual Machine	2017-08-01
`resourceId`	The fully qualified ID of the resource	2019-03-11
`sku`	Specific SKU for the VM image	2017-04-02
`securityProfile.secureBootEnabled`	Identifies if UEFI secure boot is enabled on the VM	2020-06-01
`securityProfile.virtualTpmEnabled`	Identifies if the virtual Trusted Platform Module (TPM) is enabled on the VM	2020-06-01
`securityProfile.encryptionAtHost`	Identifies if Encryption at Host is enabled on the VM	2021-11-01
`securityProfile.securityType`	Identifies if the VM is a Trusted VM or a Confidential VM	2021-12-13
`storageProfile`	See Storage Profile below	2019-06-01
`subscriptionId`	Azure subscription for the Virtual Machine	2017-08-01

Data	Description	Version introduced
`tags`	Tags for your Virtual Machine	2017-08-01
`tagsList`	Tags formatted as a JSON array for easier programmatic parsing	2019-06-04
`userData`	The set of data specified when the VM was created for use during or after provisioning (Base64 encoded)	2021-01-01
`version`	Version of the VM image	2017-04-02
`virtualMachineScaleSet.id`	ID of the Virtual Machine Scale Set created with flexible orchestration the Virtual Machine is part of. This field isn't available for Virtual Machine Scale Sets created with uniform orchestration.	2021-03-01
`vmId`	Unique identifier for the VM. The blog referenced only suits for VMs that have SMBIOS < 2.6. For VMs that have SMBIOS >= 2.6, the UUID from DMI is displayed in little-endian format, thus, there's no requirement to switch bytes.	2017-04-02
`vmScaleSetName`	Virtual Machine Scale Set Name of your scale set	2017-12-01
`vmSize`	VM size	2017-04-02
`zone`	Availability Zone of your virtual machine	2017-12-01

† This version isn't fully available yet and may not be supported in all regions.

Storage profile

The storage profile of a VM is divided into three categories: image reference, OS disk, and data disks, plus an additional object for the local temporary disk.

The image reference object contains the following information about the OS image, please note that an image could come either from the platform, marketplace, community gallery, or direct shared gallery but not both:

Expand table

Data	Description	Version introduced
`id`	Resource ID	2019-06-01
`offer`	Offer of the platform or marketplace image	2019-06-01
`publisher`	Publisher of the platform or marketplace image	2019-06-01
`sku`	Sku of the platform or marketplace image	2019-06-01
`version`	Version of the image	2019-06-01
`communityGalleryImageId`	Resource ID of the community image, empty otherwise	2023-07-01
`sharedGalleryImageId`	Resource ID o direct shared image, empty otherwise	2023-07-01
`exactVersion`	Version of the community or direct shared image	2023-07-01

The OS disk object contains the following information about the OS disk used by the VM:

Expand table

Data	Description
`caching`	Caching requirements
`createOption`	Information about how the VM was created
`diffDiskSettings`	Ephemeral disk settings
`diskSizeGB`	Size of the disk in GB
`image`	Source user image virtual hard disk
`managedDisk`	Managed disk parameters
`name`	Disk name
`vhd`	Virtual hard disk
`writeAcceleratorEnabled`	Whether or not writeAccelerator is enabled on the disk

The data disks array contains a list of data disks attached to the VM. Each data disk object contains the following information:

Expand table

Data	Description	Version introduced
`bytesPerSecondThrottle`*	Disk read/write quota in bytes	2021-05-01
`caching`	Caching requirements	2019-06-01
`createOption`	Information about how the VM was created	2019-06-01
`diffDiskSettings`	Ephemeral disk settings	2019-06-01
`diskCapacityBytes`*	Size of disk in bytes	2021-05-01
`diskSizeGB`	Size of the disk in GB	2019-06-01
`encryptionSettings`	Encryption settings for the disk	2019-06-01
`image`	Source user image virtual hard disk	2019-06-01
`isSharedDisk`*	Identifies if the disk is shared between resources	2021-05-01
`isUltraDisk`	Identifies if the data disk is an Ultra Disk	2021-05-01
`lun`	Logical unit number of the disk	2019-06-01
`managedDisk`	Managed disk parameters	2019-06-01
`name`	Disk name	2019-06-01
`opsPerSecondThrottle`*	Disk read/write quota in IOPS	2021-05-01
`osType`	Type of OS included in the disk	2019-06-01
`vhd`	Virtual hard disk	2019-06-01
`writeAcceleratorEnabled`	Whether or not writeAccelerator is enabled on the disk	2019-06-01

*These fields are only populated for Ultra Disks; they are empty strings from non-Ultra Disks.

The encryption settings blob contains data about how the disk is encrypted (if it's encrypted):

Expand table

Data	Description	Version introduced
`diskEncryptionKey.sourceVault.id`	The location of the disk encryption key	2021-11-01
`diskEncryptionKey.secretUrl`	The location of the secret	2021-11-01
`keyEncryptionKey.sourceVault.id`	The location of the key encryption key	2021-11-01
`keyEncryptionKey.keyUrl`	The location of the key	2021-11-01

The resource disk object contains the size of the Local Temp Disk attached to the VM, if it has one, in kilobytes. If there's no local temp disk for the VM, this value is 0.

Expand table

Data	Description	Version introduced
`resourceDisk.size`	Size of the local temp disk for the VM (in kB)	2021-02-01

Network

Expand table

Data	Description	Version introduced
`ipv4.ipAddress.privateIpAddress`	Local IPv4 address of the VM	2017-04-02
`ipv4.ipAddress.publicIpAddress`	Public IPv4 address of the VM	2017-04-02
`ipv4.subnet.address`	Subnet address of the VM	2017-04-02
`ipv4.subnet.prefix`	Subnet prefix, example 24	2017-04-02
`ipv6.ipAddress.privateIpAddress`	Local IPv6 address of the VM	2017-04-02
`macAddress`	VM mac address	2017-04-02

> **Note**
>
> The nics returned by the network call are not guaranteed to be in order.

Get user data

When creating a new VM, you can specify a set of data to be used during or after the VM provision, and retrieve it through IMDS. Check the end to end user data experience here.

To set up user data, utilize the quickstart template here. The sample below shows how to retrieve this data through IMDS. This feature is released with version `2021-01-01` and above.

> **Note**
>
> Security notice: IMDS is open to all applications on the VM, sensitive data should not be placed in the user data.

Windows

PowerShell

```
$userData = Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy
-Uri "http://169.254.169.254/metadata/instance/compute/userData?api-
version=2025-04-07&format=text"
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($userData))
```

Sample 1: Tracking VM running on Azure

As a service provider, you may require to track the number of VMs running your software or have agents that need to track uniqueness of the VM. To be able to get a unique ID for a VM, use the `vmId` field from Instance Metadata Service.

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance/compute/vmId?api-version=2025-04-
07&format=text"
```

Response

Output

```
5c08b38e-4d57-4c23-ac45-aca61037f084
```

Sample 2: Placement of different data replicas

For certain scenarios, placement of different data replicas is of prime importance. For example, HDFS replica placement or container placement via an orchestrator might require you to know the `platformFaultDomain` and `platformUpdateDomain` the VM is running on. You can also use Availability Zones for the instances to make these decisions. You can query this data directly via IMDS.

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance/compute/platformFaultDomain?api-
version=2025-04-07&format=text"
```

Response

Output

```
0
```

Sample 3: Get VM tags

VM tags are included the instance API under instance/compute/tags endpoint. Tags may have been applied to your Azure VM to logically organize them into a taxonomy. The tags assigned to a VM can be retrieved by using the request below.

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance/compute/tags?api-version=2025-04-
07&format=text"
```

Response

Output

```
Department:IT;ReferenceNumber:123456;TestStatus:Pending
```

The `tags` field is a string with the tags delimited by semicolons. This output can be a problem if semicolons are used in the tags themselves. If a parser is written to programmatically extract the tags, you should rely on the `tagsList` field. The `tagsList` field is a JSON array with no delimiters, and consequently, easier to parse. The tagsList assigned to a VM can be retrieved by using the request below.

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance/compute/tagsList?api-version=2025-04-
07" | ConvertTo-Json -Depth 64
```

Response

Windows

JSON


```
{
    "value":  [
                  {
                      "name":  "Department",
                      "value":  "IT"
                  },
                  {
                      "name":  "ReferenceNumber",
                      "value":  "123456"
                  },
                  {
                      "name":  "TestStatus",
                      "value":  "Pending"
                  }
              ],
    "Count":  3
}
```


Sample 4: Get more information about the VM during support case

As a service provider, you may get a support call where you would like to know more information about the VM. Asking the customer to share the compute metadata can provide basic information for the support professional to know about the kind of VM on Azure.

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance/compute?api-version=2025-04-07" |
ConvertTo-Json -Depth 64
```

Response

> **Note**
>
> The response is a JSON string. The following example response is pretty-printed for readability.

Windows

JSON


```
{
    "azEnvironment": "AZUREPUBLICCLOUD",
    "extendedLocation": {
      "type": "edgeZone",
      "name": "microsoftlosangeles"
    },
    "evictionPolicy": "",
    "additionalCapabilities": {
        "hibernationEnabled": "false"
    },
    "hostGroup": {
      "id": "testHostGroupId"
    },
    "isHostCompatibilityLayerVm": "true",
    "licenseType":  "Windows_Client",
    "location": "westus",
    "name": "examplevmname",
    "offer": "WindowsServer",
    "osProfile": {
        "adminUsername": "admin",
        "computerName": "examplevmname",
        "disablePasswordAuthentication": "true"
    },
    "osType": "Windows",
    "placementGroupId": "f67c14ab-e92c-408c-ae2d-da15866ec79a",
    "plan": {
        "name": "planName",
        "product": "planProduct",
        "publisher": "planPublisher"
```

```
    },
    "platformFaultDomain": "36",
    "platformUpdateDomain": "42",
    "priority": "Regular",
    "publicKeys": [{
            "keyData": "ssh-rsa 0",
            "path": "/home/user/.ssh/authorized_keys0"
        },
        {
            "keyData": "ssh-rsa 1",
            "path": "/home/user/.ssh/authorized_keys1"
        }
    ],
    "publisher": "RDFE-Test-Microsoft-Windows-Server-Group",
    "physicalZone": "useast-AZ01",
    "resourceGroupName": "macikgo-test-may-23",
    "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/virtualMachines/examplevmname",
    "securityProfile": {
        "secureBootEnabled": "true",
        "virtualTpmEnabled": "false",
        "encryptionAtHost": "true",
        "securityType": "TrustedLaunch"
    },
    "sku": "2019-Datacenter",
    "storageProfile": {
        "dataDisks": [{
            "bytesPerSecondThrottle": "979202048",
            "caching": "None",
            "createOption": "Empty",
            "diskCapacityBytes": "274877906944",
            "diskSizeGB": "1024",
            "image": {
              "uri": ""
            },
            "isSharedDisk": "false",
            "isUltraDisk": "true",
            "lun": "0",
            "managedDisk": {
              "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/MicrosoftCompute/disks/exampledatadiskname",
              "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampledatadiskname",
            "opsPerSecondThrottle": "65280",
            "vhd": {
              "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        }],
        "imageReference": {
            "id": "",
            "offer": "WindowsServer",
```

```
            "publisher": "MicrosoftWindowsServer",
            "sku": "2019-Datacenter",
            "version": "latest",
            "communityGalleryImageId":
"/CommunityGalleries/testgallery/Images/1804Gen2/Versions/latest",
            "sharedGalleryImageId":
"/SharedGalleries/1P/Images/gen2/Versions/latest",
            "exactVersion": "1.1686127202.30113"
        },
        "osDisk": {
            "caching": "ReadWrite",
            "createOption": "FromImage",
            "diskSizeGB": "30",
            "diffDiskSettings": {
                "option": "Local"
            },
            "encryptionSettings": {
              "enabled": "false",
              "diskEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-source-
guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/test-kv"
                },
                "secretUrl": "https://test-disk.vault.azure.net/secrets/xxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
              },
              "keyEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-key-
guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/test-kv"
                },
                "keyUrl": "https://test-key.vault.azure.net/secrets/xxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx"
              }
            },
            "image": {
                "uri": ""
            },
            "managedDisk": {
                "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx/resourceGroups/macikgo-test-may-
23/providers/Microsoft.Compute/disks/exampleosdiskname",
                "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampleosdiskname",
            "osType": "Windows",
            "vhd": {
                "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        },
        "resourceDisk": {
            "size": "4096"
        }
    },
```

```
    "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
    "tags": "baz:bash;foo:bar",
    "version": "15.05.22",
    "virtualMachineScaleSet": {
      "id": "/subscriptions/xxxxxxxx-xxxxx-xxx-xxx-xxxx/resourceGroups/resource-group-name/providers/Microsoft.Compute/virtualMachineScaleSets/virtual-machine-scale-set-name"
    },
    "vmId": "02aab8a4-74ef-476e-8182-f6d2ba4166a6",
    "vmScaleSetName": "crpteste9vflji9",
    "vmSize": "Standard_A3",
    "zone": "3"
}
```


Sample 5: Get the Azure Environment where the VM is running

Azure has various sovereign clouds like Azure Government. Sometimes you need the Azure Environment to make some runtime decisions. The following sample shows you how you can achieve this behavior.

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/instance/compute/azEnvironment?api-version=2025-04-07&format=text"
```

Response

Output

```
AzurePublicCloud
```

The cloud and the values of the Azure environment are listed here.

Expand table

Cloud	Azure environment
All generally available global Azure regions	AzurePublicCloud
Azure Government	AzureUSGovernmentCloud

Cloud	Azure environment
Microsoft Azure operated by 21Vianet	AzureChinaCloud
Azure Germany	AzureGermanCloud

Sample 6: Retrieve network information

Request

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri
"http://169.254.169.254/metadata/instance/network?api-version=2025-04-07" |
ConvertTo-Json  -Depth 64
```

Response

JSON


```
{
  "interface": [
    {
      "ipv4": {
        "ipAddress": [
          {
            "privateIpAddress": "10.1.0.4",
            "publicIpAddress": "X.X.X.X"
          }
        ],
        "subnet": [
          {
            "address": "10.1.0.0",
            "prefix": "24"
          }
        ]
      },
      "ipv6": {
        "ipAddress": [{
          "privateIpAddress": "b4bc:8fce:f33b:4990:cced:d94e:ab4f:6ea0"
        }]
      },
      "macAddress": "000D3AF806EC"
    }
```

```
  ]
}
```

Sample 7: Retrieve public IP address

Windows

PowerShell

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/instance/network/interface/0/ipv4/ipAddress/0/publicIpAddress?api-version=2025-04-07&format=text"
```

> **Note**
>
> - If you're looking to retrieve IMDS information for **Standard** SKU Public IP address, review Load Balancer Metadata API for more infomration.

Attested data

Get Attested data

IMDS helps to provide guarantees that the data provided is coming from Azure. Microsoft signs part of this information, so you can confirm that an image in Azure Marketplace is the one you're running on Azure.

```
GET /metadata/attested/document
```

Parameters

Expand table

Name	Required/Optional	Description
`api-version`	Required	The version used to service the request.

Name	Required/Optional	Description
`nonce`	Optional	A 10-digit string that serves as a cryptographic nonce. If no value is provided, IMDS uses the current UTC timestamp.

Response

JSON


```
{
    "encoding":"pkcs7",

"signature":"MIIEEgYJKoZIhvcNAQcCoIIEAzCCA/8CAQExDzANBgkqhkiG9w0BAQsFADCBugYJKoZIhvc
NAQcBoIGsBIGpeyJub25jZSI6IjEyMzQ1NjY3NjYiLCJwbGFuIjp7Im5hbWUiOiIiLCJwcm9kdWN0IjoiIiw
icHVibGlzaGVyIjoiIn0sInRpbWVTdGFtcCI6eyJjcmVhdGVkT24iOiIxMS8yMC8xOCAyMjowNzozOSAtMDA
wMCIsImV4cGlyZXNPbiI6IjExLzIwLzE4IDIyOjA4OjI0IC0wMDAwIn0sInZtSWQiOiIifaCCAj8wggI7MII
BpKADAgECAhBnxW5Kh8dslEBA0E2mIBJ0MA0GCSqGSIb3DQEBBAUAMCsxKTAnBgNVBAMTIHRlc3RzdWJkb21
haW4ubWV0YWRhdGEuYXp1cmUuY29tMB4XDTE4MTEyMDIxNTc1N1oXDTE4MTIyMDIxNTc1NlowKzEpMCcGA1U
EAxMgdGVzdHN1YmRvbWFpbi5tZXRhZGF0YS5henVyZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoG
BAML/tBo86ENWPzmXZ0kPkX5dY5QZ150mA8lommszE71x2sCLonzv4/UWk4H+jMMWRRwIea2CuQ5RhdWAHvK
q6if4okKNt66fxm+YTVz9z0CTfCLmLT+nsdfOAsG1xZppEapC0Cd9vD6NCKyE8aYI1pliaeOnFjG0WvMY04u
Wz2MdAgMBAAGjYDBeMFwGA1UdAQRVMFOAENnYkHLa04Ut4Mpt7TkJFfyhLTArMSkwJwYDVQQDEyB0ZXN0c3V
iZG9tYWluLm1ldGFkYXRhLmF6dXJlLmNvbYIQZ8VuSofHbJRAQNBNpiASdDANBgkqhkiG9w0BAQQFAAOBgQC
LSM6aX5Bs1KHCJp4VQtxZPzXF71rVKCocHy3N9PTJQ9Fpnd+bYw2vSpQHg/AiG82WuDFpPReJvr7Pa938mZq
W9HUOGjQKK2FYDTg6fXD8pkPdyghlX5boGWAMMrf7bFkup+lsT+n2tRw2wbNknO1tQ0wICtqy2VqzWwLi45R
BwTGB6DCB5QIBATA/MCsxKTAnBgNVBAMTIHRlc3RzdWJkb21haW4ubWV0YWRhdGEuYXp1cmUuY29tAhBnxW5
Kh8dslEBA0E2mIBJ0MA0GCSqGSIb3DQEBCwUAMA0GCSqGSIb3DQEBAQUABIGAld1BM/yYIqqv8SDE4kjQo3U
l/IKAVR8ETKcve5BAdGSNkTUooUGVniTXeuvDj5NkmazOaKZp9fEtByqqPOyw/nlXaZgOO44HDGiPUJ90xVY
mfeK6p9RpJBu6kiKhnnYTelUk5u75phe5ZbMZfBhuPhXmYAdjc7Nmw97nx8NnprQ="
}
```


The signature blob is a pkcs7 -signed version of document. It contains the certificate used for signing along with certain VM-specific details.

For VMs created by using Azure Resource Manager, the document includes `vmId`, `sku`, `nonce`, `subscriptionId`, `timeStamp` for creation and expiry of the document, and the plan information about the image. The plan information is only populated for Azure Marketplace images.

For VMs created by using the classic deployment model, only the `vmId` and `subscriptionId` are guaranteed to be populated. You can extract the certificate from the response, and use it to confirm that the response is valid and is coming from Azure.

The decoded document contains the following fields:

Expand table

Data	Description	Version introduced
`licenseType`	Type of license for Azure Hybrid Benefit. This is only present for AHB-enabled VMs.	2020-09-01
`nonce`	A string that can be optionally provided with the request. If no `nonce` was supplied, the current Coordinated Universal Time timestamp is used.	2018-10-01
`plan`	The Azure Marketplace Image plan. Contains the plan ID (name), product image or offer (product), and publisher ID (publisher).	2018-10-01
`timestamp.createdOn`	The UTC timestamp for when the signed document was created	2018-20-01
`timestamp.expiresOn`	The UTC timestamp for when the signed document expires	2018-10-01
`vmId`	Unique identifier for the VM	2018-10-01
`subscriptionId`	Azure subscription for the Virtual Machine	2019-04-30
`sku`	Specific SKU for the VM image (correlates to `compute/sku` property from the Instance Metadata endpoint [`/metadata/instance`])	2019-11-01

> **Note**
>
> For Classic (non-Azure Resource Manager) VMs, only the vmId is guaranteed to be populated.

Example document:

JSON


```
{
   "nonce":"20201130-211924",
   "plan":{
      "name":"planName",
      "product":"planProduct",
      "publisher":"planPublisher"
   },
   "sku":"Windows-Server-2012-R2-Datacenter",
   "subscriptionId":"aaaa0a0a-bb1b-cc2c-dd3d-eeeeee4e4e4e",
   "timeStamp":{
      "createdOn":"11/30/20 21:19:19 -0000",
      "expiresOn":"11/30/20 21:19:24 -0000"
   },
   "vmId":"02aab8a4-74ef-476e-8182-f6d2ba4166a6"
}
```

Signature Validation Guidance

When validating the signature, you should confirm that the signature was created with a certificate from Azure. This is done by validating the certificate Subject Alternative Name (SAN).

Example SAN `DNS Name=eastus.metadata.azure.com, DNS Name=metadata.azure.com`

> **Note**
>
> The domain for the public cloud and each sovereign cloud will be different.

Expand table

Cloud	Domain in SAN
All generally available global Azure regions	*.metadata.azure.com
Azure Government	*.metadata.azure.us
Azure operated by 21Vianet	*.metadata.azure.cn
Azure Germany	*.metadata.microsoftazure.de

> **Note**
>
> The certificates might not have an exact match for the domain. For this reason, the certification validation should accept any subdomain (for example, in public cloud general availability regions accept `*.metadata.azure.com`).

We don't recommend certificate pinning for intermediate certs. For further guidance, see Certificate pinning - Certificate pinning and Azure services. Please note that the Azure Instance Metadata Service will NOT offer notifications for future Certificate Authority changes. Instead, you must follow the centralized Azure Certificate Authority details article for all future updates.

Sample 1: Validate that the VM is running in Azure

Vendors in Azure Marketplace want to ensure that their software is licensed to run only in Azure. If someone copies the VHD to an on-premises environment, the vendor needs to be able to detect that. Through IMDS, these vendors can get signed data that guarantees response only from Azure.

> **Note**

This sample requires the jq utility to be installed.

Validation

Windows

PowerShell

```
# Get the signature
$attestedDoc = Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -
NoProxy -Uri http://169.254.169.254/metadata/attested/document?api-version=2025-
04-07
# Decode the signature
$signature = [System.Convert]::FromBase64String($attestedDoc.signature)
```

Verify that the signature is from Microsoft Azure and checks the certificate chain for errors.

PowerShell

```
# Get certificate chain
$cert = [System.Security.Cryptography.X509Certificates.X509Certificate2]
($signature)
$chain = New-Object -TypeName
System.Security.Cryptography.X509Certificates.X509Chain
$chain.Build($cert)
# Print the Subject of each certificate in the chain
foreach($element in $chain.ChainElements)
{
    Write-Host $element.Certificate.Subject
}

# Get the content of the signed document
Add-Type -AssemblyName System.Security
$signedCms = New-Object -TypeName System.Security.Cryptography.Pkcs.SignedCms
$signedCms.Decode($signature);
$content =
[System.Text.Encoding]::UTF8.GetString($signedCms.ContentInfo.Content)
Write-Host "Attested data: " $content
$json = $content | ConvertFrom-Json
# Do additional validation here
```

The `nonce` in the signed document can be compared if you provided a `nonce` parameter in the initial request.

Managed identity

A managed identity, assigned by the system, can be enabled on the VM. You can also assign one or more user-assigned managed identities to the VM. You can then request tokens for managed identities from IMDS. Use these tokens to authenticate with other Azure services, such as Azure Key Vault.

For detailed steps to enable this feature, see Acquire an access token.

Load Balancer Metadata

When you place virtual machine or virtual machine set instances behind an Azure Standard Load Balancer, you can use IMDS to retrieve metadata related to the load balancer and the instances. For more information, see Retrieve load balancer information.

Scheduled events

You can obtain the status of the scheduled events by using IMDS. Then the user can specify a set of actions to run upon these events. For more information, see Scheduled events for Linux or Scheduled events for Windows.

Sample code in different languages

The following table lists samples of calling IMDS by using different languages inside the VM:

Expand table

Language	Example
Bash	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.sh
C#	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.cs
Go	https://github.com/Microsoft/azureimds/blob/master/imdssample.go
Java	https://github.com/Microsoft/azureimds/blob/master/imdssample.java
NodeJS	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.js
Perl	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.pl
PowerShell	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.ps1
Puppet	https://github.com/keirans/azuremetadata
Python	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.py

Language	Example
Ruby	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.rb

Errors and debugging

If there's a data element not found or a malformed request, the Instance Metadata Service returns standard HTTP errors. For example:

Expand table

HTTP status code	Reason
`200 OK`	The request was successful.
`400 Bad Request`	Missing `Metadata: true` header or missing parameter `format=json` when querying a leaf node
`404 Not Found`	The requested element doesn't exist
`405 Method Not Allowed`	The HTTP method (verb) isn't supported on the endpoint.
`410 Gone`	Retry after some time for a max of 70 seconds
`429 Too Many Requests`	API Rate Limits have been exceeded
`500 Service Error`	Retry after some time

Frequently asked questions

- **I'm getting the error `400 Bad Request, Required metadata header not specified`. What does this mean?**
  - IMDS requires the header `Metadata: true` to be passed in the request. Passing this header in the REST call allows access to IMDS.
- **Why am I not getting compute information for my VM?**
  - Currently, IMDS only supports instances created with Azure Resource Manager.
- **I created my VM through Azure Resource Manager some time ago. Why am I not seeing compute metadata information?**
  - If you created your VM after September 2016, add a tag to start seeing compute metadata. If you created your VM before September 2016, add or remove extensions

or data disks to the VM instance to refresh metadata.

- **Is user data the same as custom data?**
  - User data offers the similar functionality to custom data, allowing you to pass your own metadata to the VM instance. The difference is, user data is retrieved through IMDS, and is persistent throughout the lifetime of the VM instance. Existing custom data feature will continue to work as described in this article. However you can only get custom data through local system folder, not through IMDS.

- **Why am I not seeing all data populated for a new version?**
  - If you created your VM after September 2016, add a tag to start seeing compute metadata. If you created your VM before September 2016, add or remove extensions or data disks to the VM instance to refresh metadata.

- **Why am I getting the error `500 Internal Server Error` or `410 Resource Gone`?**
  - Retry your request. For more information, see Transient fault handling. If the problem persists, create a support issue in the Azure portal for the VM.

- **Would this work for scale set instances?**
  - Yes, IMDS is available for scale set instances.

- **I updated my tags in my scale sets, but they don't appear in the instances (unlike single instance VMs). Am I doing something wrong?**
  - Currently tags for scale sets only show to the VM on a reboot, reimage, or disk change to the instance.

- **Why am I'm not seeing the SKU information for my VM in `instance/compute` details?**
  - For custom images created from Azure Marketplace, Azure platform doesn't retain the SKU information for the custom image and the details for any VMs created from the custom image. This is by design and hence not surfaced in the VM `instance/compute` details.

- **Why is my request timed out (or failed to connect) for my call to the service?**

  - Metadata calls must be made from the primary IP address assigned to the primary network card of the VM. Additionally, if you've changed your routes, there must be a route for the 169.254.169.254/32 address in your VM's local routing table.

    Windows

    1. Dump your local routing table and look for the IMDS entry. For example:

       Console

```
route print
```

Output

```
IPv4 Route Table
===========================================================================
=======
Active Routes:
Network Destination        Netmask          Gateway       Interface
Metric
        0.0.0.0          0.0.0.0      172.16.69.1      172.16.69.7
10
        127.0.0.0        255.0.0.0         On-link         127.0.0.1
331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1
331
127.255.255.255  255.255.255.255         On-link         127.0.0.1
331
    168.63.129.16  255.255.255.255      172.16.69.1      172.16.69.7
11
169.254.169.254  255.255.255.255      172.16.69.1      172.16.69.7
11
... (continues) ...
```

2. Verify that a route exists for `169.254.169.254`, and note the corresponding network interface (for example, `172.16.69.7`).

3. Dump the interface configuration and find the interface that corresponds to the one referenced in the routing table, noting the MAC (physical) address.

Console

```
ipconfig /all
```

Output

```
... (continues) ...
Ethernet adapter Ethernet:

Connection-specific DNS Suffix  . :
xic3mnxjiefupcwr1mcs1rjiqa.cx.internal.cloudapp.net
Description . . . . . . . . . . . : Microsoft Hyper-V Network
Adapter
Physical Address. . . . . . . . . : 00-0D-3A-E5-1C-C0
DHCP Enabled. . . . . . . . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . :
fe80::3166:ce5a:2bd5:a6d1%3(Preferred)
IPv4 Address. . . . . . . . . . . : 172.16.69.7(Preferred)
```

```
Subnet Mask . . . . . . . . . . . : 255.255.255.0
... (continues) ...
```

4. Confirm that the interface corresponds to the VM's primary NIC and primary IP. You can find the primary NIC and IP by looking at the network configuration in the Azure portal, or by looking it up with the Azure CLI. Note the private IPs (and the MAC address if you're using the CLI). Here's a PowerShell CLI example:

   Azure PowerShell

   ```
   $ResourceGroup = '<Resource_Group>'
   $VmName = '<VM_Name>'
   $NicNames = az vm nic list --resource-group $ResourceGroup --vm-name
   $VmName | ConvertFrom-Json | Foreach-Object { $_.id.Split('/')[-1] }
   foreach($NicName in $NicNames)
   {
       $Nic = az vm nic show --resource-group $ResourceGroup --vm-name
   $VmName --nic $NicName | ConvertFrom-Json
       Write-Host $NicName, $Nic.primary, $Nic.macAddress
   }
   ```

   Output

   ```
   wintest767 True 00-0D-3A-E5-1C-C0
   ```

5. If they don't match, update the routing table so that the primary NIC and IP are targeted.

- **Fail over clustering in Windows Server**
  - When you're querying IMDS with failover clustering, it's sometimes necessary to add a route to the routing table. Here's how:

    1. Open a command prompt with administrator privileges.
    2. Run the following command, and note the address of the Interface for Network Destination (`0.0.0.0`) in the IPv4 Route Table.

    bat

    ```
    route print
    ```

> **Note**
>
> The following example output is from a Windows Server VM with failover cluster enabled. For simplicity, the output contains only the IPv4 Route Table.

Output

```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
        0.0.0.0          0.0.0.0         10.0.1.1        10.0.1.10    266
        10.0.1.0  255.255.255.192         On-link         10.0.1.10    266
        10.0.1.10  255.255.255.255         On-link         10.0.1.10    266
        10.0.1.15  255.255.255.255         On-link         10.0.1.10    266
        10.0.1.63  255.255.255.255         On-link         10.0.1.10    266
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
    169.254.0.0      255.255.0.0         On-link     169.254.1.156    271
    169.254.1.156  255.255.255.255         On-link     169.254.1.156    271
169.254.255.255  255.255.255.255         On-link     169.254.1.156    271
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link     169.254.1.156    271
255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
255.255.255.255  255.255.255.255         On-link     169.254.1.156    271
255.255.255.255  255.255.255.255         On-link         10.0.1.10    266
```

Run the following command and use the address of the Interface for Network Destination (`0.0.0.0`), which is (`10.0.1.10`) in this example.

bat

```
route add 169.254.169.254/32 10.0.1.10 metric 1 -p
```

Support

If you aren't able to get a metadata response after multiple attempts, you can create a support issue in the Azure portal.

Product feedback

You can provide product feedback and ideas to our user feedback channel under Virtual Machines > Instance Metadata Service here

Next steps

- Acquire an access token for the VM
- Scheduled events for Linux
- Scheduled events for Windows

Last updated on 07/29/2025

Metadata Security Protocol (MSP)

10/08/2025

Metadata Security Protocol (MSP) is a feature in the Azure Virtual Machines and Azure Virtual Machine Scale Sets services. It enhances the security of the Azure Instance Metadata Service and WireServer services. These services are available in Azure infrastructure as a service (IaaS) virtual machines (VMs) or virtual machine scale sets at 169.254.169.254 and 168.63.129.16, respectively.

Organizations use Instance Metadata Service and WireServer for providing metadata and bootstrapping VM credentials. As a result, threat actors frequently target these services. Common vectors include confused deputy attacks against in-guest workloads and sandbox escapes. These vectors are of particular concern for hosted-on-behalf-of workloads where untrusted code loads into the VM.

With metadata services, the trust boundary is the VM itself. Any software within the guest is authorized to request secrets from Instance Metadata Service and WireServer. VM owners are responsible for carefully sandboxing any software that they run inside the VM and ensuring that external actors can't exfiltrate data. In practice, the complexity of the problem leads to mistakes at scale, which in turn lead to exploits.

Although numerous defense-in-depth strategies exist, providing secrets over an unauthenticated HTTP API carries inherent risk. Across the industry, this class of vulnerabilities has affected hundreds of companies, affected millions of individuals, and caused financial losses in the hundreds of millions of dollars.

MSP closes the most common vulnerabilities by:

- Addressing the root cause of these attacks.
- Introducing strong authentication and authorization concepts to cloud metadata services.

Compatibility

MSP is supported on Azure IaaS VMs and virtual machine scale sets that run these operating systems:

- Windows 10 or later (x64)
- Windows Server 2019 (x64)
- Windows Server 2022 (x64)
- Windows Server 2025 (x64)
- Mariner 2.0 (x64, ARM64)
- Azure Linux (Mariner 3.0) (x64, ARM64)

- Ubuntu 20.04+ (x64, ARM64)

The following items are currently not supported:

- Red Hat Enterprise Linux 9+
- Rocky Linux 9+
- SUSE Linux Enterprise Server 15 SP4+
- Ephemeral disks
- Compatibility with Azure Backup
- ARM64

Enhanced security

The Guest Proxy Agent (GPA) hardens against these types of attacks by:

- Limiting metadata access to a subset of the VM (applying the principle of least-privileged access).

- Switching from a *default-open* model to a *default-closed* model.

  For instance, with nested virtualization, a misconfigured L2 VM that has access to the L1 VM's vNIC can communicate with a metadata service as the L1. With the GPA, a misconfigured L2 could no longer gain access, because it would be unable to authenticate with the service.

At provisioning time, the metadata service establishes a trusted delegate within the guest (the GPA). A long-lived secret is negotiated to authenticate with the trusted delegate. The delegate must endorse all requests to the metadata service by using a hash-based message authentication code (HMAC). The HMAC establishes a point-to-point trust relationship with strong authorization.

The GPA uses eBPF to intercept HTTP requests to the metadata service. eBPF enables the GPA to verify the identity of the in-guest software that made the request. The GPA uses eBPF to intercept requests without requiring an extra kernel module. GPA uses this information to compare the identity of the client against an allowlist defined as a part of the VM model in Azure Resource Manager (ARM). The GPA then endorses requests by adding a signature header. So, you can enable the MSP feature on existing workloads without breaking changes.

By default, the existing authorization levels are enforced:

- Instance Metadata Service is open to all users.
- WireServer is root/admin only.

This restriction is currently accomplished with firewall rules in the guest. This is still a default-open mechanism. If that rule can be disabled or bypassed for any reason, the metadata service still accepts the request. The authorization mechanism enabled here is default-closed. Bypassing interception maliciously or by error doesn't grant access to the metadata service.

You can set up an advanced authorization configuration (that is, authorize specific in-guest processes and users to access only specific endpoints) by defining a custom allowlist with role-based access control (RBAC) semantics.

> ⚠ **Warning**
>
> Please note that on Windows, the users can enable GuestProxyAgent (GPA) from an ARM template. However, in Linux it's a two-step process. The VM/VMSS is provisioned first and only then can the GPA be configured.

Related content

- MSP feature configuration
- Deploy a VM or virtual machine scale set with MSP
- Enable MSP on an existing VM or virtual machine scale set

MSP feature configuration

10/08/2025

Metadata Security Protocol (MSP) offers customization to maximally restrict metadata server access in your workload. This article introduces fundamental concepts that you can quickly enable on any supported virtual machine (VM) or virtual machine scale set.

If you're familiar with how your workload uses metadata services, you can harden access further by following the guide for advanced configuration.

You can configure MSP via:

- Azure Resource Manager templates
- REST API
- PowerShell
- Azure portal

Concepts

Expand table

Term	Definition
Metadata services	A general term for the industry practice of offering private, non-internet, anonymous HTTP APIs available at a fixed location. The location is typically `169.254.169.254`. Although the exact features, details, and service count vary by cloud provider, they all share a similar architecture and purpose.
Azure Instance Metadata Service	A user-facing metadata service that comes with support and full API documentation. It's intended for both Azure and non-Microsoft integrations. It offers a mix of metadata and credentials.
WireServer	A platform-facing metadata service that organizations use to implement core infrastructure as a service (IaaS) functionality. It isn't intended for general consumption, and Azure offers no support for doing so. Although the consumers of the service are related to an Azure implementation, the service must still be considered in security assessments. WireServer resources are treated as privileged by default. However, you shouldn't view Instance Metadata Service resources as nonprivileged. Depending on your workload, Instance Metadata Service can also provide sensitive information.
Guest Proxy Agent (GPA)	An in-guest agent that enables MSP protections.

Azure Resource Manager fields

You manage all configuration via new properties in the `securityProfile` section of your resource's configuration.

The minimum API version to configure MSP is `2024-03-01`.

General configuration

`ProxyAgentSettings` is the new property in the VM model for Azure VMs and virtual machine scale sets that you use to configure the MSP feature.

Expand table

Parameter name	Type	Default	Details
`enabled`	`Bool`	`false`	Specifies whether the MSP feature should be enabled on the VM or virtual machine scale set. Default is `false`. This property controls: • Applying other settings. • Platform latched-key generation. • Automatic GPA installation (when `true`) or uninstallation (when `false`) on a Windows VM or virtual machine scale set.
`imds`	`HostEndpointSettings`	Not applicable	Instance Metadata Service-specific configuration. See Configuration for each metadata service later in this article.
`wireServer`	`HostEndpointSettings`	Not applicable	Settings for the WireServer endpoint. See Configuration for each metadata service later in this article.
`keyIncarnationId`	`Integer`	`0`	A counter for key generation. Increasing this value instructs MSP to reset the key used for securing a communication channel between guest and host. The GPA notices the key reset and automatically acquires a new one. This property is only for recovery and troubleshooting purposes.

Configuration for each metadata service

You can configure individual protections for each metadata service. The schema is the same for each service.

Inline and linked configurations are mutually exclusive on a per-service basis.

Inline configuration

You can define an inline configuration by using the `mode` property of `HostEndpointSettings`. Inline configurations don't support customization.

In the following table, `mode` is an enumeration that's expressed as `String`.

Expand table

`mode` value	GPA behavior	Service behavior
`Disabled`	The GPA doesn't set up eBPF interception of requests to this service. Requests go directly to the service.	Unchanged. The service does *not* require the GPA to endorse (sign) requests.
`Audit`	The GPA intercepts requests to this service and determines if the current configuration authorizes them. The request is always forwarded to the service, but the result and caller information are logged.	Unchanged. The service does *not* require the GPA to endorse (sign) requests.
`Enforce`	The GPA intercepts requests to this service and determines if the current configuration authorizes them. If the caller is authorized, the GPA signs the request to endorse it. If the caller isn't authorized, the GPA responds with status code `401` or `403`. All outcomes are recorded in the audit log.	The GPA *must* endorse (sign) requests to the service. Unsigned requests are rejected with a `401` status code.

> **Important**
>
> You can't use this property at the same time as `inVMAccessControlProfileReferenceId`.

In WireServer `Enforce` mode, the GPA implicitly requires that a process runs as `Administrator` or `root` to access WireServer. This requirement is equivalent to the in-guest firewall rule that exists even when MSP is disabled, but with a more secure implementation.

Linked configuration

The `inVMAccessControlProfiles` resource type defines a per-service configuration that can be linked against a VM or virtual machine scale set. These configurations support full

customizations. For full details, see Advanced configuration for MSP.

Expand table

`HostEndpointSettings` property	Type	Details
`inVMAccessControlProfileReferenceId`	`String`	The resource ID of the configuration to apply.

This property must be a full Resource Manager ID, which takes this form: `/subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Microsoft.Compute/galleries/{galleryName}/inVMAccessControlProfiles/{profile}/versions/{version}`.

> **Important**
>
> You can't use this property at the same time as `mode`.

Full example

- VM: `https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/{resource-group-name}/providers/Microsoft.Compute/virtualMachines/{virtualMachine-Name}?api-version=2024-03-01`
- Virtual machine scale set: `https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/{resource-group-name}/providers/Microsoft.Compute/virtualMachineScaleSets/{vmScaleSet-name}/virtualMachines/{instance-id}?api-version=2024-03-01`

The following example uses `mode`:

JSON

```
{
  "name": "GPAWinVM",
  "location": "centraluseuap",
  "properties": {
    ...
    ...
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": true,
        "wireServer": {
          "mode": "Enforce"
        },
        "imds": {
```

```
            "mode": "Audit"
          }
        }
      },
    }
}
```

The following example uses `inVMAccessControlProfileReferenceId:`

JSON

```
{
  "name": "GPAWinVM",
  "location": "centraluseuap",
  "properties": {
    ...
    ...
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": true,
        "wireServer": {
          "inVMAccessControlProfileReferenceId":
"/subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Micr
osoft.Compute/galleries/{galleryName}/inVMAccessControlProfiles/{profile}/versions
/{version}"
        },
        "imds": {
          "inVMAccessControlProfileReferenceId":
"/subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Micr
osoft.Compute/galleries/{galleryName}/inVMAccessControlProfiles/{profile}/versions
/{version}"
        }
      },
    }
}
```

Audit logs

In `Audit` and `Enforce` modes, audit logs are generated on the local disk.

Expand table

OS family	Audit log location
Linux	`/var/lib/azure-proxy-agent/ProxyAgent.Connection.log`
Windows	`C:\WindowsAzure\ProxyAgent\Logs\ProxyAgent.Connection.log`

Recommended progression

Customers often aren't aware of the full extent of WireServer and Instance Metadata Service usage in their workload. Your VM's usage is a combination of any custom integrations that you create and the implementation details of any other software that you run. We recommend starting with a simpler configuration and iterating over time as you gain more information. In general, it's best to go in this order:

1. Enable MSP for WireServer and Instance Metadata Service, in `Audit` mode. This step enables you to confirm compatibility and learn more about your workload. It also generates valuable data for detecting threat actors.
2. Review the audit data. Pay close attention to any flagged requests. These requests might be signs of an ongoing attack or a workload incompatibility.
3. Remediate any problems. Under the default setup, anything that's flagged is a misuse of the metadata services. Contact your security team or the corresponding service owners as needed.
4. Switch from `Audit` mode to `Enforce` mode. Flagged requests are rejected to help protect your workload.
5. Experiment with advanced configuration. The default configuration mirrors pre-MSP security rules, which are often overly permissive. Defining a custom configuration for your workload enables you to further enhance security by applying the principle of least-privileged access.

Don't worry about following a perfect progression. *Any* level of MSP enablement greatly improves your security and helps protect against most real-world attacks.

Related content

- Deploy a VM or virtual machine scale set with MSP
- Enable MSP on an existing VM or virtual machine scale set
- Advanced configuration for MSP

Deploy a VM or virtual machine scale set with MSP

10/08/2025

This article explains how to enable the Metadata Security Protocol (MSP) feature while provisioning a new virtual machine (VM) or virtual machine scale set.

> ⚠ **Warning**
>
> Please note that on Windows, the users can enable GuestProxyAgent (GPA) from an ARM template. However, in Linux it's a two-step process. The VM/VMSS is provisioned first and only then can the GPA be configured.

Prerequisites

- Ensure that your image of choice is compatible.
- Familiarize yourself with the basic configuration options.

Deploy a VM with MSP

Use an ARM template

To deploy a VM with MSP via an Azure Resource Manager template (ARM template), update `proxyAgentSettings` and ensure that the minimum API version is `2024-03-01`. See the ARM template examples.

Use the REST API

To use the REST API, deploy a VM as you normally would, but with the MSP configuration also applied. Here's an example that references a linked advanced configuration:

HTTP

```
https://management.azure.com/subscriptions/{subscription-
id}/resourceGroups/{resource-group-
name}/providers/Microsoft.Compute/virtualMachines/{virtualMachine-Name}?api-
version=2024-03-01

{
  "name": "GPAWinVM",
```

```
  "location": "centraluseuap",
  "properties": {
    ...
    ...
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": true,
        "keyIncarnationId": 0,
        "wireServer": {
          "inVMAccessControlProfileReferenceId":
"/subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Micr
osoft.Compute/galleries/{galleryName}/inVMAccessControlProfiles/{wireServerProfile
Name}/versions/{version}"
        },
        "imds": {
          "inVMAccessControlProfileReferenceId":
"/subscriptions/{SubscriptionId}/resourceGroups/{ResourceGroupName}/providers/Micr
osoft.Compute/galleries/{galleryName}/inVMAccessControlProfiles/{imdsProfileName}/
versions/{version}"
        }
      }
    },
  }
}
```

Validate linked rules

After you deploy the VM, validate that the linked rules were applied to your VM. The steps for this validation in a newly provisioned VM are the same as the steps for an existing VM.

Deploy a virtual machine scale set with MSP

Using the same steps for a virtual machine scale set applies MSP to every VM within that scale set.

Enable MSP on an existing VM or virtual machine scale set

09/18/2025

This article explains the ways that you can enable Metadata Security Protocol (MSP) on an existing virtual machine (VM) or virtual machine scale set. You can enable MSP on a VM by using the Azure portal, an Azure Resource Manager template (ARM template), or the REST API.

Prerequisites

- Ensure that your image of choice is compatible.
- Familiarize yourself with the basic configuration options.

Enable MSP on a VM

Use the Azure portal

See Configure MSP via the portal.

Use an ARM template

See examples

With REST API

Enable MSP with both WireServer and Azure Instance Metadata Service protected in `Audit` mode:

HTTP

```
PATCH https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/{resource-group-name}/providers/Microsoft.Compute/virtualMachines/{virtualMachine-Name}?api-version=2024-03-01

{
  "properties": {
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": true,
        "wireServer": {
```

```
          "mode": "Audit"
        },
        "imds": {
          "mode": "Audit"
        }
      }
    },
  }
}
```

For Windows VMs, if `ProxyAgentSettings.Enabled` is `true`, the `Microsoft.CPlat.ProxyAgent.ProxyAgentWindows` extension is installed by default.

For Linux VMs, setting `ProxyAgentSettings.Enabled` to `true` doesn't implicitly install the Proxy Agent extension. To enable the Proxy Agent through the Proxy Agent extension, there must be an explicit `PUT` REST API call on the `Microsoft.CPlat.ProxyAgent.ProxyAgentLinux` extension. The purpose of the Proxy Agent extension is to enable, automatically update, and report the status of the Proxy Agent.

Validate linked rules

To validate that the linked rules were applied to your VM, check the VM instance view to confirm the status of the `Microsoft.CPlat.ProxyAgent.ProxyAgentWindows` extension for Windows and the `Microsoft.CPlat.ProxyAgent.ProxyAgentLinux` extension for Linux.

The `ComponentStatus/ProxyAgentStatus/succeeded` status should have an `imdsRuleId` value and a `wireServerRuleId` value. These values should map to the reference ID of `InVMAccessControlProfile`.

JSON


```
"extensions": [
    {
      "name": "AzureGuestProxyAgentExtension",
      "type": "Microsoft.CPlat.ProxyAgent.ProxyAgentWindows",
      "typeHandlerVersion": "1.0.21",
      "substatuses": [
        {
          "code": "ComponentStatus/ProxyAgentConnectionSummary/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "
[{\"userName\":\"SYSTEM\",\"ip\":\"169.254.169.254\",\"port\":80,\"processCmdLine\
":\"\\\"C:\\\\Packages\\\\Plugins\\\\Microsoft.Azure.Geneva.GenevaMonitoring\\\\2.
45.0.4\\\\GenevaMonitoringExtension.exe\\\" collectLogs\",\"responseStatus\":\"200
OK\",\"count\":344,\"userGroups\":
[],\"processFullPath\":\"C:\\\\Packages\\\\Plugins\\\\Microsoft.Azure.Geneva.Genev
```

```
aMonitoring\\\\2.45.0.4\\\\GenevaMonitoringExtension.exe\"]"
        },
        {
          "code": "ComponentStatus/ProxyAgentStatus/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "
{\"version\":\"1.0.21\",\"status\":\"SUCCESS\",\"monitorStatus\":
{\"status\":\"RUNNING\",\"message\":\"proxy_agent_status thread
started.\"},\"keyLatchStatus\":{\"status\":\"RUNNING\",\"message\":\"Found key
details from local and ready to use. - 122\",\"states\":
{\"imdsRuleId\":\"/SUBSCRIPTIONS/A53F7094-A16C-47AF-ABE4-
B05C05D0D79A/RESOURCEGROUPS/HUIYARG-
EASTUS2EUAP/PROVIDERS/MICROSOFT.COMPUTE/GALLERIES/GALLERYXX/INVMACCESSCONTROLPROFI
LES/WINDOWSIMDS/VERSIONS/1.3.0\",\"secureChannelState\":\"WireServer Audit -  IMDS
Audit\",\"keyGuid\":\"7512289d-6e5c-449b-9cbf-
06728c1c1e4a\",\"wireServerRuleId\":\"/SUBSCRIPTIONS/A53F7094-A16C-47AF-ABE4-
B05C05D0D79A/RESOURCEGROUPS/HUIYARG-
EASTUS2EUAP/PROVIDERS/MICROSOFT.COMPUTE/GALLERIES/GALLERYXX/INVMACCESSCONTROLPROFI
LES/WINDOWSWIRESERVER/VERSIONS/1.2.0\"}},\"ebpfProgramStatus\":
{\"status\":\"RUNNING\",\"message\":\"Started Redirector with eBPF maps -
79\"},\"proxyListenerStatus\":{\"status\":\"RUNNING\",\"message\":\"Started proxy
listener 127.0.0.1:3080, ready to accept request - 9\"},\"telemetryLoggerStatus\":
{\"status\":\"RUNNING\",\"message\":\"Telemetry event logger thread
started.\"},\"proxyConnectionsCount\":259356}"
        },
        {
          "code": "ComponentStatus/EbpfStatus/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "Ebpf Drivers successfully queried."
        }
      ],
      "statuses": [
        {
          "code": "ProvisioningState/succeeded",
          "level": "Info",
          "displayStatus": "Provisioning succeeded",
          "message": "Update Proxy Agent command output successfully",
          "time": "2024-09-24T16:42:59+00:00"
        }
      ]
    }
```


Enable MSP on a virtual machine scale set

The preceding steps for enabling MSP on a VM also apply to a virtual machine scale set.

Configure MSP via the portal

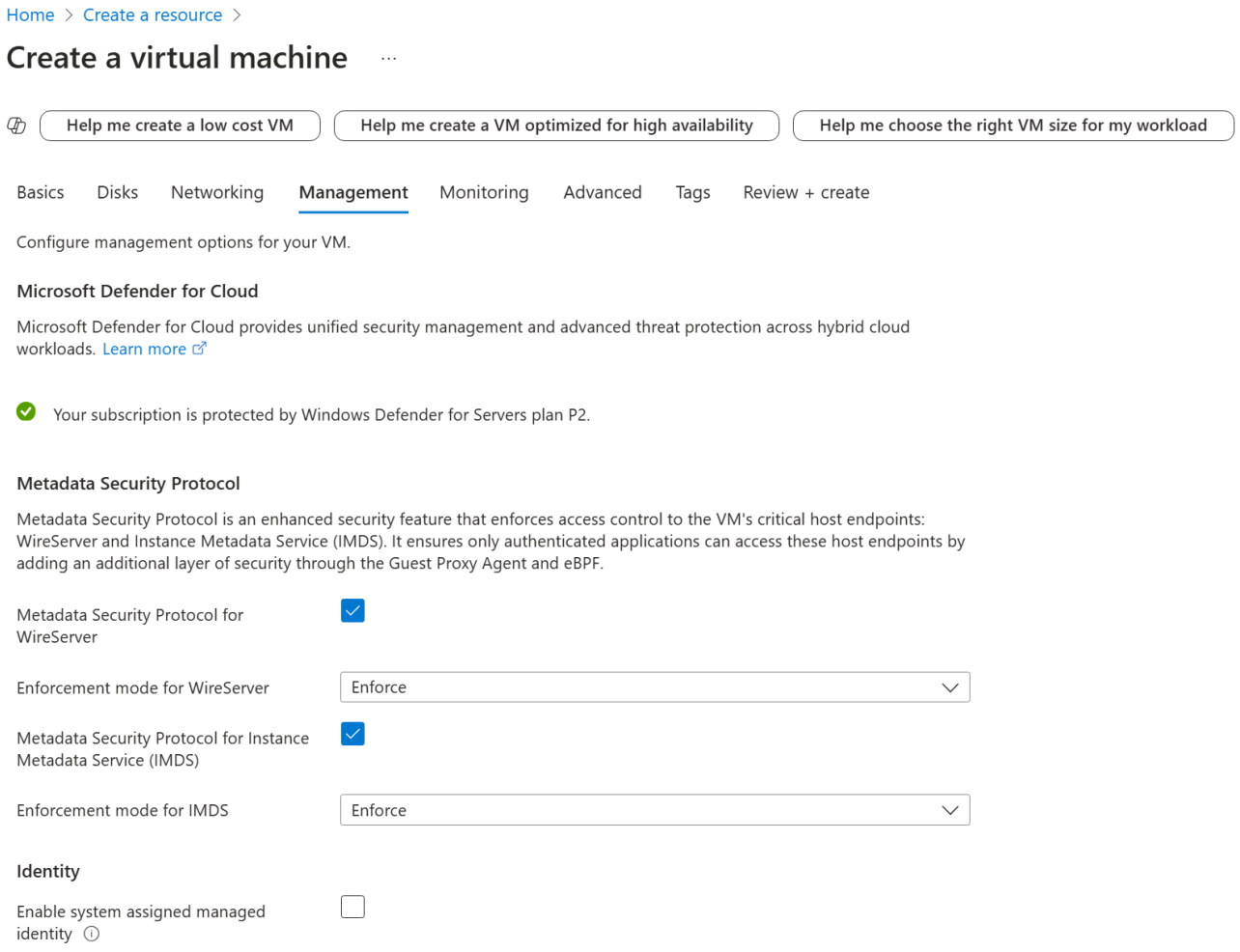
06/19/2025

Azure portal has preview support for configuring some aspects of Metadata Security Protocol (MSP).

> **Note**
>
> Currently, the portal is available only for subscriptions registered with the `Microsoft.Compute/ProxyAgentPreview` Azure Feature Exposure Control (AFEC) flag.

Deploy a VM with MSP

On the **Create a virtual machine** pane, make selections in the **Metadata Security Protocol** area.

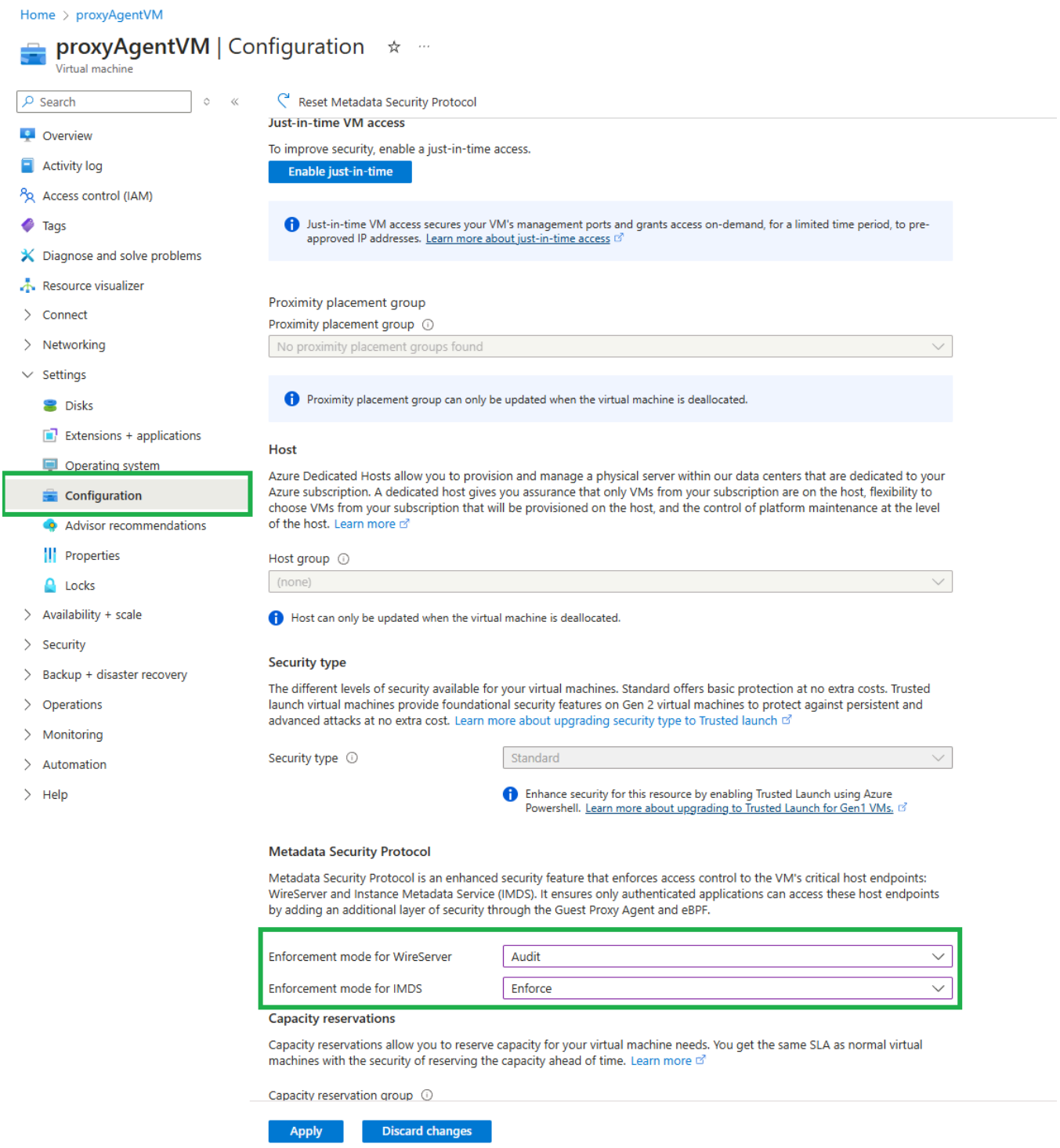


> **Note**

The portal currently supports only inline/basic configuration.

Enable MSP on an existing VM

Go to your virtual machine, and then go to **Settings** > **Configuration**. Make your selections in the **Metadata Security Protocol** area.



Disable MSP

05/29/2025

You can disable Metadata Security Protocol (MSP) by using the REST API and (partially) by using the Azure portal.

REST API

Set `securityProfile.proxyAgentSettings.enabled` to `false`:

HTTP

```
PATCH https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/{resource-group-name}/providers/Microsoft.Compute/virtualMachines/{virtualMachine-Name}?api-version=2024-03-01
{
  "properties": {
    "securityProfile": {
      "proxyAgentSettings": {
        "enabled": false,
      }
    },
  }
}
```

Azure portal

The portal doesn't allow disabling MSP completely at this time. But you can disable enforcement of security requirements to allow requests to flow normally.

Go to your virtual machine, and then go to **Settings** > **Configuration**. In the **Metadata Security Protocol** area, in the boxes for enforcement mode, select **Disabled**.

proxyAgentVM | Configuration

Virtual machine

Search

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Connect
Networking
Settings
Disks
Extensions + applications
Operating system
Configuration
Advisor recommendations
Properties
Locks
Availability + scale
Security
Backup + disaster recovery
Operations
Monitoring
Automation
Help

Reset Metadata Security Protocol

Just-in-time VM access secures your VM's management ports and grants access on-demand, for a limited time period, to pre-approved IP addresses. Learn more about just-in-time access

Proximity placement group

Proximity placement group

No proximity placement groups found

Proximity placement group can only be updated when the virtual machine is deallocated.

Host

Azure Dedicated Hosts allow you to provision and manage a physical server within our data centers that are dedicated to your Azure subscription. A dedicated host gives you assurance that only VMs from your subscription are on the host, flexibility to choose VMs from your subscription that will be provisioned on the host, and the control of platform maintenance at the level of the host. Learn more

Host group

(none)

Host can only be updated when the virtual machine is deallocated.

Security type

The different levels of security available for your virtual machines. Standard offers basic protection at no extra costs. Trusted launch virtual machines provide foundational security features on Gen 2 virtual machines to protect against persistent and advanced attacks at no extra cost. Learn more about upgrading security type to Trusted launch

Security type: Standard

Enhance security for this resource by enabling Trusted Launch using Azure Powershell. Learn more about upgrading to Trusted Launch for Gen1 VMs.

Metadata Security Protocol

Metadata Security Protocol is an enhanced security feature that enforces access control to the VM's critical host endpoints: WireServer and Instance Metadata Service (IMDS). It ensures only authenticated applications can access these host endpoints by adding an additional layer of security through the Guest Proxy Agent and eBPF.

Enforcement mode for WireServer: Disabled

Select enforce mode to enforce the access control

Enforcement mode for IMDS: Disabled

Select enforce mode to enforce the access control

Capacity reservations

Capacity reservations allow you to reserve capacity for your virtual machine needs. You get the same SLA as normal virtual

Reset a latched key

05/29/2025

If a virtual machine (VM) loses its copy of the latched key, a disk is migrated to a new VM. If any other key mismatch occurs, the VM can't access WireServer or Azure Instance Metadata Service. Resetting the key brings the VM back to a healthy state if the key is lost or unmatched between host and guest.

> **Note**
>
> The VM owner must request the key reset. Metadata services can't distinguish between an attacker or the Guest Proxy Agent requesting a reset when the key is lost, so resets can't be issued from within the VM.

Reset a VM's key

The platform always ensures that the `keyIncarnationId` value in the VM model matches the actual key in storage. Incrementing this value triggers a key reset. For more information, see MSP feature configuration.

HTTP

```
PATCH
https://management.azure.com/subscriptions/{subscription_id}/resourceGroups/{resou
rce_group_name}/providers/Microsoft.Compute/virtualMachines/{virtualMachine_Name}?
api-version=2024-03-01
{
  "properties": {
    "securityProfile": {
      "proxyAgentSettings": {
        "keyIncarnationId": 10
      }
    }
  }
}
```

Confirm the reset

Check the `AzureProxyAgentExtension` value of the VM instance view to confirm that the new key is generated. Your new `keyIncarnationId` value appears after the change propagates end to end.

JSON


```
"keyLatchStatus":{
    "status":"RUNNING",
    "message":"Found key details from local and ready to use. - 122",
    "states":{

"imdsRuleId":"/SUBSCRIPTIONS/{subscription_id}/RESOURCEGROUPS/{resource_group}/PRO
VIDERS/MICROSOFT.COMPUTE/GALLERIES/GALLERYXX/INVMACCESSCONTROLPROFILES/WINDOWSIMDS
/VERSIONS/{data_version}",
        "secureChannelState":"WireServer Enforce - IMDS Enforce",
        "keyIncarnationId":"10",
        "keyGuid":"e3882f98-da8d-4410-8394-06c23462781c",

"wireServerRuleId":"/SUBSCRIPTIONS/{subscription_id}/RESOURCEGROUPS/{resource_grou
p}/PROVIDERS/MICROSOFT.COMPUTE/GALLERIES/GALLERYXX/INVMACCESSCONTROLPROFILES/WINDO
WSWIRESERVER/VERSIONS/{data_version}"
        }
    }
```


> **Note**
>
> These requests are idempotent. However, if multiple requests are made with multiple `keyIncarnationId` values, there's no guarantee on the number and order of `keyIncarnationId` values that you observe. The final state reflects whichever request used the largest value.

Reset a virtual machine scale set's key

Key data is unique for each instance in a virtual machine scale set. The key must be reset on a per-instance basis.

You can reset a key only for a particular scale set's VM instance. You can't reset a key for all the scale set instances in a single API.

See the earlier instructions for resetting a VM's key, and substitute in your scale set instance's resource ID. For example:

`https://management.azure.com/subscriptions/{subscription_id}/resourceGroups/{resource_group_name}/providers/Microsoft.Compute/virtualMachineScaleSets/{vmScaleSet_name}/virtualMachines/{instance_id}?api-version=2024-03-01`

Convert audit logs to an allowlist

09/18/2025

With Metadata Security Protocol (MSP), you can define a custom role-based access control (RBAC) allowlist to help secure metadata service endpoints. The contents of the allowlist come from audit logs. A new resource type in Azure Compute Gallery, `InVMAccessControlProfile`, enables the allowlist.

To learn more about RBAC and the `InVMAccessControlProfile` resource type, see Advanced configuration for MSP.

Structure of an allowlist

An allowlist consists of:

- **Identities**: Processes on the machine.
- **Privileges**: Endpoints that the identities access.
- **Roles**: A grouping of privileges.
- **Role assignments**: Roles and the list of identities granted access for those roles.

Collect audit logs

If you enable MSP in `audit` or `enforce` mode, you can collect log data from virtual machine client via Azure Monitor.

- Windows: Azure monitoring agent collects Windows Events via Custom XPath `Windows Azure!*[System[Provider[@Name=`GuestProxyAgent`]]]`
- Linux: Collect Syslog events with Azure Monitor Agent by selecting `Log_DAEMON` and `LOG_DEBUG`

Query audit logs

Once the audit logs are collected as explained in the previous section, you can verify the logs:

Windows Kusto Query: Go to Log Analytics and run query

```
Kusto
```

```
/// Windows VMs
Event
| where TimeGenerated >ago(1d)
| where EventLog == "Windows Azure" and Source == "GuestProxyAgent"
| where _ResourceId startswith "/subscriptions/<your subscription
id>/resourcegroups/<your recrouce group>"
| where RenderedDescription  has  "processFullPath" and RenderedDescription  has
"runAsElevated" and RenderedDescription has "processCmdLine"
| extend json = parse_json(RenderedDescription)
| extend method = json.method, url = json.url, processFullPath =
json.processFullPath, username = json.userName, runAsElevated = json.runAsElevated
| extend userGroups = json.userGroups, ip = json.ip, port = json.port,
processCmdLine = json.processCmdLine
| project TimeGenerated, _ResourceId, url, ip, port, processFullPath, method,
username, runAsElevated, processCmdLine
```

Linux Kusto Query: Go to Log Analytics and run query

Kusto

```
/// Linux VMs
Syslog
| where TimeGenerated >ago(1d)
| where  ProcessName == "azure-proxy-agent"
| where _ResourceId startswith "/subscriptions/<your subscription
id>/resourcegroups/<your recrouce group>"
| where SyslogMessage  has  "processFullPath" and SyslogMessage  has
"runAsElevated" and SyslogMessage has "processCmdLine"
| extend message = substring(SyslogMessage, indexof(SyslogMessage, "{"))
| extend json = parse_json(message)
| extend method = json.method, url = json.url, processFullPath =
json.processFullPath, username = json.userName, runAsElevated = json.runAsElevated
| extend userGroups = json.userGroups, ip = json.ip, port = json.port,
processCmdLine = json.processCmdLine
| project TimeGenerated, _ResourceId, url, ip, port, processFullPath, method,
username, runAsElevated, processCmdLine
```

If you own the Azure VM, you can get the file logs inside the Azure VM by following these steps

1. Find the Proxy Agent json config file:
    - Windows VM: `GuestProxyAgent.json` under the folder as `GuestProxyAgent windows service`
    - Linux VM: `etc/azure/proxy-agent.json` file
2. Turn on the file log by updating setting `logFolder` in the json config file. Set it to

- `%SYSTEMDRIVE%\\WindowsAzure\\ProxyAgent\\Logs` for Windows VMs
- `/var/log/azure-proxy-agent` for Linux VMs.

3. Restart Service

- `GuestProxyAgent` for Windows VMs
- `azure-proxy-agent` for Linux VMs4

4. Wait for `ProxyAgent.Connection.log` to populate while customer services is running.
5. Pull the ProxyAgent connection file logs from the Azure VMs

    - Windows: `C:\WindowsAzure\ProxyAgent\Logs\ProxyAgent.Connection.log`
    - Linux: `/var/log/azure-proxy-agent/ProxyAgent.Connection.log`

Convert logs to rules

To create an allowlist, you can use an automated method or a manual method.

Generate an allowlist automatically

You can use an allowlist generator tool to generate the access control rules. The tool helps parse the audit logs and provides a UI to generate the rules.

1. Download and run the allowlist generator tool. On the latest release page, under **Assets**, select `allowListTool.exe`.

    The tool parses the `ProxyAgentConnection` logs and displays the current privileges and identities on the VM.

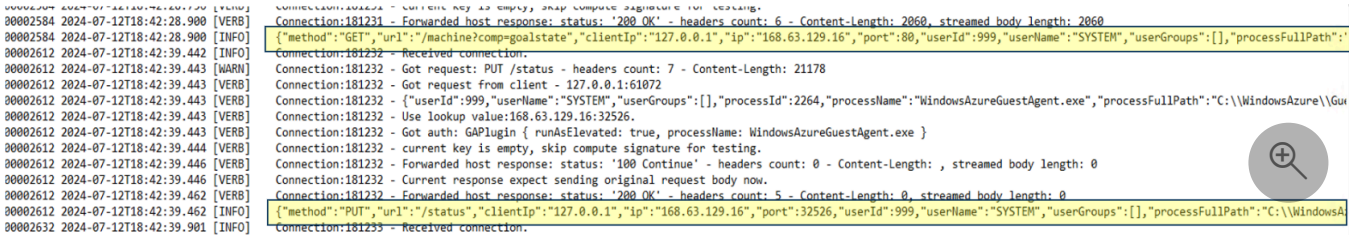
2. Create roles and role assignments:

    - To create a role, select a grouping of privileges and give the role a descriptive name.
    - To create a role assignment, select a role and a grouping of identities. These identities can access the privileges grouped in that role. Give the role assignment a descriptive name.

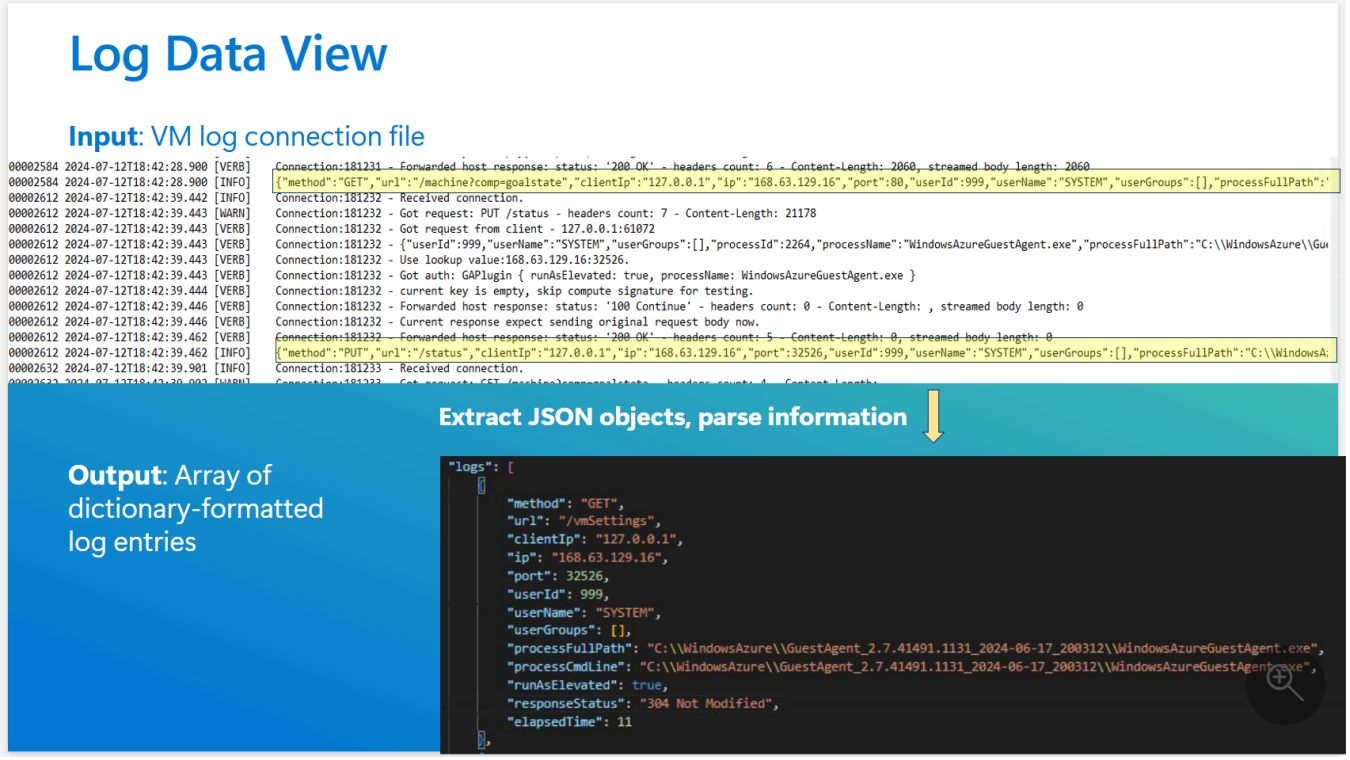
Manually create an allowlist

After you enable a VM with MSP in `Audit` or `Enforce` mode, the proxy agent captures all the requests being made to the host endpoints.

In the connection logs, you can analyze the applications that are making the requests to the Azure Instance Metadata Service or WireServer endpoints.

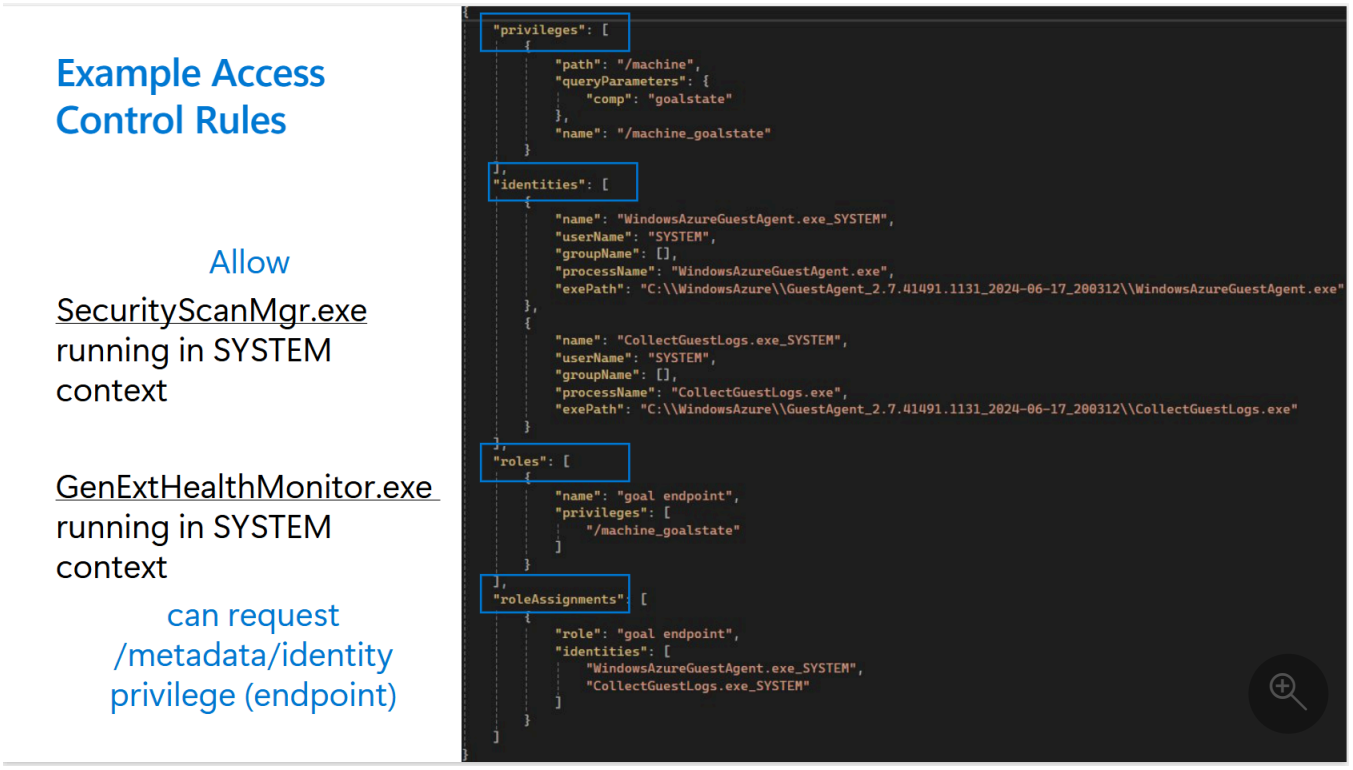


The following example shows the format of the captured JSON.



In the log file, you can identify the endpoints that you want to secure. These endpoints appear in `privileges` in the final `InVMAccessControlProfile` instance. You can also identify the identities (`identities`) that should have access.

A simple rules schema might look like the following example.



Create an InVMAccessControlProfile instance by using an ARM template

1. Create a new private gallery in Azure Compute Gallery.

2. Create an `InVMAccessControlProfile` definition with parameters for:

   - Gallery name to store in (from step 1)
   - Profile name
   - OS type
   - Host endpoint type (WireServer or Instance Metadata Service)

3. Create a specific version.

Sample InVMAccessControlProfile

Here's a sample `InVMAccessControlProfile` instance:

```
"properties": {
    "mode": "Enforce",
    "defaultAccess": "Allow",
    "rules": {
      "privileges": [
        {
```

```
      "name": "GoalState",
      "path": "/machine",
      "queryParameters": {
        "comp": "goalstate"
      }
    }
  ],
  "roles": [
    {
      "name": "Provisioning",
      "privileges": [
        "GoalState"
      ]
    },
    {
      "name": "ManageGuestExtensions",
      "privileges": [
        "GoalState"
      ]
    },
    {
      "name": "MonitoringAndSecret",
      "privileges": [
        "GoalState"
      ]
    }
  ],
  "identities": [
    {
      "name": "WinPA",
      "userName": "SYSTEM",
      "exePath": "C:\\Windows\\System32\\cscript.exe"
    },
    {
      "name": "GuestAgent",
      "userName": "SYSTEM",
      "processName": "WindowsAzureGuestAgent.exe"
    },
    {
      "name": "WaAppAgent",
      "userName": "SYSTEM",
      "processName": "WaAppAgent.exe"
    },
    {
      "name": "CollectGuestLogs",
      "userName": "SYSTEM",
      "processName": "CollectGuestLogs.exe"
    },
    {
      "name": "AzureProfileExtension",
      "userName": "SYSTEM",
      "processName": "AzureProfileExtension.exe"
    },
    {
      "name": "AzurePerfCollectorExtension",
```

```
          "userName": "SYSTEM",
          "processName": "AzurePerfCollectorExtension.exe"
        },
        {
          "name": "WaSecAgentProv",
          "userName": "SYSTEM",
          "processName": "WaSecAgentProv.exe"
        }
      ],
      "roleAssignments": [
        {
          "role": "Provisioning",
          "identities": [
            "WinPA"
          ]
        },
        {
          "role": "ManageGuestExtensions",
          "identities": [
            "GuestAgent",
            "WaAppAgent",
            "CollectGuestLogs"
          ]
        },
        {
          "role": "MonitoringAndSecret",
          "identities": [
            "AzureProfileExtension",
            "AzurePerfCollectorExtension",
            "WaSecAgentProv"
          ]
        }
      ]
    },
```

Using PowerShell

If you're using PowerShell to generate an `InVMAccessControlProfile`, ensure that you have the minimum version of 10.1.0 of PowerShell

Follow the step-by-step guide below to generate an `InVMAccessControlProfile`:

1. Log in to your Azure Account

PowerShell

```
Connect-AzAccount
```

2. Create the Resource Group where the private gallery is created. You can skip this step if you already have a Resource Group created.

PowerShell

```
$resourceGroup = "MyResourceGroup4"
$location = "EastUS2EUAP"
New-AzResourceGroup -Name $resourceGroup -Location $location
```

3. Create a private gallery. This gallery is used as a container for the `InVMAccessControlProfile` artifact

PowerShell

```
$galleryName = "MyGallery4"
New-AzGallery -ResourceGroupName $resourceGroup -GalleryName $galleryName -
Location $location -Description "My custom image gallery"
```

4. Create the `InVMAccessControlProfile` artifact in the private gallery created in the previous step. Click here to learn more about the various parameters for this artifact.

PowerShell

```
$InVMAccessControlProfileName= "testInVMAccessControlProfileP"

New-AzGalleryInVMAccessControlProfile -ResourceGroupName  $resourceGroup  -
GalleryName $galleryName   -GalleryInVMAccessControlProfileName
$InVMAccessControlProfileName -Location $location -OsType "Windows" -
ApplicableHostEndPoint "WireServer" -Description "this test1"
```

5. Get Gallery `InVMAccessControlProfile`

PowerShell

```
$inVMAccessCP=Get-AzGalleryInVMAccessControlProfile -ResourceGroupName
$resourceGroup  -GalleryName $galleryName   -GalleryInVMAccessControlProfileName
$InVMAccessControlProfileName
```

![Screenshot of the output for Get command for InVMAccessControlProfile.](../images/

6. Update Gallery `InVMAccessControlProfile` Once the `InVMAccessControlProfile` is created, the only attribute editable is the description. For any other changes, create a new artifact.

To update the description:

PowerShell

```
Update-AzGalleryInVMAccessControlProfile -ResourceGroupName  $resourceGroup  -
GalleryName $galleryName   -GalleryInVMAccessControlProfileName
```

```
$InVMAccessControlProfileName -Location $location -Description "this test2"
```

```
PS C:\Users\       > $inVMAccessCP=Get-AzGalleryInVMAccessControlProfile -ResourceGroupName  $resourceGroup  -GalleryNam
e $galleryName   -GalleryInVMAccessControlProfileName $InVMAccessControlProfileName
PS C:\Users\       > $inVMAccessCP | ConvertTo-Json -Depth 10
{
  "Id": "/subscriptions/                                    /resourceGroups/MyResourceGroup4/providers/Microsoft.Compute
/galleries/MyGallery4/inVMAccessControlProfiles/testInVMAccessControlProfileP",
  "Name": "testInVMAccessControlProfileP",
  "Type": "Microsoft.Compute/galleries/inVMAccessControlProfiles",
  "Location": "EastUS2EUAP",
  "Tags": {},
  "Properties": {
    "Description": "this test2",
    "OsType": "Windows",
    "ApplicableHostEndpoint": "WireServer",
    "ProvisioningState": "Succeeded"
  }
}
```

7. Create `InVMAccessControlProfileVersion`

To create an InVMAccessControlProfileVersion, a payload is required. Since these payloads can be large, especially due to the rules property, it's not practical to use a single PowerShell command to create the entire resource in one go. The rules property in any version payload consists of four arrays: privileges, roles, identities, and roleAssignments. These arrays can make the payload large and complex. To simplify this process, we introduced the GalleryInVMAccessControlProfileVersionConfig PowerShell object. You can learn more about it here.

This object allows you to incrementally build the configuration using various commands to add or remove rule properties. Once the configuration object is ready, you can use it to create an `InVMAccessControlProfileVersion`, described in the upcoming sections.

Alternatively, if you already have the rules property as a JSON string and prefer not to use the configuration object, you can skip these steps and create the InVMAccessControlProfileVersion using an ARM template deployment, which is also covered later in the section.

Payload for reference:

JSON

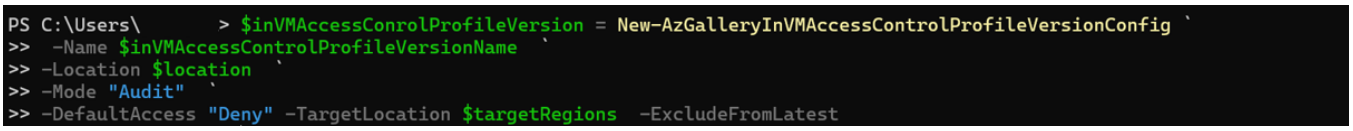
```
{
  "name": "1.0.0",
  "location": "East US 2 EUAP",
  "properties": {
    "mode": "Audit",
    "defaultAccess": "Deny",
    "rules": {
      "privileges": [
        {
          "name": "GoalState",
          "path": "/machine",
          "queryParameters": {
            "comp": "goalstate"
          }
```

```
        }
      ],
      "roles": [
        {
          "name": "Provisioning",
          "privileges": [
            "GoalState"
          ]
        }
      ],
      "identities": [
        {
          "name": "WinPA",
          "userName": "SYSTEM",
          "groupName": "Administrators",
          "exePath": "C:\\Windows\\System32\\cscript.exe",
          "processName": "cscript"
        }
      ],
      "roleAssignments": [
        {
          "role": "Provisioning",
          "identities": [
            "WinPA"
          ]
        }
      ]
    },
    "targetLocations": [
      {
        "name": "East US 2 EUAP"
      }
    ],
    "excludeFromLatest": false
  }
}
```

8. Create `InVMAccessControlProfileVersion` Config

PowerShell

```
$inVMAccessControlProfileVersionName= "1.0.0"
$targetRegions= @("EastUS2EUAP", "CentralUSEUAP")
$inVMAccessConrolProfileVersion = New-
AzGalleryInVMAccessControlProfileVersionConfig `
 -Name $inVMAccessControlProfileVersionName  `
-Location $location  `
-Mode "Audit"  `
-DefaultAccess "Deny" -TargetLocation $targetRegions  -ExcludeFromLatest
```



Run this command to add each privilege:

PowerShell

```
Add-AzGalleryInVMAccessControlProfileVersionRulesPrivilege `
  -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
  -PrivilegeName "GoalState" `
  -Path "/machine" `
  -QueryParameter @{ comp = "goalstate" }
```

To remove a privilege:

PowerShell

```
Remove-AzGalleryInVMAccessControlProfileVersionRulesPrivilege `
 -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
 -PrivilegeName "GoalState2"
```

Run this command to add each role:

PowerShell

```
Add-AzGalleryInVMAccessControlProfileVersionRulesRole `
 -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
 -RoleName "Provisioning" `
 -Privilege @("GoalState")
```

Run this command to remove roles:

PowerShell

```
Remove-AzGalleryInVMAccessControlProfileVersionRulesRole `
 -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
 -RoleName "Provisioning2"
```

Add `RulesIdentity`:

PowerShell

```
Add-AzGalleryInVMAccessControlProfileVersionRulesIdentity `
  -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
  -IdentityName "WinPA" `
  -UserName "SYSTEM" `
```

```
  -GroupName "Administrators" `
  -ExePath "C:\Windows\System32\cscript.exe" `
  -ProcessName "cscript"
```

Remove `RulesIdentity`:

PowerShell

```
Remove-AzGalleryInVMAccessControlProfileVersionRulesIdentity `
  -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
  -IdentityName "WinPA2"
```

Run this command to add each Role Assignment:

PowerShell

```
Add-AzGalleryInVMAccessControlProfileVersionRulesRoleAssignment `
  -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
  -Role "Provisioning" `
  -Identity @("WinPA")
```

Remove Role Assignment:

PowerShell

```
Remove-AzGalleryInVMAccessControlProfileVersionRulesRoleAssignment `
  -GalleryInVmAccessControlProfileVersion $inVMAccessConrolProfileVersion `
  -Role "Provisioning2"
```

9. Create Gallery `InVMAccessControlProfileVersion`

PowerShell

```
New-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName $resourceGroup -
GalleryName $galleryName -GalleryInVMAccessControlProfileName
$InVMAccessControlProfileName   -GalleryInVmAccessControlProfileVersion
$inVMAccessConrolProfileVersion
```

```
PS C:\myWorkingDirectory\MSPGallery_v1> New-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName $resourceGroup -
GalleryName $galleryName -GalleryInVMAccessControlProfileName   $InVMAccessControlProfileName   -GalleryInVmAccessContro
lProfileVersion $inVMAccessConrolProfileVersion

Id                : /subscriptions/                                    /resourceGroups/MyResourceGroup/providers/Micros
                    oft.Compute/galleries/MyGallery/inVMAccessControlProfiles/testInVMAccessControlProfileP/versions/1.
                    0.0
Name              : 1.0.0
Type              : Microsoft.Compute/galleries/inVMAccessControlProfiles
Location          : EastUS2EUAP
Tags              : {}
TargetLocations   : {eastus2euap, centraluseuap}
ExcludeFromLatest : False
PublishedDate     : 19-06-2025 18:20:10
ProvisioningState : Succeeded
ReplicationStatus : Microsoft.Azure.Management.Compute.Models.ReplicationStatus
Mode              : Audit
DefaultAccess     : Deny
Rules             : Microsoft.Azure.Management.Compute.Models.AccessControlRules
```

10. Get `InVMAccessControlProfileVersion`

PowerShell

```
$ver = Get-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName
$resourceGroup -GalleryName $galleryName -GalleryInVMAccessControlProfileName
$InVMAccessControlProfileName `
 -GalleryInVMAccessControlProfileVersionName  $inVMAccessControlProfileVersionName
 $ver | ConvertTo-Json -Depth 10
```

11. Update `InVMAccessControlProfileVersion`

It's recommended to create a new `InVMAccessControlProfileVersion` since most parameters can't be updated. Here's an example:

PowerShell

```
$targetRegions= @("EastUS2EUAP")

$ver = Get-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName
$resourceGroup -GalleryName $galleryName -GalleryInVMAccessControlProfileName
$InVMAccessControlProfileName `
 -GalleryInVMAccessControlProfileVersionName  $inVMAccessControlProfileVersionName


Update-AzGalleryInVMAccessControlProfileVersion `
  -GalleryInVmAccessControlProfileVersion $ver `
-TargetLocation $targetRegions -ExcludeFromLatest $true
```

```
PS C:\Users\       > Update-AzGalleryInVMAccessControlProfileVersion `
>>   -GalleryInVmAccessControlProfileVersion $ver `
>> -TargetLocation $targetRegions -ExcludeFromLatest $true

Id                : /subscriptions/                                    /resourceGroups/MyResourceGroup4/providers/Mic
                    rosoft.Compute/galleries/MyGallery4/inVMAccessControlProfiles/testInVMAccessControlProfileP/versi
                    ons/1.0.0
Name              : 1.0.0
Type              : Microsoft.Compute/galleries/inVMAccessControlProfiles
Location          : EastUS2EUAP
Tags              : {}
TargetLocations   : {eastus2euap}
ExcludeFromLatest : True
PublishedDate     : 11-07-2025 06:00:12
ProvisioningState : Succeeded
ReplicationStatus : Microsoft.Azure.Management.Compute.Models.ReplicationStatus
Mode              : Audit
DefaultAccess     : Deny
Rules             :
    Privileges    : Microsoft.Azure.Management.Compute.Models.AccessControlRulesPrivilege
    Roles         : Microsoft.Azure.Management.Compute.Models.AccessControlRulesRole
    Identities    : Microsoft.Azure.Management.Compute.Models.AccessControlRulesIdentity
    RoleAssignments : Microsoft.Azure.Management.Compute.Models.AccessControlRulesRoleAssignment

PS C:\Users\       > $ver = Get-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName $resourceGroup -GalleryName
```

12. Delete `InVMAccessControlProfileVersion`

PowerShell

```
Remove-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName $resourceGroup
-GalleryName $galleryName -GalleryInVMAccessControlProfileName
$InVMAccessControlProfileName `
 -GalleryInVMAccessControlProfileVersionName  $inVMAccessControlProfileVersionName
```

```
PS C:\Users\       > Remove-AzGalleryInVMAccessControlProfileVersion -ResourceGroupName $resourceGroup -GalleryName $gal
leryName -GalleryInVMAccessControlProfileName   $InVMAccessControlProfileName `
>>  -GalleryInVMAccessControlProfileVersionName  $inVMAccessControlProfileVersionName

Remove-AzGalleryInVMAccessControlProfileVersion operation
This cmdlet will remove the specified resource.  Do you want to continue?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): y

Name      : 5018c6a7-bebf-48cf-8482-677fb5dc406a
StartTime : 11-07-2025 12:07:17
EndTime   : 11-07-2025 12:08:24
Status    : Succeeded
Error     :
```

13. List all gallery `InVMAccessControlProfile`

PowerShell

```
Get-AzGalleryInVMAccessControlProfile -ResourceGroupName "myResourceGroup" -
GalleryName "myGallery"
```

```
PS C:\myWorkingDirectory\MSPGallery_v2> Get-AzGalleryInVMAccessControlProfile -ResourceGroupName  $resourceGroup  -Galle
ryName $galleryName

Id         : /subscriptions/                                    /resourceGroups/MyResourceGroup2/providers/Microsoft.Co
             mpute/galleries/MyGallery2/inVMAccessControlProfiles/testInVM2
Name       : testInVM2
Type       : Microsoft.Compute/galleries/inVMAccessControlProfiles
Location   : EastUS2EUAP
Tags       : {}
Properties : Microsoft.Azure.Management.Compute.Models.GalleryInVMAccessControlProfileProperties

Id         : /subscriptions/                                    /resourceGroups/MyResourceGroup2/providers/Microsoft.Co
             mpute/galleries/MyGallery2/inVMAccessControlProfiles/testInVMAccessControlProfileP
Name       : testInVMAccessControlProfileP
Type       : Microsoft.Compute/galleries/inVMAccessControlProfiles
Location   : EastUS2EUAP
Tags       : {}
Properties : Microsoft.Azure.Management.Compute.Models.GalleryInVMAccessControlProfileProperties
```

Using CLI

Follow the step-by-step guide below to generate an `InVMAccessControlProfile`:

1. Log in to your Azure Account

```cli
az login
```

2. Create a resource group in your desired location

```cli
az group create  --resource-group ResourceGroupForINVM  --location eastus
```

3. Create a gallery that will house the `InVMAccessControlProfile` resource

```cli
 az sig create --resource-group ResourceGroupForINVM --gallery-name MyGallery67 --
location eastus
```

3. Create an `InVMAccessControlProfile` under the gallery created in the prior step For additional details of commands: az sig in-vm-access-control-profile | Microsoft Learn

```cli
 az sig in-vm-access-control-profile create --resource-group ResourceGroupForINVM
--gallery-name MyGallery67 --name myInVMAccessControlProfileName --location eastus
--os-type Linux  --applicable-host-endpoint WireServer
```

Run this command to see additional details about the commands and properties

```cli
az sig in-vm-access-control-profile create --help
```

```
C:\Users\       >az sig in-vm-access-control-profile create --resource-group ResourceGroupForINVM --gallery-name MyGalle
ry67 --name myInVMAccessControlProfileName --location eastus --os-type Linux  --applicable-host-endpoint WireServer
{
  "id": "/subscriptions/                                  /resourceGroups/ResourceGroupForINVM/providers/Microsoft.Com
pute/galleries/MyGallery67/inVMAccessControlProfiles/myInVMAccessControlProfileName",
  "location": "eastus",
  "name": "myInVMAccessControlProfileName",
  "properties": {
    "applicableHostEndpoint": "WireServer",
    "osType": "Linux",
    "provisioningState": "Succeeded"
  },
  "resourceGroup": "ResourceGroupForINVM",
  "type": "Microsoft.Compute/galleries/inVMAccessControlProfiles"
}
```

4. Update the `InVMAccessControlProfile` You can only update the description of the `InVMAccessControlProfile`. If additional changes needed, delete the current `InVMAccessControlProfile` and create a new one

cli

```
az sig in-vm-access-control-profile update --resource-group ResourceGroupForINVM -
--gallery-name MyGallery67 --name myInVMAccessControlProfileName  --description
test
```

```
C:\Users\       >az sig in-vm-access-control-profile update --resource-group ResourceGroupForINVM --gallery-name MyGalle
ry67 --name myInVMAccessControlProfileName  --description test
{
  "id": "/subscriptions/                                  /resourceGroups/ResourceGroupForINVM/providers/Microsoft.Com
pute/galleries/MyGallery67/inVMAccessControlProfiles/myInVMAccessControlProfileName",
  "location": "eastus",
  "name": "myInVMAccessControlProfileName",
  "properties": {
    "applicableHostEndpoint": "WireServer",
    "description": "test",
    "osType": "Linux",
    "provisioningState": "Succeeded"
  },
  "resourceGroup": "ResourceGroupForINVM",
  "type": "Microsoft.Compute/galleries/inVMAccessControlProfiles"
}
```

5. Get operation for `InVMAccessControlProfile`

cli

```
az sig in-vm-access-control-profile show --resource-group ResourceGroupForINVM --
gallery-name MyGallery67 --name myInVMAccessControlProfileName
```

```
C:\Users\       >az sig in-vm-access-control-profile show --resource-group ResourceGroupForINVM --gallery-name MyGallery
67 --name myInVMAccessControlProfileName
{
  "id": "/subscriptions/                                  /resourceGroups/ResourceGroupForINVM/providers/Microsoft.Com
pute/galleries/MyGallery67/inVMAccessControlProfiles/myInVMAccessControlProfileName",
  "location": "eastus",
  "name": "myInVMAccessControlProfileName",
  "properties": {
    "applicableHostEndpoint": "WireServer",
    "description": "test",
    "osType": "Linux",
    "provisioningState": "Succeeded"
  },
  "resourceGroup": "ResourceGroupForINVM",
  "type": "Microsoft.Compute/galleries/inVMAccessControlProfiles"
}
```

6. List all `InVMAccessControlProfile` artifacts

This command shows the list of all `InVMAccessControlProfile` under a specific gallery

cli

```
az sig in-vm-access-control-profile show --resource-group ResourceGroupForINVM --
gallery-name MyGallery67
```

```
C:\myWorkingDirectory>az sig in-vm-access-control-profile list --resource-group ResourceGroupForINVM --gallery-name MyGa
llery67
[
  {
    "id": "/subscriptions/                                /resourceGroups/ResourceGroupForINVM/providers/Microsoft.C
ompute/galleries/MyGallery67/inVMAccessControlProfiles/myInVMAccessControlProfileName",
    "location": "eastus",
    "name": "myInVMAccessControlProfileName",
    "properties": {
      "applicableHostEndpoint": "WireServer",
      "description": "test",
      "osType": "Linux",
      "provisioningState": "Succeeded"
    },
    "resourceGroup": "ResourceGroupForINVM",
    "type": "Microsoft.Compute/galleries/inVMAccessControlProfiles"
  }
]
```

7. Create `InVMAccessControlProfileVersion`

You must provide a payload to create an `InVMAccessControlProfileVersion`. This payload can be large, especially because of the rule's property, which may contain extensive configurations. Instead of passing individual parts of the rules property directly, we've introduced a rules parameter that accepts a JSON file as an input. This approach simplifies the process and keeps the command clean and manageable. Here is an example rules.json file:

```
{
  "privileges": [
    {
      "name": "GoalState",
      "path": "/machine",
      "queryParameters": {
        "comp": "goalstate"
      }
    }
  ],
  "roles": [
    {
      "name": "Provisioning",
      "privileges": [
        "GoalState"
      ]
    }
  ],
  "identities": [
    {
      "name": "WinPA",
      "userName": "SYSTEM",
```

```
      "groupName": "Administrators",
      "exePath": "C:\\Windows\\System32\\cscript.exe",
      "processName": "cscript"
    }
  ],
  "roleAssignments": [
    {
      "role": "Provisioning",
      "identities": [
        "WinPA"
      ]
    }
  ]
}
```

See addtional command details here: az sig in-vm-access-control-profile-version | Microsoft Learn

Once you have create your version of rules.json file, use the following command to create the `InVMAccessControlProfileVersion`

cli

```
az sig in-vm-access-control-profile-version create \
  --resource-group ResourceGroupForINVM  \
  --gallery-name MyGallery67  \
  --profile-name myInVMAccessControlProfileName \
  --version-name 1.0.0 \
  --mode Audit \
  --default-access Deny  \
  --target-regions EastUS2EUAP  \
  --exclude-from-latest true \
  --rules @rules.json
```

8. Get `InVMAccessControlProfileVersion`

cli

```
az sig in-vm-access-control-profile-version show --resource-group
ResourceGroupForINVM --gallery-name MyGallery67 --profile-name
myInVMAccessControlProfileName --profile-version 1.0.0
```



9. List all `InVMAccessControlProfileVersion` artifacts

cli

```
az sig in-vm-access-control-profile-version list --resource-group
ResourceGroupForINVM --gallery-name MyGallery67 --profile-name
myInVMAccessControlProfileName
```

10. Delete `InVMAccessControlProfileVersion`

Before you delete any `InVMAccessControlProfileVersion` ensure it is not being used by any `InVMAccessControlProfile` on any Virtual Machine or Vitual Machine Scale Sets.

cli

```
az sig in-vm-access-control-profile-version delete --resource-group
ResourceGroupForINVM --gallery-name MyGallery67 --profile-name
myInVMAccessControlProfileName --profile-version 1.0.0
```

11. Delete `InVMAccessControlProfile`

cli

```
az sig in-vm-access-control-profile delete --resource-group ResourceGroupForINVM -
-gallery-name MyGallery67 --name myInVMAccessControlProfileName
```

Advanced configuration for MSP

05/29/2025

When you're working with Metadata Security Protocol (MSP), you can optionally define custom role-based access control (RBAC) allowlists for metadata service endpoints on a per-user and/or per-process basis. A new resource type in Azure Compute Gallery, `InVMAccessControlProfile`, enables these allowlists.

Motivation

Traditionally, metadata services treat the entire virtual machine (VM) as the trust boundary and allow any software running in the guest to access them. This access is overly permissive, especially if your workload isn't strictly implemented with a modern cloud-native architecture. Microsoft Security Response Center (MSRC) found that most security attacks can be prevented if organizations limit access to only the necessary software.

The `InVMAccessControlProfile` resource type enables you to define more granular control on the applications or users who can access the endpoints. You can use it to write rules like these examples:

- Only the Guest Proxy Agent (GPA) and SQL Server can access the network configuration.
- Only the monitoring extension can access tokens for managed identities.

The new resource type facilitates managing these configurations at scale. The rules that are appropriate for your VM depend on what software it's running. You can use `InVMAccessControlProfile` to define your configuration once and link it against all applicable VMs.

> **Note**
>
> We recommend treating these controls as *defense in depth* in your threat modeling. These controls can significantly enhance the security of a workload, but they're best used as extra protection instead of as core isolation mechanisms.
>
> In cloud-native multitenant workloads, especially if untrusted code is executed, hardware-backed or hypervisor-backed isolation provides greater protection. (For example, Azure Container Instances offers this kind of isolation.) MSP authorization rules complement hardware-backed isolation.

Properties

Expand table

Property	Type	Details
`mode`	Enumeration: `"Audit" | "Enforce" | "Disabled"`	See Inline configuration.
`defaultAccess`	Enumeration: `"Allow" | "Deny"`	Controls if an endpoint is accessible or restricted when no privileges for that resource are specified. See Privileges later in this article.
`rules`	`object`	The RBAC definition to apply. See Writing rules later in this article.

MSP allowlist (RBAC) concepts

The first step of access control is to verify the identity of the client. Normally, identity verification uses some type of a credential. MSP was designed to avoid introducing any breaking changes to clients, to maximize compatibility and make onboarding easier. Additionally, having clients upgraded with credentials introduces another layer of trust bootstrapping problems to solve.

Instead, MSP supports defining virtual identities against existing OS and process metadata. A VM's kernel keeps track of which user account a process belongs to, in addition to other metadata about its invocation. The GPA can identify which process made the HTTP request. As a result, the GPA can access that metadata to determine the identity of a caller and make authorization decisions.

Roles

You use roles to group multiple privileges into a named, reusable unit. Grouping privileges into roles improves organization and readability.

Role assignments

Role assignments pair an identity with a role. Requests that come from a process that matches the identity have all the privileges associated with that role.

Identities

With MSP, you can define a set of conditions to be evaluated against a property bag of process metadata. If all the conditions match, the process is considered to belong to this identity and is granted privileges.

The GPA supports writing *exact match* rules against these metadata items:

Expand table

Process metadata	Description
`username`	The human-readable name of the account that the process is running under.
`groupName`	The human-readable name of the group that contains the account that the process is running under. A user can belong to multiple groups. Conditions here are implemented as a *set contains* operation.
`processName`	The display name of the process. This name is purely defense in depth.
`exePath`	The full path of the running executable file. For some programs, the runtime (not the file) appears. An example of such a program is Python.

The ideal way to configure your workload is to run each application in a dedicated user account with uniquely scoped groups, and then write rules only against those properties. The user ID can't be spoofed, and there's no pattern matching to contend with. But because this method isn't practical in all workloads, we offer the other properties.

If you specify multiple properties, the matching is performed as a logical `AND`. For example, consider this identity definition:

JSON

```
{
    "name": "WebServer",
    "exePath": "/usr/sbin/apache2",
    "processName": "apache2",
    "groupName": "www-data"
}
```

The definition would generate the following validation:

c#

```
bool isWebServerIdentity = properties["exePath"] == "/usr/sbin/apache2"
    && properties["exePath"] == "apache2"
    && properties["groups"].contains("www-data");
```

Privileges

Privileges grant access to specific endpoints. Some endpoints are RESTful and can be expressed purely as a path. Others are influenced by query parameters.

Privileges are defined with a `name` value, a `path` value, and an optional set of `queryParameters` values:

JSON

```
"privileges": [
    {
        "name": "GoalState",
        "path": "/machine",
        "queryParameters": {"comp": "goalstate"}
    },
    {
        "name": "config",
        "path": "/machine",
        "queryParameters": {"comp": "config"}
    }
]
```

Comparison rules

- String matching is *case-insensitive*.
- If *no* query parameters are specified, the privilege grants access to *all* possible values on that path.

Default access rules

You can use default access (`defaultAccess`) modes to slowly lock down a workload. You can also use them to simplify rule authoring when you want to restrict access to only specific endpoints.

Expand table

`defaultAccess` mode	Behavior
`Deny`	If no explicit assignment exists to authorize the guest app to access the requested resource, the request is rejected.
`Allow`	If no rules exist for a specific privilege, access is allowed by default. If any privilege for that resource exists, the behavior reverts to denying by default for that resource.

- If query parameters are provided, the question "*Is a privilege defined?*" for determining if `defaultAccess` applies becomes: "*Does any rule match this path + every specified query parameter in the rule is present + matches in the request?*". Extra query parameters are ignored in this assessment.
- The request is rejected if duplicate query parameter keys in a request are invalid.

Writing rules

Specifying an allowlist via RBAC rules is optional. But if you specify one, you must supply *all* fields. For example, if you didn't have any `roles` values to define, you would still provide the empty array.

The easiest way to generate rules is by first running MSP in `Audit` mode. Then, use the audit logs to generate rules.

The following full Azure Instance Metadata Service schema expansion shows an example of the `rules` subsection:

JSON

```
{
  "defaultAccess": "allow",
  "mode": "enforce",
  "rules": {
    "privileges": [
      {
        "name": "Msi",
        "path": "/metadata/identity/oauth2/token"
      }
    ],
    "roles": [
      {
        "name": "MyWebApp",
        "privileges": [
          "Msi"
        ]
      }
    ],
    "identities": [
      {
        "name":"MyWebApp1",
        "processName":"WebApp1"
      }
    ],
    "roleAssignments": [
      {
        "role": "MyWebApp",
        "identities": [
          "MyWebApp1"
```

```
        ]
      }
    ]
  },
  "id": "D/uTRPye9b9xSUCRIgfRDF41zeY="
}
```

The following full WireServer schema expansion shows an example of the `rules` subsection:

JSON

```
{
  "defaultAccess": "allow",
  "mode": "enforce",
  "rules": {
    "privileges": [
      {
        "name": "GoalState",
        "path": "/machine",
        "queryParameters": {
          "comp": "goalstate"
        }
      }
    ],
    "roles": [
      {
        "name": "Provisioning",
        "privileges": [
          "GoalState"
        ]
      }
    ],
    "identities": [
      {
        "name": "WinPA",
        "exePath": "C:\\Windows\\OEM\\Unattend.wsf.exe",
        "processName": "winpa.exe",
        "userName": "SYSTEM"
      }
    ],
    "roleAssignments": [
      {
        "role": "Provisioning",
        "identities": [
          "WinPA"
        ]
      }
    ]
  },
  "id": "D/uTRPye9b9xSUCRIgfRDF41zeY="
}
```

Implicit and default configurations

The following RBAC definition is applied as default behavior for defense-in-depth access rules that existed before MSP:

JSON

```
{
    "imds": {
        "defaultAccess": "Allow",
        "mode": "Disabled",
    },
    "wireServer": {
        "defaultAccess": "Allow",
        "mode": "Enforce",
    }
}
```

Remember that WireServer allows only access from administrator or root processes. The `defaultAccess` value is still `Allow` here because this rule isn't user configurable. It's part of the GPA. The GPA checks if the client is an admin user before evaluating any RBAC rules for a WireServer request. If the client is not an admin user, the request is immediately rejected.

Troubleshoot MSP

05/29/2025

This article provides some basic troubleshooting guidance for using the Metadata Security Protocol (MSP) feature with virtual machines (VMs) or virtual machine scale sets. If you can't resolve your problem by using this guidance, contact the support team.

Deployment failed for a new VM or virtual machine scale set

Windows

In Windows, the platform automatically installs the necessary components: Guest Proxy Agent (GPA) and eBPF. If the installation fails:

1. Ensure that you're using a supported OS version.
2. Ensure that the failure code is MSP related.
3. Retry the deployment. Transient failures are a part of cloud computing.

Linux

Ensure that you're using a valid image:

- The GPA must be included in your image. That's a requirement for deploying with MSP enabled.
- Your cloud-init version must be a newer, MSP-aware version. If it's not, a race condition occurs between it and the GPA.

The exact failure depends on whether your image is misconfigured or a platform failure occurred:

Expand table

GPA installed	Cloud-init version	Expected failure	Cause
No	Earlier than 24.3	`Linux.OSProvisioningInternalError` `Linux.Cloud-Init` successfully reported ready for provisioning, but the platform failed to record success.	VM might fail provisioning even though cloud-init reports ready.
No	24.3 or later	`Linux.Cloud-Init` failed because `azure-proxy-agent` is not found.	Cloud-init reports failure to the platform after detecting that the GPA is not installed.
Yes	Earlier than 24.3	`Linux.OSProvisioningInternalError`	Cloud-init might report ready before the GPA is configured, because it's GPA-unaware. Failures might happen up to 100% of the time, depending on the scenario.
Yes	24.3 or later	Cloud-init reports that the GPA is unhealthy.	Any of these items could be the cause: eBPF setup failure, failure to enable cgroup v2, generic startup failure, failure to acquire a key.

MSP is enabled but not applied

To avoid service disruptions when you're enabling MSP on an existing VM or virtual machine scale set, protections aren't applied until the VM indicates that it successfully set up and acquired the long-lived key. This delay means the VM model can show that MSP is enabled. However, the GPA might still be unhealthy and indicate that protections aren't activated.

Confirm that the problem still exists

Sometimes the delay is longer than anticipated. To begin, check the status report of the GPA:

1. Get the `status.json` file:

   - For a Windows VM, the ProxyAgent service logs from inside the VMs because they have more details. The log folder is `C:\WindowsAzure\ProxyAgent\Logs`. ProxyAgent captures its overall status in `C:\WindowsAzure\ProxyAgent\Logs\status.json`.
   - For a Linux VM, the ProxyAgent service (`azure-proxy-agent`) logs from inside the VMs because they have more details. The log folder is `/var/log/azure-proxy-agent/`. ProxyAgent captures its overall status in `/var/log/azure-proxy-agent/status.json`.

2. Check `proxyAgentStatus` in the `status.json` file and ensure that the value is `SUCCESS`. If it's not `SUCCESS`, check these other statuses:

Expand table

Status	Expected value
`keyLatchStatus`	`RUNNING`
`ebpfProgramStatus`	`RUNNING`
`proxyListenerStatus`	`RUNNING`

GPA is missing or not running

Windows VM

Control Resource Plane (CRP) implicitly installs these Windows services by using guest extensions:

- GPA: `GuestProxyAgent`
- Kernel drivers for eBPF: `eBPFCore`, `NetEbpfExt`

Linux VM

For Linux VMs, the GPA must be included in the base image. Or you can explicitly add the GPA VM extension `Microsoft.CPlat.ProxyAgent.ProxyAgentLinux` before you enable the MSP feature.

Prerequisites are:

- Linux kernel 5.15 or later, which has all the required eBPF features.
- The cgroup v2 feature mounted by default. The GPA hooks up the cgroup/connect4 eBPF event.

When you enable MSP, CRP installs one service (`azure-proxy-agent`) in the VM.

```
# systemctl status azure-proxy-agent.service
● azure-proxy-agent.service - Microsoft Azure GuestProxyAgent
     Loaded: loaded (/lib/systemd/system/azure-proxy-agent.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-09-13 18:52:00 UTC; 1 week 5 days ago
   Main PID: 4040671 (azure-proxy-age)
      Tasks: 10 (limit: 19169)
     Memory: 491.2M
        CPU: 1d 8h 13min 2.321s
     CGroup: /system.slice/azure-proxy-agent.service
             └─4040671 /usr/sbin/azure-proxy-agent
```

GPA is running, but you can't set it up

eBPF setup failure

For eBPF setup to be successful, `eBPFCore` and `NetEbpfExt` must be in a `RUNNING` state.

Here's the code that shows the `RUNNING` state for a Windows VM:

```
c:\>sc query eBPFCore

SERVICE_NAME: eBPFCore
        TYPE               : 1  KERNEL_DRIVER
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Here's the code that shows the `RUNNING` state for a Linux VM:

```
c:\>sc query NetEbpfExt

SERVICE_NAME: NetEbpfExt
        TYPE               : 1  KERNEL_DRIVER
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Failure to load eBPF

The GPA uses some eBPF features that require Linux kernel 5.15 or later. If you're trying to enable the MSP feature on an unsupported Linux distribution, you might see a message similar to this example:

```
    "ebpfProgramStatus": {
      "status": "STOPPED",
      "message": "Failed to load program 'connect4' with error: the BPF_PROG_LOAD syscall failed. Verifier output: ; int
connect4(struct bpf_sock_addr *ctx)\n0: (bf) r6 = r1\n; __u64 cookie = bpf_get_socket_cookie(ctx);\n1: (85) call
bpf_get_socket_cookie#46\n2: (b7) r1 = 0\n; destination_entry entry = {0};\n3: (63) *(u32 *)(r10 -44) = r1\nlast_idx 3 first_idx
0\nregs=2 stack=0 before 2: (b7) r1 = 0\n4: (63) *(u32 *)(r10 -48) = r1\n5: (63) *(u32 *)(r10 -52) = r1\n;
entry.destination_ip.ipv4 = ctx->user_ip4;\n6: (61) r1 = *(u32 *)(r6 +4)\n; entry.destination_ip.ipv4 = ctx->user_ip4;\n7: (63) *
(u32 *)(r10 -56) = r1\n; entry.destination_port = ctx->user_port;\n8: (61) r1 = *(u32 *)(r6 +24)\n; entry.destination_port = ctx-
>user_port;\n9: (63) *(u32 *)(r10 -40) = r1\n; entry.protocol = ctx->protocol;\n10: (61) r1 = *(u32 *)(r6 +36)\n; entry.protocol
= ctx->protocol;\n11: (63) *(u32 *)(r10 -36) = r1\n12: (bf) r2 = r10\n; \n13: (07) r2 += -56\n; destination_entry *policy =
bpf_map_lookup_elem(&policy_map, &entry);\n14: (18) r1 = 0xffff8baa17403400\n16: (85) cal..."
    },
```

Try get the kernel version by using `uname -r`. If the version is earlier than 5.15, opt out of the MSP feature, upgrade your OS image to the latest version, and then enable MSP.

Failure to attach cgroup for redirection

`GuestProxyAgent` requires cgroup v2 mounted by default to hook up cgroup/connect4 eBPF events. If you try to enable the MSP feature on an unsupported Linux distribution, you might see a message like the following `status.json` example:

```
    "ebpfProgramStatus": {
      "status": "STOPPED",
      "message": "Failed to attach program 'connect4' with error: `bpf_link_create` failed"
    },
```

To resolve this problem, try these methods:

- Check if the current Linux VM supports cgroup v2 by using this command:

  ```
  grep cgroup /proc/filesystems
  ```

  If `nodev cgroup2` is listed, this OS supports cgroup v2. If `nodev cgroup2` isn't listed, opt out of the MSP feature, update to the latest OS version, and then enable MSP.

- Check if cgroup v2 is mounted by default. Use the following command:

  ```
  mount | grep cgroup2
  ```

  If no record appears, ask the VM owner to set it up by using the following command and then restart the VM:

```
sudo grubby --update-kernel=ALL --args="systemd.unified_cgroup_hierarchy=1"
```

Failure to acquire a key or rejected key

The GPA can't acquire a key if the platform no longer offers it. Actions like migrating a disk to a new VM, or deleting the VM's OS disk and replacing it, can cause this failure.

Follow the instructions for resetting a key in this article. Resetting the key allows the platform to offer a new one. The GPA periodically attempts to recover. After you complete the reset, the GPA automatically acquires a new key and goes back to a healthy state.

You need to reset a key

If your VM's long-lived key is lost, it no longer communicates with Azure Instance Metadata Service or WireServer. Without the key, a new one can't be safely issued to the VM automatically. Additionally, if the key is compromised in some way, it must be reset to maintain security.

To reset a key:

1. Refer to Reset a latched key for complete documentation on the `KeyIncarnationId` field of the `ProxyAgentSettings` section of the VM model.
2. Select a new `KeyIncarnationId` value, and apply it to the VM model by using a `PUT` operation.

> **Note**
>
> If you're sending multiple requests or otherwise using automation, you must ensure that the values you send are strictly monotonically increasing. Otherwise, the change might not be applied.

You got an error code

Expand table

Error code	Error message
`ProxyAgentNotSupportedInRegion`	"Creation of VMs or Virtual Machine Scale Sets with ProxyAgent feature isn't supported in this re
`SubscriptionNotEnabledForProxyAgentFeature`	"The subscription is not registered for the private preview of ProxyAgent feature."
`BadRequest`	"The property `securityProfile.proxyAgentSettings.wireServer.inVMAccessControlProfileReferer` `securityProfile.proxyAgentSettings.wireServer.mode`'" "The property `securityProfile.proxyAgentSettings.imds.inVMAccessControlProfileReferenceId`
`BadRequest`	"The value `securityProfile.proxyAgentSettings.keyIncarnationId` can only be incremented."
`BadRequest`	"The value of parameter `securityProfile.proxyAgentSettings.wireServer.mode` is invalid." "The value of parameter `securityProfile.proxyAgentSettings.imds.mode` is invalid."
`InvalidParameter`	"The resource id '{0}' isn't a valid gallery `inVMAccessControlProfile` reference. A gallery `inVMAcce` '/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Com
`BadRequest`/`GalleryInVMAccessControlProfileNotMatchOSDisk`	"Current Gallery `InVMAccessControlProfile` Version {0} supports OS {1}, while current OSDisk's O
`BadRequest`/`GalleryInVMAccessControlProfileNotMatchHostEndpointType`	"Current Gallery `InVMAccessControlProfile` Version {0} is for host endpoint {1}, while current refe

Error code	Error message
`InVMAccessControlProfileNotFound`	"The gallery `InVMAccessControlProfile` '{0}' isn't available. Verify that the `InVMAccessControlProf`
`InVMAccessControlProfileNotFound`	"Failed to prepare the `InVMAccessControlProfile` '{0}' metadata for one or more resources due to

Frequently asked questions about MSP

05/29/2025

This article answers common questions from customers who work with the Metadata Security Protocol (MSP) feature. If you don't see a question that can help you, contact the support team.

Usage

How do I check if the feature is enabled?

You can check MSP enablement programmatically by using the GET VM API to retrieve the virtual machine (VM) model. The `proxyAgentSettings` properties report MSP configuration.

Additionally, the `GuestProxyAgent` instance view in the VM runtime status reports the state of MSP from its perspective in the VM. If MSP is disabled, the value is `"Disabled"`.

How do I check component health?

The `GuestProxyAgent` instance view reports the status of the Guest Proxy Agent (GPA) and its dependencies. Each component's `status` value shows `RUNNING` when it's healthy.

Expand table

Component	Field
GPA	`keyLatchStatus.status`
eBPF integration	`ebpfProgramStatus.status`

Here's a full example:

JSON


```
{
  "version": "1.0.20",
  "status": "SUCCESS",
  "monitorStatus": {
    "status": "RUNNING",
    "message": "Monitor thread has not started yet."
  },
  "keyLatchStatus": {
    "status": "RUNNING",
    "message": "Found key details from local and ready to use. 122",
    "states": {
```

```
      "imdsRuleId":
"/SUBSCRIPTIONS/<guid>/RESOURCEGROUPS/<rg_name>/PROVIDERS/MICROSOFT.COMPUTE/GALLER
IES/GALLERYXX/INVMACCESSCONTROLPROFILES/...",
      "secureChannelState": "Wireserver Enforce - IMDS Enforce",
      "keyGuid": "e3882f98-da8d-4410-8394-06c23462781c",
      "WireserverRuleId":
"/SUBSCRIPTIONS/<guid>/RESOURCEGROUPS/<rg_name>/PROVIDERS/MICROSOFT.COMPUTE/GALLER
IES/GALLERYXX/INVMACCESSCONTROLPROFILES/..."
    }
  },
  "ebpfProgramStatus": {
    "status": "RUNNING",
    "message": "Started Redirector with eBPF maps - 143"
  },
  "proxyListenerStatus": {
    "status": "RUNNING",
    "message": "Started proxy listener 127.0.0.1:3080, ready to accept request -
27"
  },
  "telemetryLoggerStatus": {
    "status": "RUNNING",
    "message": "Telemetry event logger thread started."
  },
  "proxyConnectionsCount": 590
}
```


Can I provision a new Linux VM with MSP enabled without including the GPA in my image?

No. Linux provisioning doesn't have a step where the platform can install other software before the customer receives the VM. Also, most Linux customers don't want this model. It's more in line with the Windows philosophy.

For security reasons, MSP doesn't allow you to successfully provision a VM if you didn't establish a secure connection. The GPA is responsible for establishing the secure connection, so it needs to already be present.

If you don't want to use an image with the GPA included, you must create the VM first and then enable MSP afterward. During the preview, we monitor usage patterns and evaluate potential ways to support the scenario, by allowing the GPA to be installed as a VM extension rather than directly included.

Features

Is ARM64 supported?

ARM64 support is not available while the MSP feature is in preview.

Design

How is the latched key created, and how does it bind to the VM?

An Azure host generates the latched key and stores it as platform data. The key is unique to each VM. You can use it only on that VM's private connection to WireServer and Azure Instance Metadata Service.

How is the allowlist (policy) provisioned?

The allowlist is defined as part of the VM model. The MSP feature configuration article explains how to create an allowlist.

The Guest Proxy Agent retrieves this policy periodically from WireServer, like any other VM metadata. The allowlist allows users to centrally manage the policy in the same way that they would manage other VM settings. It also prevents users within the VM from tampering with it.

Where is the allowlist enforced (GPA or WireServer)?

Both components play a role in enforcement. MSP uses a shared responsibility model:

- WireServer and Instance Metadata Service are responsible for ensuring that only the trusted delegate (the GPA), and clients that the trusted delegate endorses, can access the VM's metadata and secrets.

  Metadata services can't, and shouldn't, perform VM introspection. They alone can't determine which software within a VM made a request.

- The trusted delegate (the GPA) is responsible for determining the identity of clients and endorsing requests.

  The GPA can rely on the OS kernel to authoritatively identify which process within the VM made a request.

The most important element is that MSP is a *default-closed* model, whereas other security measures (like in guest firewall rules) are *default-open*. If the GPA is down or otherwise bypassed, WireServer rejects any requests because the latched key doesn't endorse them.

Why is this feature opt-in and not required or automatically enabled?

Nearly every infrastructure as a service (IaaS) VM or virtual machine scale set in Azure uses Instance Metadata Service. The diversity of workloads requires flexibility. Although the design of MSP abstracts away most breaking changes, if a workload is ultimately dependent on a less secure configuration, enforcing stronger security breaks the workload.

Why does MSP require a new agent instead of using an Azure Guest Agent update?

The GPA has different responsibilities than the Azure Guest Agent. The GPA also has higher performance requirements.

Get Virtual Machine usage metrics using the REST API

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This example shows how to retrieve the CPU usage for a Linux Virtual Machine using the Azure REST API.

Complete reference documentation and samples for the REST API are available in the Azure Monitor REST reference article.

Build the request

Use the following GET request to collect the Percentage CPU metric from a virtual machine.

HTTP

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmname}/prov
iders/microsoft.insights/metrics?api-version=2018-01-
01&metricnames=Percentage%20CPU&timespan=2018-06-05T03:00:00Z/2018-06-
07T03:00:00Z
```

Request headers

The following headers are required:

Expand table

Request header	Description
Content-Type:	Required. Set to `application/json`.
Authorization:	Required. Set to a valid `Bearer` access token.

URI parameters

Expand table

Name	Description
subscriptionId	The subscription ID that identifies an Azure subscription. If you have multiple subscriptions, see Working with multiple subscriptions.
resourceGroupName	The name of the Azure resource group associated with the resource. You can get this value from the Azure Resource Manager API, CLI, or the portal.
vmname	The name of the Azure Virtual Machine.
metricnames	Comma-separated list of valid Load Balancer metrics.
api-version	The API version to use for the request. This document covers api-version `2018-01-01`, included in the above URL.
timespan	String with the following format `startDateTime_ISO/endDateTime_ISO` that defines the time range of the returned metrics. This optional parameter is set to return a day's worth of data in the example.

Request body

No request body is needed for this operation.

Handle the response

Status code 200 is returned when the list of metric values is returned successfully. A full list of error codes is available in the reference documentation.

Example response

JSON


```
{
    "cost": 0,
    "timespan": "2018-06-08T23:48:10Z/2018-06-09T00:48:10Z",
    "interval": "PT1M",
    "value": [
        {
            "id":
"/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmname}/providers/microsoft.insights/metrics?api-version=2018-01-01&metricnames=Percentage%20CPU",
            "type": "Microsoft.Insights/metrics",
            "name": {
```

```
                "value": "Percentage CPU",
                "localizedValue": "Percentage CPU"
            },
            "unit": "Percent",
            "timeseries": [
                {
                    "metadatavalues": [],
                    "data": [
                        {
                            "timeStamp": "2018-06-08T23:48:00Z",
                            "average": 0.44
                        },
                        {
                            "timeStamp": "2018-06-08T23:49:00Z",
                            "average": 0.31
                        },
                        {
                            "timeStamp": "2018-06-08T23:50:00Z",
                            "average": 0.29
                        },
                        {
                            "timeStamp": "2018-06-08T23:51:00Z",
                            "average": 0.29
                        },
                        {
                            "timeStamp": "2018-06-08T23:52:00Z",
                            "average": 0.285
                        } ]
                } ]
        } ]
}
```


Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure boot diagnostics

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Boot diagnostics is a debugging feature for Azure virtual machines (VM) that allows diagnosis of VM boot failures. Boot diagnostics enables a user to observe the state of their VM as it is booting up by collecting serial log information and screenshots.

Boot diagnostics storage account

When you create a VM in Azure portal, boot diagnostics is enabled by default. The recommended boot diagnostics experience is to use a managed storage account, as it yields significant performance improvements in the time to create an Azure VM. Using a managed storage account eliminates the time needed to create a user storage account to store the boot diagnostics data. When using a managed storage account: The managed storage account is not directly accessible, and the boot diagnostics data blobs (which comprise of logs and snapshot images) stored in the managed storage account are not currently billed.

An alternative boot diagnostic experience is to use a custom storage account. A user can either create a new storage account or use an existing one. When the storage firewall is enabled on the custom storage account (**Enabled from all networks** option isn't selected), you must:

- Make sure that access through the storage firewall is allowed for the Azure platform to publish the screenshot and serial log. To do this, go to the custom boot diagnostics storage account in the Azure portal and then select **Networking** from the **Security + networking** section. Check if the **Allow Azure services on the trusted services list to access this storage account** checkbox is selected.
- Allow storage firewall for users to view the boot screenshots or serial logs. To do this, add your network or the client/browser's Internet IPs as firewall exclusions. For more information, see Configure Azure Storage firewalls and virtual networks.

To configure the storage firewall for Azure Serial Console, see Use Serial Console with custom boot diagnostics storage account firewall enabled.

> ⓘ **Note**
>
> The custom storage account associated with boot diagnostics requires the storage account and the associated virtual machines reside in the same region and subscription. To view the boot diagnostics data, users must also have the correct level of view and read permissions.

Boot diagnostics view

Go to the virtual machine blade in the Azure portal, the boot diagnostics option is under the *Help* section in the Azure portal. Selecting boot diagnostics display a screenshot and serial log information. The serial log contains kernel messaging and the screenshot is a snapshot of your VMs current state. Based on if the VM is running Windows or Linux determines what the expected screenshot would look like. For Windows, users see a desktop background and for Linux, users see a login prompt.

myVM | Boot diagnostics

Virtual machine

Search

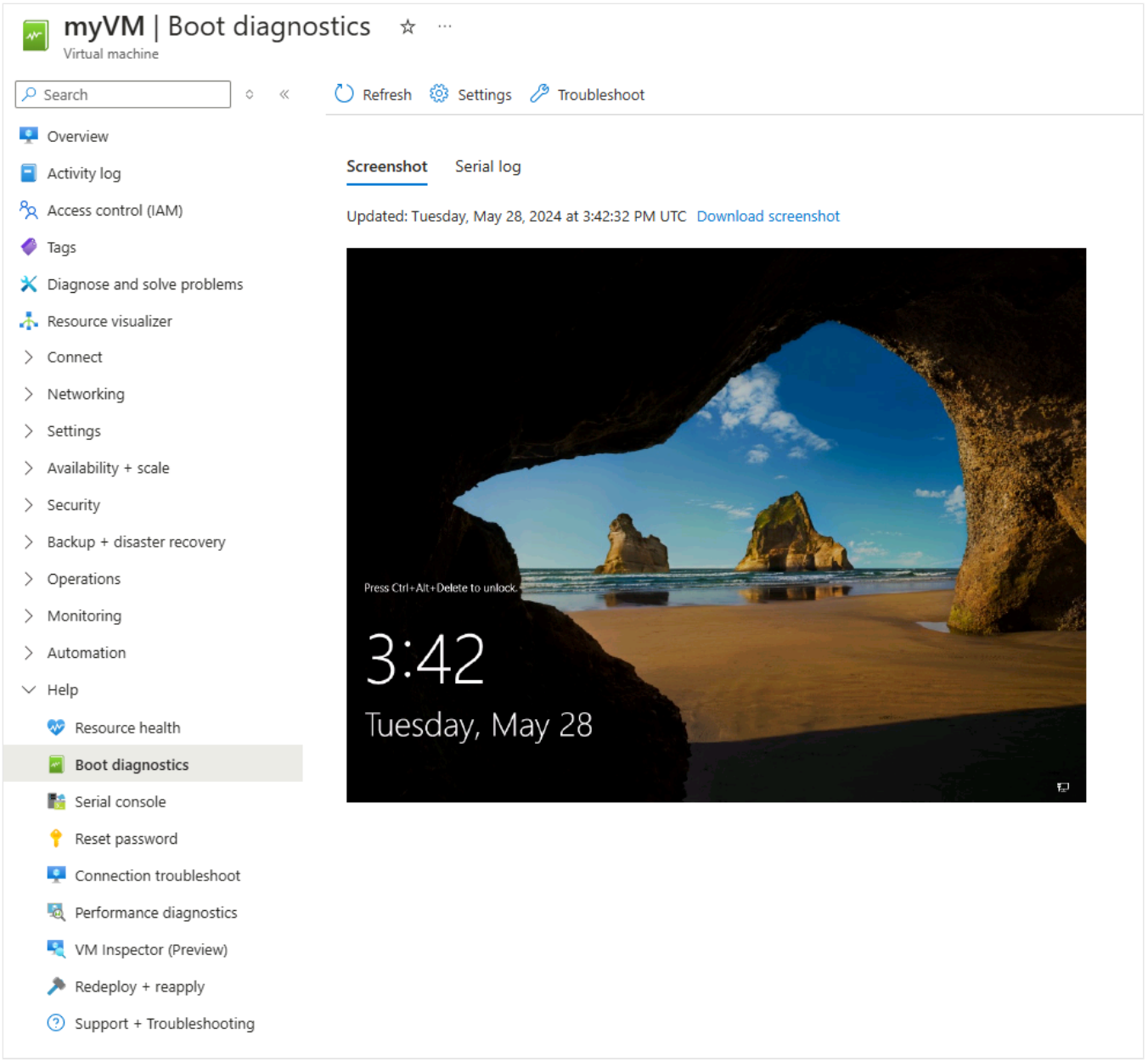
Refresh | Settings | Troubleshoot

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource visualizer
- Connect
- Networking
- Settings
- Availability + scale
- Security
- Backup + disaster recovery
- Operations
- Monitoring
- Automation
- Help
  - Resource health
  - Boot diagnostics
  - Serial console
  - Reset password
  - Connection troubleshoot
  - Performance diagnostics
  - VM Inspector (Preview)
  - Ubuntu Advantage support plan
  - Redeploy + reapply
  - Support + Troubleshooting

Screenshot | Serial log

Updated: Tuesday, May 28, 2024 at 3:38:28 PM UTC Download screenshot



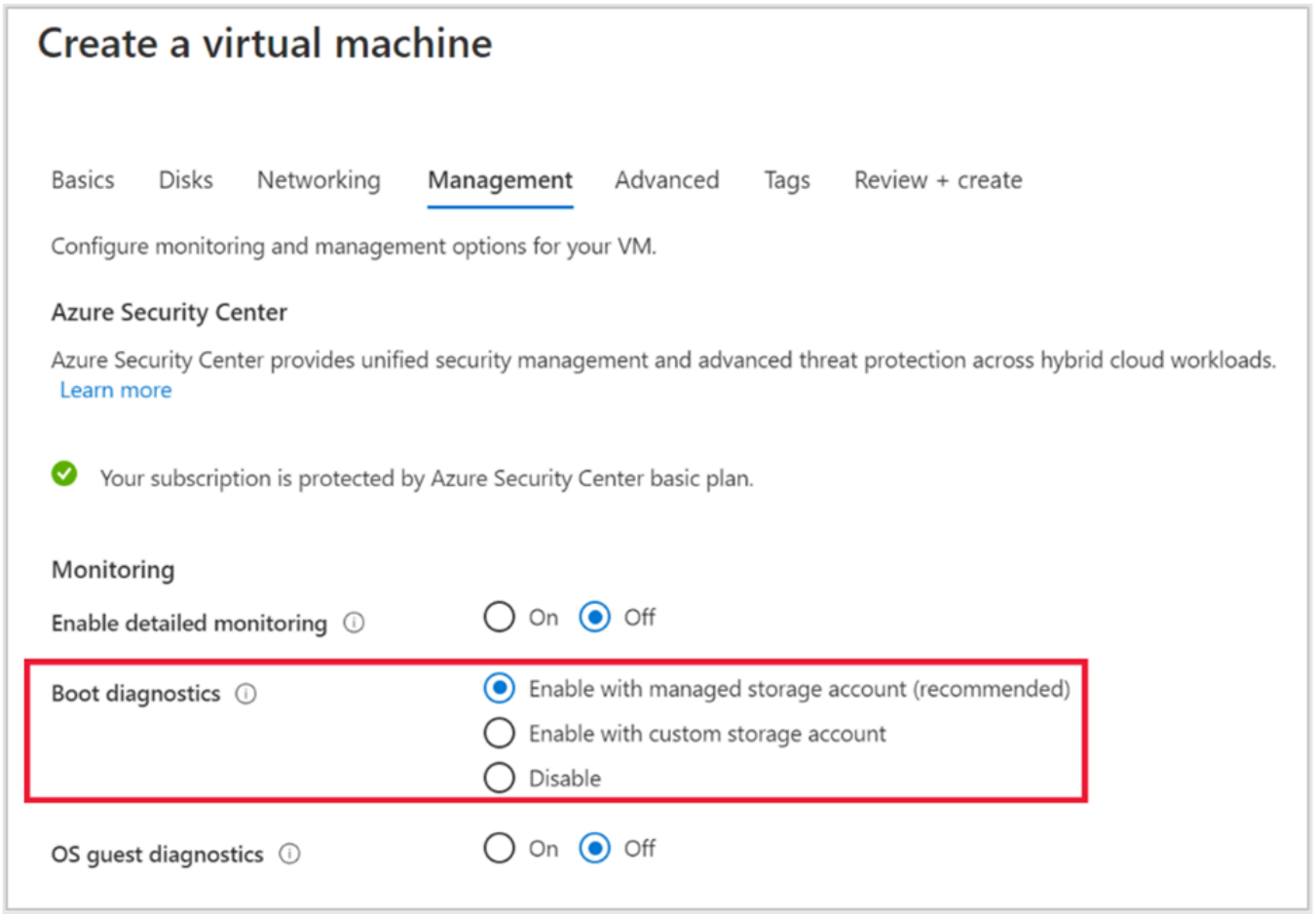


Enable managed boot diagnostics

Managed boot diagnostics can be enabled through the Azure portal, CLI and ARM Templates.

Enable managed boot diagnostics using the Azure portal

When you create a VM in the Azure portal, the default setting is to have boot diagnostics enabled using a managed storage account. Navigate to the *Management* tab during the VM creation to view it.



Enable managed boot diagnostics using CLI

Boot diagnostics with a managed storage account is supported in Azure CLI 2.12.0 and later. If you don't input a name or URI for a storage account, a managed account is used. For more information and code samples, see the CLI documentation for boot diagnostics.

Enable managed boot diagnostics using PowerShell

Boot diagnostics with a managed storage account is supported in Azure PowerShell 6.6.0 and later. If you don't input a name or URI for a storage account, a managed account is used. For more information and code samples, see the PowerShell documentation for boot diagnostics.

Enable managed boot diagnostics using Azure Resource Manager (ARM) templates

Everything after API version 2020-06-01 supports managed boot diagnostics. For more information, see boot diagnostics instance view.

ARM

```
            "name": "[parameters('virtualMachineName')]",
            "type": "Microsoft.Compute/virtualMachines",
            "apiVersion": "2020-06-01",
            "location": "[parameters('location')]",
            "dependsOn": [
                "[concat('Microsoft.Network/networkInterfaces/',
parameters('networkInterfaceName'))]"
            ],
            "properties": {
                "hardwareProfile": {
                    "vmSize": "[parameters('virtualMachineSize')]"
                },
                "storageProfile": {
                    "osDisk": {
                        "createOption": "fromImage",
                        "managedDisk": {
                            "storageAccountType": "[parameters('osDiskType')]"
                        }
                    },
                    "imageReference": {
                        "publisher": "publisherName",
                        "offer": "imageOffer",
                        "sku": "imageSKU",
                        "version": "imageVersion"
                    }
                },
                "networkProfile": {
                    "networkInterfaces": [
                        {
                            "id": "
[resourceId('Microsoft.Network/networkInterfaces',
parameters('networkInterfaceName'))]"
                        }
                    ]
                },
                "osProfile": {
                    "computerName": "[parameters('virtualMachineComputerName')]",
                    "adminUsername": "[parameters('adminUsername')]",
                    "linuxConfiguration": {
                        "disablePasswordAuthentication": true
                    }
                },
                "diagnosticsProfile": {
                    "bootDiagnostics": {
                        "enabled": true
                    }
                }
            }
        }
    ],
```

> ⓘ Note

Replace publisherName, imageOffer, imageSKU and imageVersion accordingly.

Limitations

- Managed boot diagnostics is only available for Azure Resource Manager VMs.
- Managed boot diagnostics doesn't support VMs using unmanaged OS disks.
- Boot diagnostics doesn't support premium storage accounts or zone redundant storage accounts. If either of these are used for boot diagnostics users receive an `StorageAccountTypeNotSupported` error when starting the VM.
- Managed storage accounts are supported in Resource Manager API version "2020-06-01" and later.
- Portal only supports the use of boot diagnostics with a managed storage account for single instance VMs.
- Users can't configure a retention period for Managed Boot Diagnostics. The logs are overwritten when the total size crosses 1 GB.
- If the OS disk is swapped after enabling boot diagnostics, the VM will required a stop/start before boot diagnostics will work again. To avoid this, disable boot diagnostics before swapping the OS disk and enable again after the OS disk swap has been completed.

Next steps

Learn more about the Azure Serial Console and how to use boot diagnostics to troubleshoot virtual machines in Azure.

Last updated on 12/01/2025

Essential machine management (preview)

Essential machine management simplifies the onboarding and configuration of management for Azure virtual machines (VMs) and arc-enabled servers. When you enable a subscription for essential machine management, all VMs and arc-enabled servers in that subscription are automatically enrolled and configured with a curated set of management features. This ensures that your machines are consistently configured for monitoring, security, and management.

Prerequisites

- Log Analytics workspace to collect log data collected from VMs.
- Azure Monitor workspace to collect metrics data collected from VMs.
- User assigned managed identity as described in Managed identity section below.

Required permissions

User

The user performing the enrollment must have the following roles in the subscription being enabled:

- Essential Machine Management Administrator
- Managed Identity Operator roles
- Resource Policy Contributor

If you're using a Log Analytics workspace or Azure Monitor workspace in a different subscription than the one being enabled for essential machine management:

- The user account must also have the **Essential Machine Management Administrator** role in the resource group of the Log analytics workspace or Azure Monitor workspace.
- The **Microsoft.ManagedOps** resource provider needs to be registered in the subscription of the Log analytics workspace or Azure Monitor workspace. Use the Azure PowerShell command: `Register-AzResourceProvider -ProviderNamespace "Microsoft.ManagedOps"`.

Managed identity

The enrollment requires a user assigned managed identity with **Contributor** permission for the subscription.

If you're using a Log Analytics workspace or Azure Monitor workspace in a different subscription than the one being enabled for essential machine management, then the managed identity must also have **Contributor** permissions in the resource group of the Log Analytics workspace or Azure Monitor workspace.

Features enabled

Essential machine management enables a standard set of features and allows you to optionally enable additional security features.

Essentials tier

The following features are part of the essentials tier.

Expand table

Feature	Description
Azure Monitor	Monitors and provides insights into VM performance and health. Configures metric-based recommended alerts.
Azure Update Manager	Automates the deployment of operating system updates to VMs.
Azure Machine Configuration	Audits the Azure security baseline policy
Azure Change Tracking and Inventory	Tracks changes to VM configurations and maintains an inventory of resources.

Essentials tier pricing

> **Note**
>
> During the initial phase of public preview, the Essential Machine Management features are provided at no extra charge. Logs generated from Change Tracking and Inventory incur a separate charge for both Azure Virtual Machines and Arc-enabled servers.

- For Azure Virtual Machines only, capabilities enabled by Essential Machine Management are provided at no extra charge.
- For Azure Arc-enabled servers with Windows Server Software Assurance, Windows Server PayGo, and Windows Server Extended Security Updates, capabilities enabled by Essential Machine Management are provided at no extra charge.

- For all other Arc-enabled servers, Essential Machine Management will be priced at $9 per server per month once billing is enabled at a future date. An announcement and documentation update will be posted when billing begins.

Security tier

The following security features are available as part of essential machine management. You can choose to enable any combination of these features for the enrolled VMs. Features in this section may incur an additional charge.

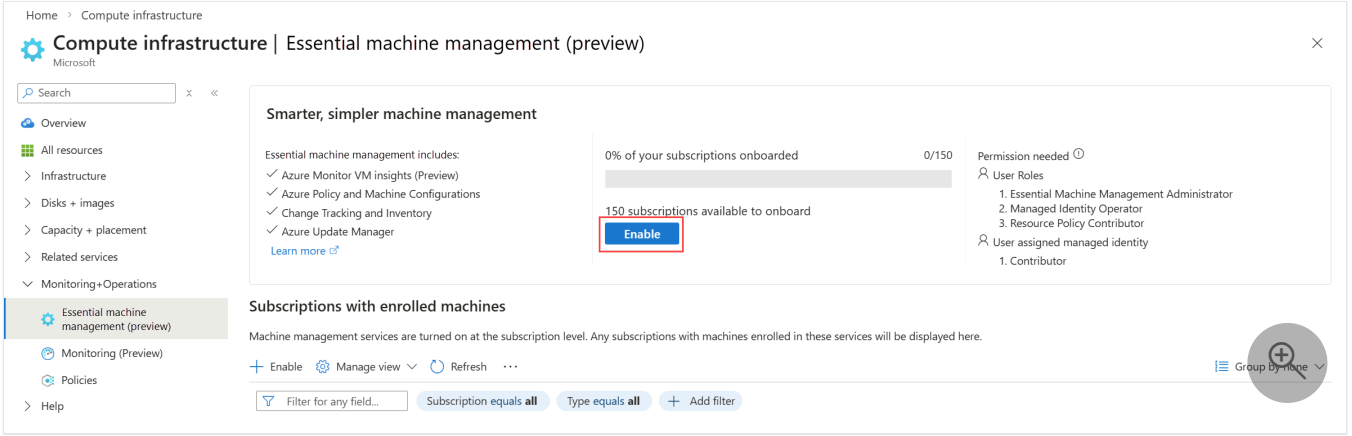
Expand table

Feature	Description	Cost
Foundational CSPM	Provides foundational cloud security posture management (CSPM) capabilities to assess and improve the security of your cloud resources.	No
Defender CSPM	Advanced cloud security posture management (CSPM) capabilities to enhance the security of your cloud resources.	Yes
Defender for cloud	Advanced threat protection and security management for VMs.	Yes

Enable a subscription

To enable machine management for a subscription, select **Essential machine management** from the **Configuration** menu, and click **Enable**.

> **Note**
>
> During public preview, the Azure portal is the only supported method for enabling machine management.



▼ Scope tab

The **Scope** tab includes the subscription that you want to enable and the managed identity.

Expand table

Setting	Description
Select a subscription	Click to select the subscription to enable. A list is provided with all subscriptions you have access to and the number of Azure VMs and Arc-enabled VMs in each.
Required user role assignments	Lists the required roles that your user account must be assigned to.
Current user role assignments	Lists the roles that are currently assigned to your user account.
User assigned managed identity	Select the managed identity to use for onboarding VMs in the subscription.
Required identity role assignment	Lists the required roles the managed identity must be assigned.
Current identity role assignment	Lists the roles currently assigned to the managed identity.

▼ Configure tab

The **Configure** tab includes the Log Analytics workspace and Azure Monitor workspace that will collect data from the managed VMs.

Expand table

Setting	Description
Log Analytics workspace	Select the Log Analytics workspace to use for collecting log data from VMs.
Azure Monitor workspace	Select the Azure Monitor workspace to use for collecting metrics data from VMs.

▼ Security tab

The **Security** tab allows you to select additional security services for the managed VMs.

Expand table

Setting	Description
Foundational CSPM	Continuously assess your cloud environment with agentless, risk-prioritized insights. Recommended for all workloads. This add-on incurs no additional charge.
Defender CSPM	Continuously assess your cloud environment with agentless, risk-prioritized insights. Recommended for all workloads. This add-on incurs an additional charge.
Defender for cloud	Comprehensive server protection with integrated endpoint detection and response (EDR), vulnerability management, file integrity monitoring, and advanced threat detection. Recommended for business-critical workloads. This add-on incurs an additional charge.

Existing VMs

Essential machine management is enabled for each subscription to automatically onboard all Azure VMs and arc-enabled servers in that subscription. Once enabled, any VMs added to the subscription are enrolled and configured with the selected features. The following behavior applies to existing VMs in the subscription when essential machine management is enabled.

- Existing services will retain their configuration. For example, if a VM is already using Update Management with a maintenance schedule, it will still follow that maintenance schedule.
- After the subscription is enabled, remediation tasks are created to enable the selected service for all existing VMs in the subscription.

> ⚠ **Warning**
>
> Use caution with the public preview if you have existing VMs with Change Tracking enabled. In this case, an additional Change Tracking DCR will be created and associated with the VM. Since Change Tracking supports only a single DCR though, either DCR could be assigned. If you would like to use the ManagedOps DCR, please remove the existing DCR.

Excluding VMs

There is currently no ability to exclude VMs in the enabled subscription. All VMs in the subscription are onboarded and configured with the selected features.

Disable a subscription

Disable a subscription by selecting it and then clicking **Offboard**. When you disable a subscription, any VMs added to that subscription are no longer configured with the selected management features. The configuration isn't changed for existing VMs though. They will continue to be managed with the existing features until you manually remove them.

> ⚠ **Warning**
>
> When you disable a subscription, machines in that subscription no longer use consolidated pricing. Pricing for these machines will revert to standard pricing for each individual service, which will most likely increase your costs. Ensure that you disable any unneeded services on existing VMs to avoid additional charges.

Troubleshooting

See Troubleshoot essential machine management (preview) for help resolving common issues with essential machine management. This article also identifies the objects created during enrollment and how to verify their creation.

Detailed configuration

The following table describes the specific configuration applied to each VM when essential machine management is enabled.

Expand table

Feature	Configuration details
Azure Monitor	- Installs Azure Monitor agent - Collects standard set of performance counters. - Configures metric-based recommended alerts
Azure Update Manager	- Installs extension (`Microsoft.CPlat.Core.LinuxPatchExtension` or `Microsoft.CPlat.Core.WindowsPatchExtension`) - Periodic assessment enabled.

Feature	Configuration details
Azure Machine Configuration	- Installs extension (`Microsoft.GuestConfiguration.ConfigurationforLinux` or `Microsoft.GuestConfiguration.ConfigurationforWindows`) - Applies the Linux security baseline and Windows security baseline in **Audit only** mode.
Azure Change Tracking and Inventory	- Installs extension (`Microsoft.Azure.ChangeTrackingAndInventory. ChangeTracking-Windows` or `Microsoft.Azure.ChangeTrackingAndInventory.ChangeTracking-Linux`) - Uses Log Analytics workspace specified in onboarding. - Collects basic files and registry keys.
Defender CSPM	- All settings on by default.

Next steps

- Troubleshoot any issues with essential machine management.

Last updated on 04/06/2026

Troubleshoot essential machine management (preview)

Summarize this article for me

This article provides troubleshooting steps for issues that may occur when enabling essential machine management. If you receive an error during enrollment, then there should be a specific resolution. If you don't get any errors during enrollment, but the machines in the subscription are not being onboarded to the selected services, then use the sections below to validate the different steps of the enrollment process to identify where any issues may have occurred.

Errors during enrollment

Following are common errors that may occur when enabling essential machine management for a subscription.

Could not validate resource existence.
The error message will include the resource ID of the Log Analytics workspace or Azure Monitor workspace that you selected during enrollment.

1. Check whether you have the **Essential Machine Management Administrator** role in the resource group of the Log analytics workspace or Azure Monitor workspace.
2. If the workspaces are in a different subscription than the one being enabled for essential machine management, verify that you have `Microsoft.ManagedOps` resource provider registered in the subscription.

Change Log Analytics workspace or Azure Monitor workspace
If you've already configured essential machine management and then enable it again using a different Log Analytics workspace or Azure Monitor workspace, you'll get an error saying that the workspace can't be changed once it's set.

To change either of the workspaces, you must first disable the subscription and then re-enable it with the new workspaces. All machines in the subscription will be re-enrolled and configured with the new workspaces, but any data already collected in the old workspace will be retained.

Disable Defender for cloud
You'll receive and error if you attempt to disable Defender for cloud for subscription that was already enabled for essential machine management. You must disable the subscription from the Defender for cloud portal.

Verify objects created

If you didn't see any errors during enrollment, start by verifying that the objects in the following table are created in the resource group for the Log Analytics workspace and Azure Monitor workspace. These are the data collection rules (DCRs) and solutions that enable change tracking and data collection for Azure Monitor.

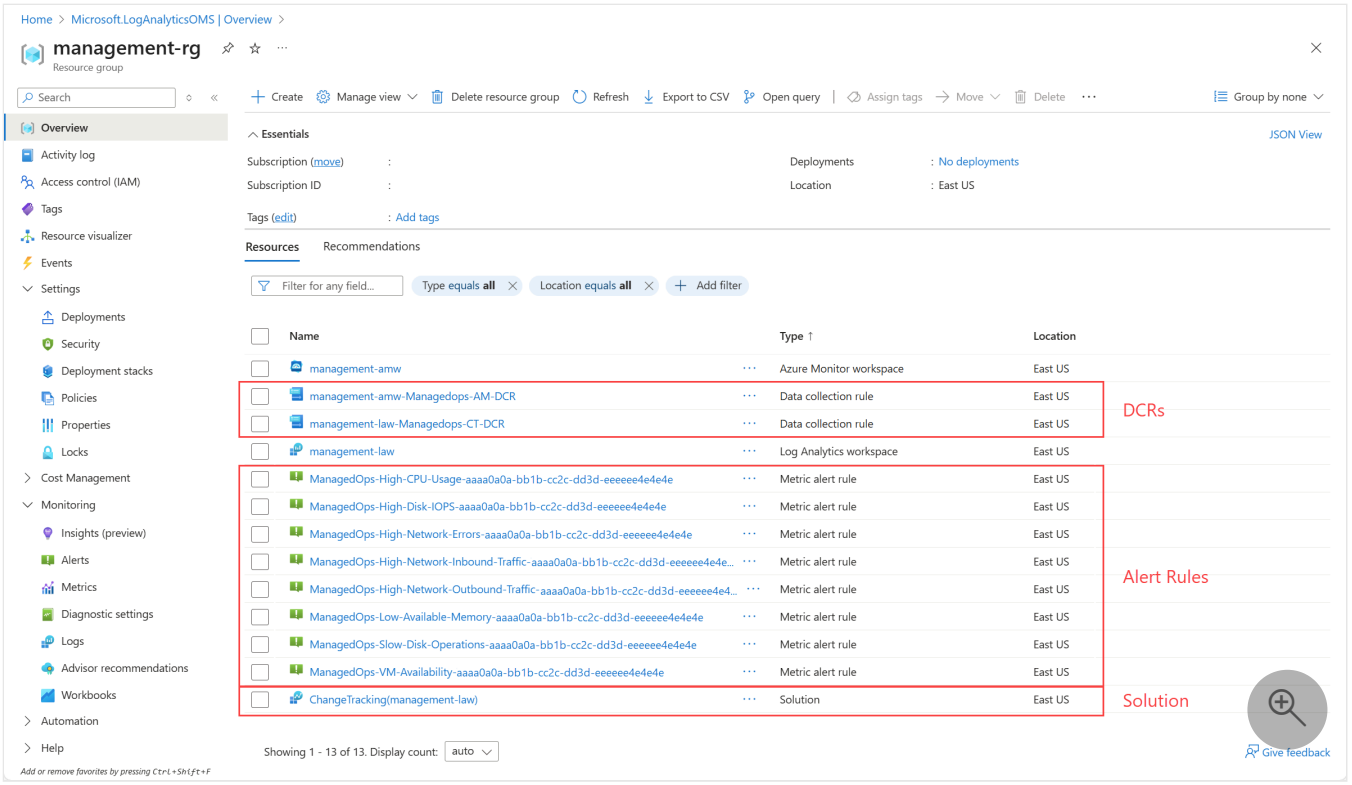
Expand table

Type	Name	Description
DCR	`<workspace id>-Managedops-AM-DCR` [1]	OpenTelemetry metrics from VM guests
DCR	`<workspace id>-Managedops-CT-DCR` [1]	Change tracking and inventory. Collects files, registry keys, softwares, Windows services, Linux daemons
Solution	`ChangeTracking(workspace name)`	Solution added to Log Analytics workspace to support Change tracking and inventory.

[1] The `<workspace id>` in the DCR name may be truncated.

Verify that the alert rules were created by checking for the following rules in the resource group for the Azure Monitor workspace.

- `ManagedOps-High-CPU-Usage-Alert`
- `ManagedOps-High-Disk-IOPS-Alert`
- `ManagedOps-High-Network-Errors-Alert`
- `ManagedOps-High-Network-Inbound-Traffic-Alert`
- `ManagedOps-High-Network-Outbound-Traffic-Alert`
- `ManagedOps-Low-Available-Memory-Alert`
- `ManagedOps-Slow-Disk-Operations-Alert`
- `ManagedOps-VM-Availability-Alert`

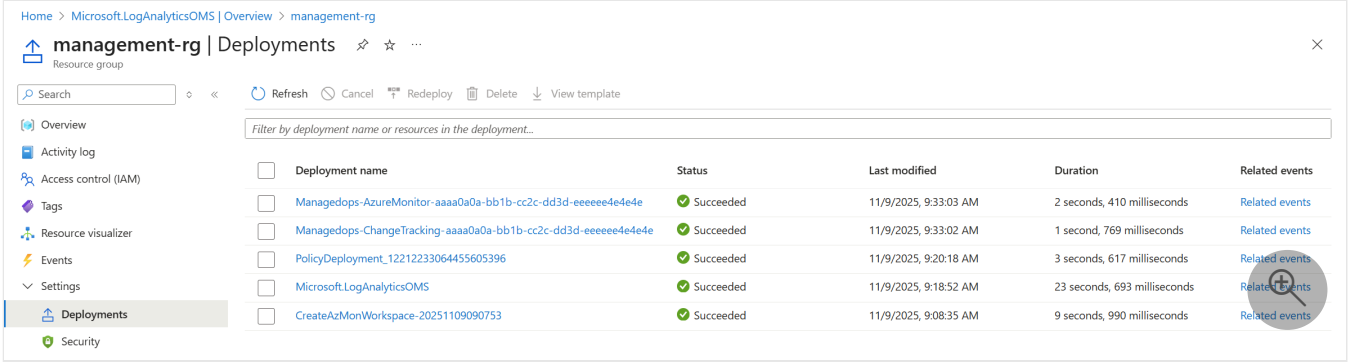


Check deployments for errors

If you don't see any of these objects within a few minutes of enabling the subscription, check for any errors in the deployments that are responsible for creating them. Open **Deployments** in the resource group and search for deployments with `Managedops` in the name. Click **Related events** to view the Activity log entries for the deployment. Alternatively, you can check the Activity log directly for the resource group and search for `Managedops` to identify any activity related to machine enablement.

The deployment names will look similar to the following:

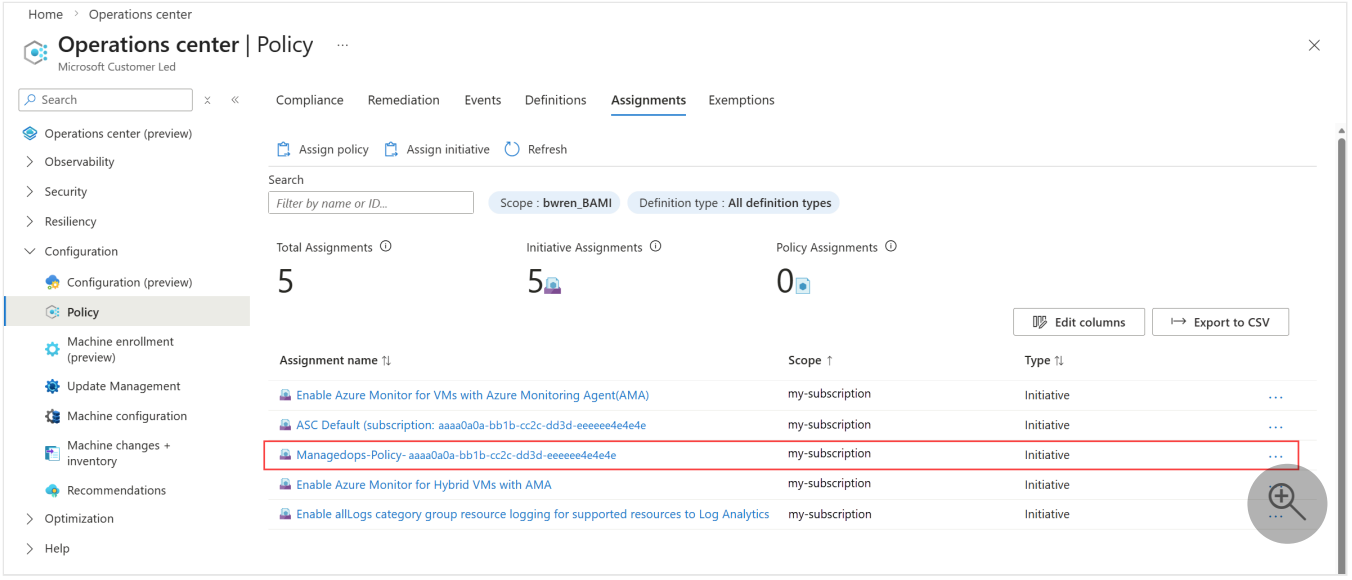
- `Managedops-ChangeTracking-{Subscription Id}`
- `Managedops-AzureMonitor-{Subscription Id}`



Verify policy assignments

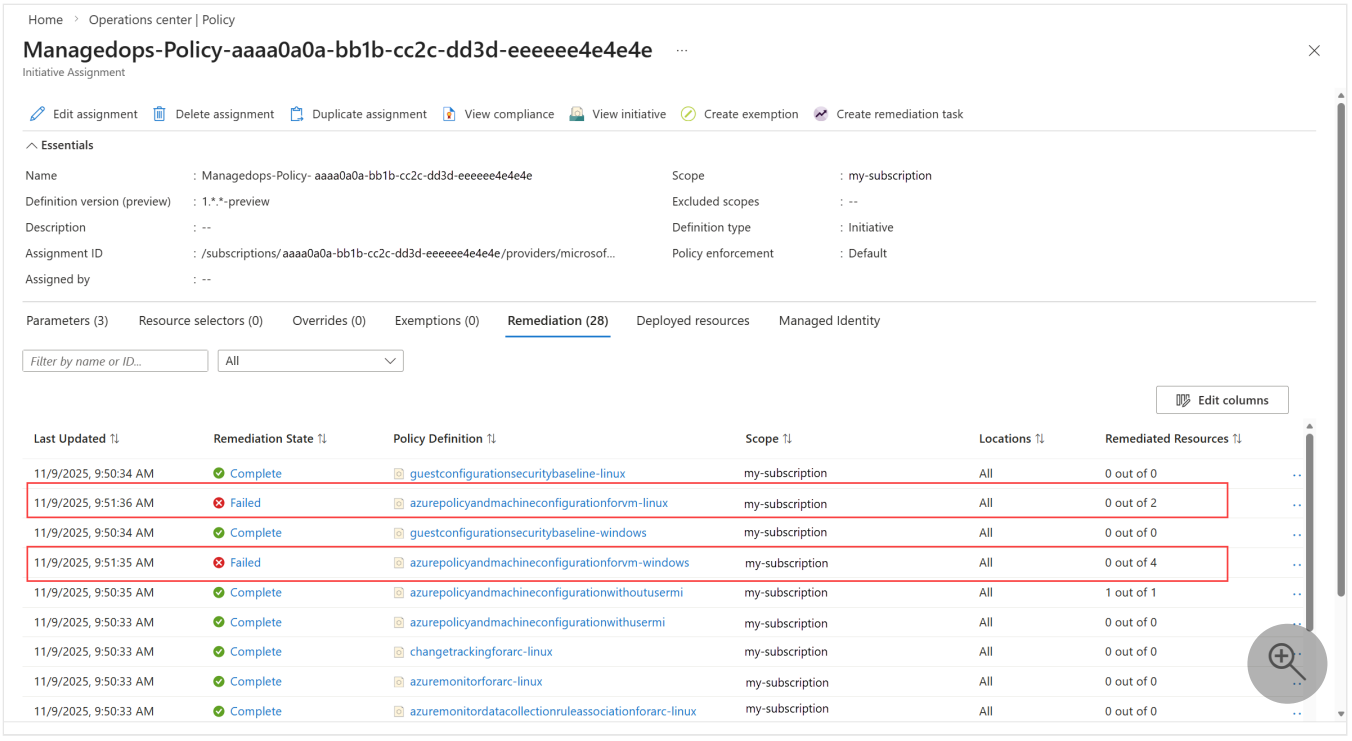
If the required objects have been created, and there are no errors in the deployments, verify that the policy assignment exists in the subscription. The assignment is responsible for applying the required configurations to the VMs in the subscription.

Open the **Policy** page in the Azure portal and select **Assignments**. Search for `ManagedOpsPolicy`. If you don't see the policy assignment, then you may not have enough permission to make a policy assignment in that subscription. Verify permissions at Required permissions.

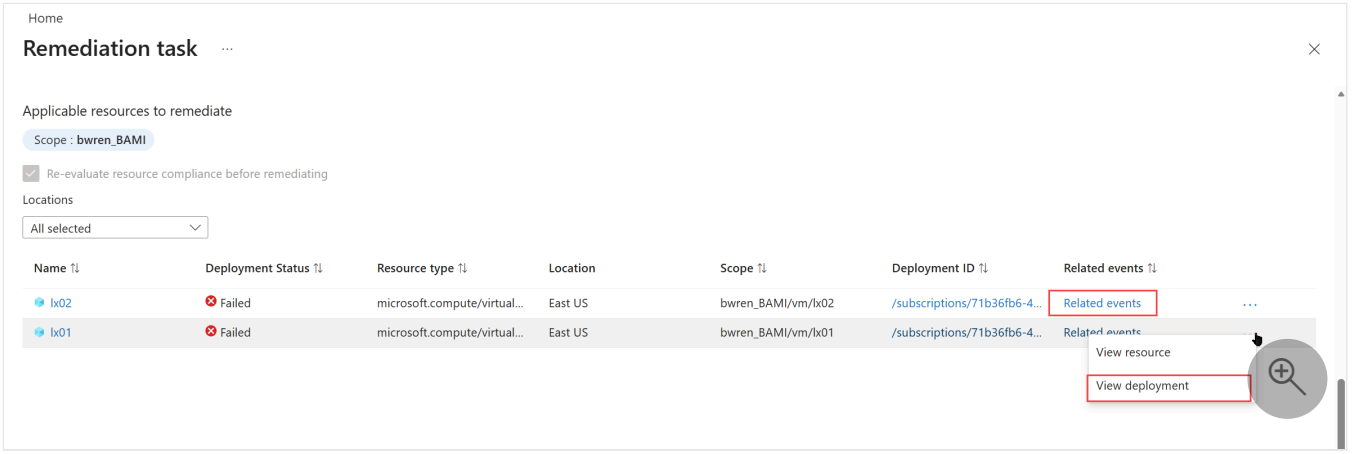


Check remediation tasks

If the assignments look correct, check the remediation tasks which are created to enable the selected features on all existing VMs in the subscription. Click on the assignment and select **Remediation**. If any of the **Remediated Resources** are not showing that all resources are remediated, or if the **Remediation State** is **Failed**, then click on the **Complete** or **Failed** status to open the details for that remediation task.



In the details of the remediation task, click on **Related Events** or **View Deployment** to locate a detailed error message.



Contact Microsoft

If you've completed all of the previous steps and are still having issues with essential machine management, contact Microsoft at machineenrollmentsupport@microsoft.com.

Last updated on 03/11/2026

Backup and restore options for virtual machines in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

You can protect your data by taking backups at regular intervals. There are several backup options available for virtual machines (VMs), depending on your use-case.

Azure Backup

You'll use Azure Backup for most use-cases involving backup operations on Azure VMs running production workloads. Azure Backup supports application-consistent backups for both Windows and Linux VMs. Azure Backup creates recovery points that are stored in geo-redundant recovery vaults. When you restore from a recovery point, you can restore entire VM or specific files.

For a simple, hands-on introduction to Azure Backup for Azure VMs, see the Azure Backup quickstart.

For more information on how Azure Backup works, see Plan your VM backup infrastructure in Azure

Azure Site Recovery

Azure Site Recovery protects your VMs from a major disaster scenario. These scenarios may include widespread service interruptions or regional outages caused by natural disasters. You can configure Azure Site Recovery for your VMs so that your applications are recoverable in matter of minutes with a single click. You can replicate to an Azure region of your choice, since recovery isn't restricted to paired regions.

You can run disaster-recovery drills with on-demand test failovers, without affecting your production workloads or ongoing replication. Create recovery plans to orchestrate failover and failback of the entire application running on multiple VMs. The recovery plan feature is integrated with Azure Automation runbooks.

You can get started by replicating your virtual machines.

Managed snapshots

In development and test environments, snapshots provide a quick and simple option for backing up VMs that use managed disks. A managed snapshot is a full, read-only copy of a managed disk. Snapshots exist independently of their source disks.

Snapshots can be used to create new managed disks when a VM is rebuilt. They're billed based on the used portion of the disk. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GB and actual used data size of 10 GB, snapshot will be billed only for the used data size of 10 GB.

For more information on creating snapshots, see:

- Create copy of VHD stored as a Managed Disk

Virtual machine restore points

At this time, you can use Azure REST APIs to back up and restore your VMs. This approach is most often used by independent software vendor (ISVs) or organizations with a relatively small number of VMs to manage.

You can use the API to create a VM restore point collection. The restore point collection itself contains individual restore points for specific VMs. Each restore point stores a VM's configuration and a snapshot for each attached managed disk. To save space and costs, you can exclude any disk from your VM restore points.

Once created, VM restore points can then be used to restore individual disks. To restore a VM, restore all relevant disks and attach them to a new VM.

Learn more about working with VM restore points and the restore point collections API.

Next steps

You can try out Azure Backup by following the Azure Backup quickstart.

You can also plan and implement reliability for your virtual machine configuration. For more information see Virtual Machine Reliability.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Reliability in Azure Virtual Machines

Azure Virtual Machines provides on-demand, scalable compute resources. As a foundational infrastructure service, it's designed to deliver enterprise-grade reliability and availability for mission-critical workloads.

When you use Azure, reliability is a shared responsibility. Microsoft provides a range of capabilities to support resiliency and recovery. You're responsible for understanding how those capabilities work within all of the services you use, and selecting the capabilities you need to meet your business objectives and uptime goals.

This article describes how to make Virtual Machines resilient to a variety of potential outages and problems, including transient faults, availability zone outages, region outages, and service maintenance. It also describes how you can use backups to recover from other types of problems, and highlights some key information about the Virtual Machines service level agreement (SLA).

> **Important**
>
> When you consider the reliability of a virtual machine (VM), you also need to consider the reliability of your disks, network infrastructure, and applications that run on your VMs. Improving the resiliency of the VM alone might have limited impact if the other components aren't equally resilient. Depending on your resiliency requirements, you might need to make configuration changes across multiple areas.

Production deployment recommendations

For more information about how to deploy VMs to support your solution's reliability requirements and how reliability affects other aspects of your architecture, see Architecture best practices for Virtual Machines and scale sets in the Azure Well-Architected Framework.

Reliability architecture overview

VMs are the fundamental compute unit in Azure, whether you provision the VMs yourself or use other Azure compute services that transparently provision and manage them for you.

An individual VM is also known as a *single instance VM*. It runs on a specific host, which is a physical server. Most VMs share their host with other VMs.

When you create your VMs, you can influence where they run within the underlying infrastructure. Typically, you make placement decisions based on your requirements for

reliability, latency, and isolation. Azure provides several configuration options that affect how your VMs are placed.

- **Region:** You can select which Azure region that your VM should run in. A region is a geographic area that might contain multiple datacenters, each with a large number of hosts.

- **Availability zone:** Availability zones are physically separate groups of datacenters within each Azure region. In regions that support availability zones, you can select which zone the VM runs in. For more information, see Resilience to availability zone failures.

- **Availability sets:** An availability set is a logical grouping of VMs that allows Azure to understand how your application is built to provide for redundancy and availability.

  When you use availability sets, Azure distributes a group of VMs across different fault domains. This distribution minimizes the risk of localized hardware failures by grouping VMs that share a common power source and network switch.

  Availability sets can also place different VMs in different update domains, which controls how the Azure platform rolls out platform updates. By using update domains, you can ensure that only a subset of your VMs is restarted for updates at one time.

- **Proximity placement groups:** For workloads that need to achieve the lowest possible latency between VMs, you can use a proximity placement group to ensure that Azure places the VMs physically close to each other. However, proximity placement means that an outage of the datacenter can affect all of the VMs in the group. To achieve high reliability, you might need to provision multiple proximity placement groups in different availability zones.

- **Dedicated hosts:** You can use Azure Dedicated Host to provision your own physical server that runs one or more VMs, such as for strict compliance requirements. However, when you provision a dedicated host, an outage in its datacenter can affect all of the VMs on that host. To achieve high reliability, you might need to provision multiple dedicated hosts in different availability zones.

If you create a set of VMs that perform similar functions, consider using Azure Virtual Machine Scale Sets to create and manage the VMs as a group. Scale sets also provide more reliability options, such as spreading the VMs across multiple availability zones.

For more information about availability for VMs, see Availability options for Virtual Machines.

Resilience to transient faults

Transient faults are short, intermittent failures in components. They occur frequently in a distributed environment like the cloud, and they're a normal part of operations. Transient faults correct themselves after a short period of time. It's important that your applications can handle transient faults, usually by retrying affected requests.

All cloud-hosted applications should follow the Azure transient fault handling guidance when they communicate with any cloud-hosted APIs, databases, and other components. For more information, see Recommendations for handling transient faults.

Applications that run on your VMs should implement appropriate fault-handling strategies to ensure that any temporary interruptions in service don't affect your workload.

Resilience to availability zone failures

Availability zones are physically separate groups of datacenters within an Azure region. When one zone fails, services can fail over to one of the remaining zones.

An individual VM can be deployed in a *zonal* configuration, which means that it's pinned to a single availability zone that you select. By itself, a zonal VM isn't resilient to zone outages. However, you can create multiple VMs and place them in different availability zones, then spread your applications and data across the VM instances. Alternatively, you can use virtual machine scale sets to deploy a set of VMs across multiple availability zones.

If you don't configure a VM to be zonal, it's considered *nonzonal* or *regional*. Nonzonal VMs might be placed in any availability zone within the region. If any availability zone in the region experiences an outage, nonzonal VMs might be in the affected zone and can experience downtime.

Region support

Zonal VMs can be deployed into any region that supports availability zones.

However, some VM types and sizes are only available in specific regions, or specific zones within a region. To check which regions and zones support the VM types that you need, use the following resources:

- To check the VM types available in each region, see Products available by region.
- To check the supported VM types and sizes within each zone of a specific region, see Check VM SKU availability.

Cost

There's no cost difference between a zonal and nonzonal VM.

Configure availability zone support

This section explains how to configure availability zone support for your VM instance.

> **Note**
>
> When you select which availability zones to use, you're actually selecting the *logical availability zone*. If you deploy other workload components in a different Azure subscription, they might use a different logical availability zone number to access the same physical availability zone. For more information, see **Physical and logical availability zones**.

- **Create a zonal VM.** You can create a zonal VM by using the following guides:
  - Create VMs in an availability zone by using the Azure portal
  - Create a VM in an availability zone by using the Azure CLI
  - Create a VM in an availability zone by using Azure PowerShell
- **Convert existing VMs to a zonal configuration.** You can move from a nonzonal VM to a zonal VM. This process creates a new VM in the target availability zone and requires that the VM be stopped during the move process. For more information, see Move Azure single instance VMs from regional to zonal target availability zones.
- **Change the availability zone of an existing zonal VM.** Zonal VMs can't be moved to a different availability zone. You need to deploy a new zonal VM in the desired availability zone instead.
- **Convert a zonal VM to a nonzonal configuration.** Zonal VMs can't be converted to a nonzonal configuration. You need to deploy a new nonzonal VM instead.

Behavior when all zones are healthy

This section describes what to expect when VM instances are configured with availability zone support and all availability zones are operational.

- **Traffic routing between zones:** You're responsible for routing traffic between VMs, including VMs that are in different availability zones. Common approaches include Azure Load Balancer and Azure Application Gateway. For more information, see Load balancing options.

- **Data replication between zones:** You're responsible for any data replication that needs to happen between VMs, including across VMs in different availability zones. Databases and other similar stateful applications that run on VMs often provide capabilities to replicate data.

Behavior during a zone failure

This section describes what to expect when VM instances are configured with availability zone support and there's an outage in their availability zones.

- **Detection and response:** You're responsible for detecting and responding to zone failures that affect your VMs.
- **Notification**: Microsoft doesn't automatically notify you when a zone is down. However, you can use Azure Resource Health to monitor for the health of an individual resource, and you can set up Resource Health alerts to notify you of problems. You can also use Azure Service Health to understand the overall health of the service, including any zone failures, and you can set up Service Health alerts to notify you of problems.
- **Active requests:** Any active requests or other work that occurs on the VM during the zone failure are likely to be terminated.
- **Expected data loss:** Zonal VM disks might be unavailable during a zone failure.

  If you use zone-redundant storage (ZRS) disks and an outage affects your VM, you can force detach your ZRS disks from the failed VM. This approach allows you to attach the ZRS disks to another VM.
- **Expected downtime:** VMs remain down until the availability zone recovers.
- **Traffic rerouting:** You're responsible for rerouting traffic to other VMs in healthy zones.

  If you configure a zone-resilient load balancer and it performs health checks, the load balancer typically detects failed VMs and can route traffic to other VM instances in healthy zones.

Zone recovery

When the zone is healthy, VMs in the zone restart. You're responsible for any zone recovery procedures and data synchronization that your workloads require.

Test for zone failures

You can use Azure Chaos Studio to simulate the loss of a VM as part of an experiment. Chaos Studio provides built-in faults for VMs, including the ability to shut down a VM. You can use these capabilities to simulate zone-level failures and test your failover processes.

Custom multi-zone solutions for resiliency

When you deploy multiple VMs into different zones, you're responsible for configuring and managing replication, load balancing, failover, and failback processes.

Some applications provide built-in capabilities that can help when you deploy across multiple VMs. For example, SQL Server on Azure VMs provides a set of capabilities to simplify your configuration and management processes across availability zones.

You can consider using Azure Site Recovery zone-to-zone disaster recovery (DR) when your application runs in a single zone at a time and you don't require near-instant failover between zones. Zone-to-zone DR has some important limitations, so review your requirements thoroughly.

Resilience to region-wide failures

VMs are single-region resources. If the region becomes unavailable, your VM is also unavailable.

Custom multi-region solutions for resiliency

You can deploy multiple VMs into different regions, but you need to implement replication, load balancing, and failover processes.

Site Recovery is a service that enables DR by replicating VMs and their data to a secondary region. You can select almost any Azure region as your secondary region, including nonpaired region combinations. For more information, see Azure to Azure DR architecture.

Some applications create clusters or other constructs to replicate data and distribute work across multiple VMs, including in different regions. These applications can simplify the configuration of a multi-region solution.

For an example architecture that illustrates using VMs across multiple regions, see Multi-region load balancing with Azure Traffic Manager, Azure Firewall, and Application Gateway.

Resilience to service maintenance

Azure performs regular periodic maintenance on VMs to ensure reliability. There are multiple ways that you can ensure your workloads remain operational during maintenance activities:

- When you use availability sets or virtual machine scale sets, you can configure update domains. Update domains help distribute maintenance activities across different VMs at different times, so your VMs don't all restart simultaneously.
- You can customize the timing that maintenance is applied to your VMs by using *maintenance control*. You can use maintenance configurations to schedule it at a time that suits your workload.
- You can receive notifications of upcoming maintenance activities.

For more information, see Guest updates and host maintenance overview.

Backup and restore

Azure Backup provides native backup support for VMs. Azure Backup creates and manages backups, and provides application-consistent protection for the entire VM, including all attached disks. A VM backup solution with Azure Backup is ideal when you need coordinated backup of multiple disks or application-aware backups. However, for database workloads, consider application-specific backup solutions that provide transaction-consistent protection and faster recovery options.

With Azure Backup for VMs, you can customize the backup frequency, retention duration, and storage configuration to suit your needs. For more information, see Azure Backup for VMs.

Backup also supports disks that are attached to VMs. For more information, see Overview of Azure Disk Backup.

For most solutions, you shouldn't rely exclusively on backups. Instead, use the other capabilities described in this guide to support your resiliency requirements. However, backups protect against some risks that other approaches don't. For more information, see What are redundancy, replication, and backup?.

Service-level agreement

The service-level agreement (SLA) for Azure services describes the expected availability of each service and the conditions that your solution must meet to achieve that availability expectation. For more information, see SLAs for online services.

For Virtual Machines, the SLA provides a base level of availability. The uptime percentage defined in the SLA increases when you have two or more VMs and you take the following

actions:

- Configure those VMs to be deployed across two or more availability zones.
- Configure those VMs to be deployed into an availability set.

Next steps

- VMs in Azure
- Virtual Machine Scale Sets

Related resource

- Reliability in Azure

Last updated on 01/23/2026

Virtual machines in Azure

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

Azure virtual machines (VMs) are one of several types of on-demand, scalable computing resources that Azure offers. Typically, you choose a virtual machine when you need more control over the computing environment than the other choices offer. This article gives you information about what you should consider before you create a virtual machine, how you create it, and how you manage it.

An Azure virtual machine gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it. However, you still need to maintain the virtual machine by performing tasks, such as configuring, patching, and installing the software that runs on it.

Azure virtual machines can be used in various ways. Some examples are:

- **Development and test** – Azure virtual machines offer a quick and easy way to create a computer with specific configurations required to code and test an application.
- **Applications in the cloud** – Because demand for your application can fluctuate, it might make economic sense to run it on a virtual machine in Azure. You pay for extra virtual machines when you need them and shut them down when you don't.
- **Extended datacenter** – virtual machines in an Azure virtual network can easily be connected to your organization's network.

What do I need to think about before creating a virtual machine?

There's always a multitude of design considerations when you build out an application infrastructure in Azure. These aspects of a virtual machine are important to think about before you start:

- The names of your resources
- The location where the resources are stored
- The size of the virtual machine
- The maximum number of virtual machines that can be created
- The operating system that the virtual machine runs

- The configuration of the virtual machine after it starts
- The related resources that the virtual machine needs

> **Note**
>
> Trusted Launch as default (TLaD) is available in preview for new Generation 2 Virtual machines (VMs). With TLaD, any new Generation 2 VMs created through any client tools defaults to Trusted Launch with secure boot and vTPM enabled. Register for the TLaD preview to validate the default changes in your respective environment and prepare for the upcoming change.

Parts of a VM and how they're billed

When you create a virtual machine, you're also creating resources that support the virtual machine. These resources come with their own costs that should be considered.

The default resources supporting a virtual machine and how they're billed are detailed in the following table:

Expand table

Resource	Description	Cost
Virtual network	For giving your virtual machine the ability to communicate with other resources	Virtual Network pricing
A virtual Network Interface Card (NIC)	For connecting to the virtual network	There's no separate cost for NICs. However, there's a limit to how many NICs you can use based on your VM's size. Size your VM accordingly and reference Virtual Machine pricing.
A private IP address and sometimes a public IP address.	For communication and data exchange on your network and with external networks	IP Addresses pricing
Network security group (NSG)	For managing the network traffic to and from your VM. For example, you might need to open port 22 for SSH access, but you might want to block traffic to port 80. Blocking and allowing port access is done through the NSG.	There are no additional charges for network security groups in Azure.

Resource	Description	Cost
OS Disk and possibly separate disks for data.	It's a best practice to keep your data on a separate disk from your operating system, in case you ever have a VM fail, you can detach the data disk, and attach it to a new VM.	All new virtual machines have an operating system disk and a local disk. Azure doesn't charge for local disk storage. The operating system disk, which is usually 127GiB but is smaller for some images, is charged at the regular rate for disks. You can see the cost for attach Premium (SSD based) and Standard (HDD) based disks to your virtual machines on the managed disks pricing page.
In some cases, a license for the OS	For providing your virtual machine runs to run the OS	The cost varies based on the number of cores on your VM, so size your VM accordingly. The cost can be reduced through the Azure Hybrid Benefit.

You can also choose to have Azure can create and store public and private SSH keys - Azure uses the public key in your VM and you use the private key when you access the VM over SSH. Otherwise, you need a username and password.

By default, these resources are created in the same resource group as the VM.

Locations

There are multiple geographical regions around the world where you can create Azure resources. Usually, the region is called **location** when you create a virtual machine. For a virtual machine, the location specifies where the virtual hard disks are stored.

This table shows some of the ways you can get a list of available locations.

Expand table

Method	Description
Azure portal	Select a location from the list when you create a virtual machine.
Azure PowerShell	Use the Get-AzLocation command.
REST API	Use the List locations operation.

Method	Description
Azure CLI	Use the az account list-locations operation.

Availability

There are multiple options to manage the availability of your virtual machines in Azure.

- **Availability Zones** are physically separated zones within an Azure region. Availability zones guarantee virtual machine connectivity to at least one instance at least 99.99% of the time when you have two or more instances deployed across two or more Availability Zones in the same Azure region.
- **Virtual Machine Scale Sets** let you create and manage a group of load balanced virtual machines. The number of virtual machine instances can automatically increase or decrease in response to demand or a defined schedule. Scale sets provide high availability to your applications, and allow you to centrally manage, configure, and update many virtual machines. Virtual machines in a scale set can also be deployed into multiple availability zones, a single availability zone, or regionally.

Fore more information see Availability options for Azure virtual machines and SLA for Azure virtual machines.

Sizes and pricing

The size of the virtual machine that you use is determined by the workload that you want to run. The size that you choose then determines factors such as processing power, memory, storage capacity, and network bandwidth. Azure offers a wide variety of sizes to support many types of uses.

Azure charges an hourly price based on the virtual machine's size and operating system. For partial hours, Azure charges only for the minutes used. Storage is priced and charged separately.

Scaling to multiple VMs

The number of virtual machines that your application uses can scale up and out to whatever is required to meet your needs. Azure offers several automatic scaling systems and instance types based on the size of your workload. For more information on scaling, compare VM-based compute products.

Virtual machine total core limits

Your subscription has default quota limits in place that could impact the deployment of many virtual machines for your project. The current limit on a per subscription basis is 20 virtual machine total cores per region. Limits can be raised by filing a support ticket requesting an increase Quota and capacity are checked separately during deployment. Even with approved quota, deployment can still fail if there isn't enough capacity in the target region or availability zone.

Managed disks

Managed disks handles Azure Storage account creation and management in the background for you, and ensures that you don't have to worry about the scalability limits of the storage account. You specify the disk size and the performance tier (Standard or Premium), and Azure creates and manages the disk. As you add disks or scale the virtual machine up and down, you don't have to worry about the storage being used. If you're creating new virtual machines, use the Azure CLI or the Azure portal to create virtual machines with Managed OS and data disks. If you have virtual machines with unmanaged disks, you can convert your virtual machines to be backed with managed disks.

You can also manage your custom images in one storage account per Azure region, and use them to create hundreds of virtual machines in the same subscription. For more information about managed disks, see the managed disks Overview.

Distributions

Microsoft Azure supports various Linux and Windows distributions. You can find available distributions in the marketplace, Azure portal or by querying results using CLI, PowerShell, and REST APIs.

This table shows some ways that you can find the information for an image.

Expand table

Method	Description
Azure portal	The values are automatically specified for you when you select an image to use.
Azure PowerShell	Get-AzVMImagePublisher -Location *location* Get-AzVMImageOffer -Location *location* -Publisher *publisherName*

Method	Description
	Get-AzVMImageSku -Location *location* -Publisher *publisherName* -Offer *offerName*
REST APIs	List image publishers List image offers List image skus
Azure CLI	az vm image list-publishers --location *location* az vm image list-offers --location *location* --publisher *publisherName* az vm image list-skus --location *location* --publisher *publisherName* --offer *offerName*

Microsoft works closely with partners to ensure the images available are updated and optimized for an Azure runtime. For more information on Azure partner offers, see the Azure Marketplace

Cloud-init

Azure supports for cloud-init across most Linux distributions that support it. We're actively working with our Linux partners to make cloud-init enabled images available in the Azure Marketplace. These images make your cloud-init deployments and configurations work seamlessly with virtual machines and virtual machine scale sets.

For more information, see Using cloud-init on Azure Linux virtual machines.

Storage

- Introduction to Microsoft Azure Storage
- Add a disk to a Linux virtual machine using the azure-cli
- How to attach a data disk to a Linux virtual machine in the Azure portal

Local temporary storage

Some Azure VM sizes include local temporary ephemeral storage, with some of the newer sizes using Temporary local NVMe disks. Local temporary storage uses additional disks provisioned directly as local storage to an Azure virtual machine host, rather than on remote Azure Storage. This type of storage is best suited for data that does not need to be retained permanently, such as caches, buffers, and temporary files. Since data stored on these disks isn't backed up, it's lost when the VM is deallocated or deleted. The ephemeral storage is recreated on startup. Local ephemeral disks are different from Ephemeral OS disks.

Local temporary disks present just like other remote storage such as Premium SSD v2 disks, but have added performance benefits and don't count against the IOPS and throughput limits of the VM.

The following lists key characteristics of local ephemeral storage:

- **Ultra-low latency**: Read/write operations occur on hardware physically co-located with the VM's compute resources, minimizing network hops and enabling sub-millisecond response times.
- **Superior IOPS and throughput**: Performance scales with the underlying local storage, often delivering 8-10 times higher IOPS than Standard HDDs when using Premium SSD as the base, with limits tied to the VM size (up to the NVMe disk capacity, capped at 2,040 GiB).

Azure VM sizes featuring a 'd' in their naming convention - such as the Da***d***sv6, Ea***d***sv6, and FXm***d***sv2 series - include dedicated local NVMe storage. These sizes enable placing temporary files, swap space, or high-I/O components like SQL Server's `tempdb` database files on the ephemeral disk for enhanced performance.

Networking

- Virtual Network Overview
- IP addresses in Azure
- Opening ports to a Linux virtual machine in Azure
- Create a Fully Qualified Domain Name in the Azure portal

Service disruptions

At Microsoft, we work hard to make sure that our services are always available to you when you need them. Forces beyond our control sometimes impact us in ways that cause unplanned service disruptions.

Microsoft provides a Service Level Agreement (SLA) for its services as a commitment for uptime and connectivity. The SLA for individual Azure services can be found at Azure Service Level Agreements.

Azure already has many built-in platform features that support highly available applications. For more about these services, read Disaster recovery and high availability for Azure applications.

This article covers a true disaster recovery scenario, when a whole region experiences an outage due to major natural disaster or widespread service interruption. These are rare occurrences, but you must prepare for the possibility that there's an outage of an entire region. If an entire region experiences a service disruption, the locally redundant copies of your data would temporarily be unavailable. If you enabled geo-replication, three additional copies of your Azure Storage blobs and tables are stored in a different region. In the event of a complete regional outage or a disaster in which the primary region isn't recoverable, Azure remaps all of the DNS entries to the geo-replicated region.

In the case of a service disruption of the entire region where your Azure virtual machine application is deployed, we provide the following guidance for Azure virtual machines.

Option 1: Initiate a failover by using Azure Site Recovery

You can configure Azure Site Recovery for your VMs so that you can recover your application with a single click in matter of minutes. You can replicate to Azure region of your choice and not restricted to paired regions. You can get started by replicating your virtual machines. You can create a recovery plan so that you can automate the entire failover process for your application. You can test your failovers beforehand without impacting production application or the ongoing replication. In the event of a primary region disruption, you just initiate a failover and bring your application in target region.

Option 2: Wait for recovery

In this case, no action on your part is required. Know that we're working diligently to restore service availability. You can see the current service status on our Azure Service Health Dashboard.

This option is the best if you don't set up Azure Site Recovery, read-access geo-redundant storage, or geo-redundant storage prior to the disruption. If you set up geo-redundant storage or read-access geo-redundant storage for the storage account where your VM virtual hard drives (VHDs) are stored, you can look to recover the base image VHD and try to provision a new VM from it. This option isn't preferred because there are no guarantees of synchronization of data, which means this option isn't guaranteed to work.

> **Note**
>
> Be aware that you don't have any control over this process, and it will only occur for region-wide service disruptions. Because of this, you must also rely on other application-

specific backup strategies to achieve the highest level of availability. For more information, see the section on Data strategies for disaster recovery.

Resources for service disruptions

- Start protecting your applications running on Azure virtual machines using Azure Site Recovery
- To learn more about how to implement a disaster recovery and high availability strategy, see Disaster recovery and high availability for Azure applications.
- To develop a detailed technical understanding of a cloud platform's capabilities, see Azure resiliency technical guidance.
- If the instructions aren't clear, or if you would like Microsoft to do the operations on your behalf, contact Customer Support.

Data residency

In Azure, the feature to enable storing customer data in a single region is currently only available in the Southeast Asia Region (Singapore) of the Asia Pacific Geo and Brazil South (Sao Paulo State) Region of Brazil Geo. For all other regions, customer data is stored in Geo. For more information, see Trust Center.

Next steps

Create your first virtual machine!

- Portal
- Azure CLI
- PowerShell

Last updated on 12/01/2025

An overview of Azure VM backup

This article describes how the Azure Backup service backs up Azure virtual machines (VMs).

Azure Backup provides independent and isolated backups to guard against unintended destruction of the data on your VMs. Backups are stored in a Recovery Services vault with built-in management of recovery points. Configuration and scaling are simple, backups are optimized, and you can easily restore as needed.

As part of the backup process, a snapshot is taken, and the data is transferred to the Recovery Services vault with no impact on production workloads. The snapshot provides different levels of consistency, as described here. You can opt for an agent-based application-consistent/file-consistent backup or an agentless crash-consistent backup in the backup policy.

Azure Backup also has specialized offerings for database workloads like SQL Server and SAP HANA that are workload-aware, offer 15 minute RPO (recovery point objective), and allow backup and restore of individual databases.

You can now also back up your virtual machines with Azure Backup in Azure Extended Zones (preview). Azure Extended Zones (preview) provide enhanced resiliency by distributing resources across multiple physical locations within an Azure region. This approach minimizes the impact of potential failures for critical infrastructure. By using Extended Zones, your organizations can achieve higher availability and fault tolerance for their applications. Learn how to back up an Azure VM in Azure Extended Zones (preview).

Backup process

Here's how Azure Backup completes a backup for Azure VMs:

1. For Azure VMs that are selected for backup, Azure Backup starts a backup job according to the backup schedule you specify.

2. If you have opted for application or file-system consistent backups, the VM needs to have a backup extension installed to coordinate for the snapshot process.

   If you have opted for crash-consistent backups, no agents are required in the VMs.

3. During the first backup, a backup extension is installed on the VM if the VM is running.

   - For Windows VMs, the VMSnapshot extension is installed.
   - For Linux VMs, the VMSnapshotLinux extension is installed.

4. For Windows VMs that are running, Azure Backup coordinates with Windows Volume Shadow Copy Service (VSS) to take an app-consistent snapshot of the VM.

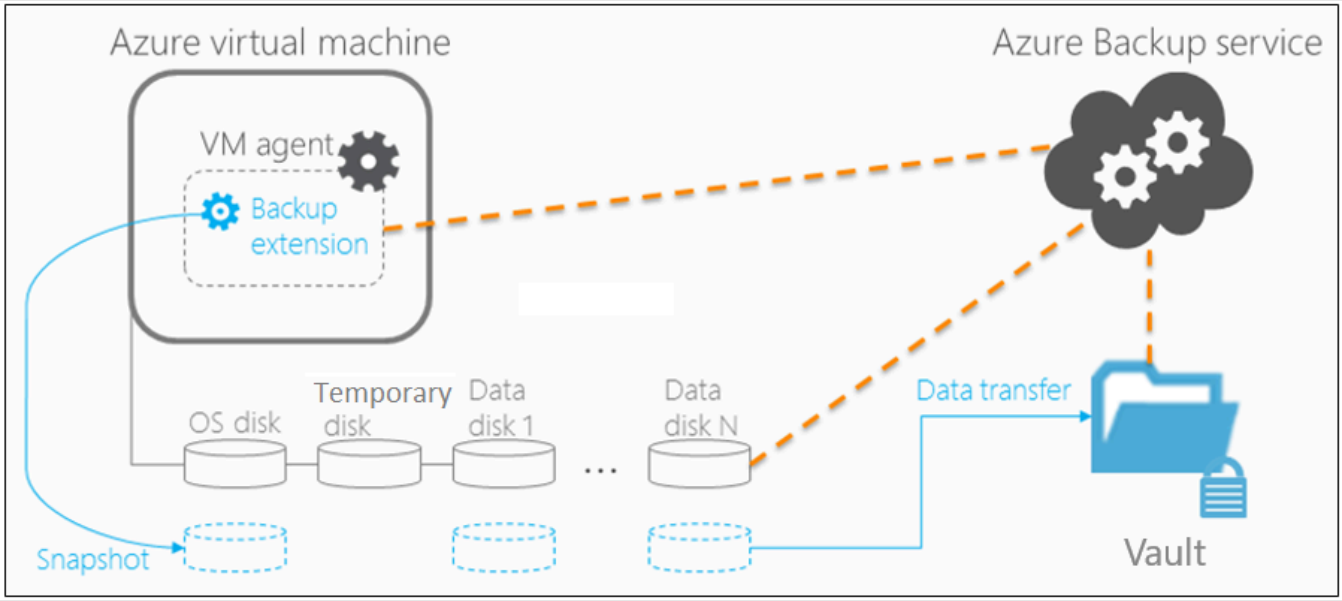
- By default, Backup takes full VSS backups.
- If Backup can't take an app-consistent snapshot, then it takes a file-consistent snapshot of the underlying storage (because no application writes occur while the VM is stopped).

5. For Linux VMs, Backup takes a file-consistent backup. For app-consistent snapshots, you need to manually customize pre/post scripts.

6. For Windows VMs, **Microsoft Visual C++ 2015 Redistributable (x64) version 14.40.33810.0** is installed, the startup of Volume Shadow Copy Service (VSS) is changed to automatic, and a Windows Service **IaaSVmProvider** is added.

7. After Backup takes the snapshot, it transfers the data to the vault.

    - The backup is optimized by backing up each VM disk in parallel.
    - For each disk that's being backed up, Azure Backup reads the blocks on the disk and identifies and transfers only the data blocks that changed (the delta) since the previous backup.
    - Snapshot data might not be immediately copied to the vault. It might take some hours at peak times. Total backup time for a VM will be less than 24 hours for daily backup policies.



Encryption of Azure VM backups

When you back up Azure VMs with Azure Backup, VMs are encrypted at rest with Storage Service Encryption (SSE). Azure Backup can also back up Azure VMs that are encrypted by using Azure Disk Encryption.

Expand table

Encryption	Details	Support
SSE	With SSE, Azure Storage provides encryption at rest by automatically encrypting data before storing it. Azure Storage also decrypts data before retrieving it. Azure Backup supports backups of VMs with two types of Storage Service Encryption: • **SSE with platform-managed keys**: This encryption is by default for all disks in your VMs. See more here. • **SSE with customer-managed keys**. With CMK, you manage the keys used to encrypt the disks. See more here.	Azure Backup uses SSE for at-rest encryption of Azure VMs.
Azure Disk Encryption	Azure Disk Encryption encrypts both OS and data disks for Azure VMs. Azure Disk Encryption integrates with BitLocker encryption keys (BEKs), which are safeguarded in a key vault as secrets. Azure Disk Encryption also integrates with Azure Key Vault key encryption keys (KEKs).	Azure Backup supports backup of managed and unmanaged Azure VMs encrypted with BEKs only, or with BEKs together with KEKs. Both BEKs and KEKs are backed up and encrypted. Because KEKs and BEKs are backed up, users with the necessary permissions can restore keys and secrets back to the key vault if needed. These users can also recover the encrypted VM. Encrypted keys and secrets can't be read by unauthorized users or by Azure.

For managed and unmanaged Azure VMs, Backup supports both VMs encrypted with BEKs only or VMs encrypted with BEKs together with KEKs.

The backed-up BEKs (secrets) and KEKs (keys) are encrypted. They can be read and used only when they're restored back to the key vault by authorized users. Neither unauthorized users, or Azure, can read or use backed-up keys or secrets.

BEKs are also backed up. So, if the BEKs are lost, authorized users can restore the BEKs to the key vault and recover the encrypted VMs. Only users with the necessary level of permissions can back up and restore encrypted VMs or keys and secrets.

Snapshot creation

Azure Backup takes snapshots according to the backup schedule.

If you have opted for application or file-system-consistent backups, the VM needs to have a backup extension installed to coordinate for the snapshot process. For *agentless multi-disk crash-consistent* backups, the VM agent is not required for snapshots.

- **Windows VMs:** For Windows VMs, the Backup service coordinates with VSS to take an app-consistent snapshot of the VM disks. By default, Azure Backup takes a full VSS backup (it truncates the logs of application such as SQL Server at the time of backup to get application level consistent backup). If you're using a SQL Server database on Azure VM backup, then you can modify the setting to take a VSS Copy backup (to preserve logs). For more information, see this article.

- **Linux VMs:** To take app-consistent snapshots of Linux VMs, use the Linux pre-script and post-script framework to write your own custom scripts to ensure consistency.
  - Azure Backup invokes only the pre/post scripts written by you.
  - If the pre-scripts and post-scripts execute successfully, Azure Backup marks the recovery point as application-consistent. However, when you're using custom scripts, you're ultimately responsible for the application consistency.
  - Learn more about how to configure scripts.

- **FSFreeze and safeFreezelockFile**: **Fsfreeze** is a utility used to freeze and thaw the *filesystem* to make their state on consistent during VM backups.

  - **Freeze**: When you freeze a filesystem, the currently running operations are allowed to complete. However, new write system operations and the operations that modify the filesystem are stopped.

  - **Thawing**: This operation reverses the freeze operation. When you thaw a filesystem, the previously stopped operations continue to run effectively, unblocking any modifications.

  During the *Freeze* operation, Azure Backup invokes a file safeFreezelockFile. This prevents the other process to invoke the similar operation so that backup process isn't interrupted. Once the operation is complete, the lock is released.

> ⓘ **Note**
>
> The snapshot process starts with invoking the backup extension (agent-based), using tools such as VSS for Windows or freeze for Linux to ensure data consistency. When consistent, the Restore Point Collection monitors the extension and initiates snapshot creation. Afterward, metadata is updated with restore point paths and commit details to finalize the backup for restoration.

Snapshot consistency

The following table explains the different types of snapshot consistency:

Expand table

Snapshot	Details	Recovery	Consideration
Application-consistent	This is the default setting in the VM backup policy. App-consistent backups capture memory content and pending I/O operations. App-consistent snapshots use a VSS writer (or pre/post scripts for Linux) to ensure the consistency of the app data before a backup occurs.	When you're recovering a VM with an app-consistent snapshot, the VM boots up. There's no data corruption or loss. The apps start in a consistent state.	Windows: All VSS writers succeeded Linux: Pre/post scripts are configured and succeeded
File-system consistent	This is the default setting in the VM backup policy. File-system consistent backups provide consistency by taking a snapshot of all files at the same time.	When you're recovering a VM with a file-system consistent snapshot, the VM boots up. There's no data corruption or loss. Apps need to implement their own "fix-up" mechanism to make sure that restored data is consistent.	Windows: Some VSS writers failed Linux: Default (if pre/post scripts aren't configured or failed)
Crash-consistent	Crash-consistent snapshot is an opt-in setting in the VM backup policy. Azure Backup also takes crash-consistent backups if the VM is not running during backup and when application/file-consistent backups fail. Only the data that already exists on the disk at the time of the backup operation is captured and backed up; data in read/write host cache isn't captured.	Starts with the VM boot process followed by a disk check to fix corruption errors. Any in-memory data or write operations that weren't transferred to disk before the crash are lost. Apps implement their own data verification. For example, a database app can use its transaction log for verification. If the transaction log has entries that aren't in the database, the database software rolls transactions back until the data is consistent.	When you have opted for application/file-system backup and VM is in shutdown (stopped/ deallocated) state and when the snapshot is retried. You have opted for agentless crash consistent backups

> **Note**

If the provisioning state is **succeeded**, Azure Backup takes file-system consistent backups. If the provisioning state is **unavailable** or **failed**, crash-consistent backups are taken. If the provisioning state is **creating** or **deleting**, that means Azure Backup is retrying the operations.

Backup and restore considerations

Expand table

Consideration	Details
Disk	Backup of VM disks is parallel. For example, if a VM has four disks, the Backup service attempts to back up all four disks in parallel. Backup is incremental (only changed data).
Scheduling	To reduce backup traffic, back up different VMs at different times of the day and make sure the times don't overlap. Backing up VMs at the same time causes traffic jams.
Preparing backups	Keep in mind the time needed to prepare the backup. The preparation time might include installing or updating the backup extension and triggering a snapshot according to the backup schedule.
Data transfer	Consider the time needed for Azure Backup to identify the incremental changes from the previous backup. In an incremental backup, Azure Backup determines the changes by calculating the checksum of the block. If a block is changed, it's marked for transfer to the vault. The service analyzes the identified blocks to attempt to further minimize the amount of data to transfer. After the evaluation of all changed blocks is complete, Azure Backup transfers the changes to the vault. There might be a lag between taking the snapshot and copying it to vault. At peak times, it can take up to eight hours for the snapshots to be transferred to the vault. The backup time for a VM will be less than 24 hours for the daily backup.
Initial backup	The total backup time for incremental backups is less than 24 hours, which might not be the case for the first backup. The time needed for the initial backup will depend on the size of the data and when the backup is processed.
Restore queue	Azure Backup processes restore jobs from multiple storage accounts at the same time, and it puts restore requests in a queue.
Restore copy	During the restore process, data is copied from the vault to the storage account. The total restore time depends on the I/O operations per second (IOPS) and the throughput of the storage account.

Consideration	Details
	To reduce the copy time, select a storage account that isn't loaded with other application writes and reads.

> **Note**
>
> Azure Backup now enables you to back up your Azure VMs multiple times a day using the Enhanced policy. With this capability, you can also define the duration in which your backup jobs would trigger and align your backup schedule with the working hours when there are frequent updates to Azure Virtual Machines. Learn more.

Backup performance

These common scenarios can affect the total backup time:

- **Adding a new disk to a protected Azure VM:** If a VM is undergoing incremental backup and a new disk is added, the backup time will increase. The total backup time might last more than 24 hours because of initial replication of the new disk, along with delta replication of existing disks.
- **Fragmented disks:** Backup operations are faster when disk changes are contiguous. If changes are spread out and fragmented across a disk, backup will be slower.
- **Disk churn:** If protected disks that are undergoing incremental backup have a daily churn of more than 200 GB, backup can take a long time (more than eight hours) to complete.
- **Backup versions:** The latest version of Backup (known as the Instant Restore version) uses a more optimized process than checksum comparison for identifying changes. But if you're using Instant Restore and have deleted a backup snapshot, the backup switches to checksum comparison. In this case, the backup operation will exceed 24 hours (or fail).

Restore performance

These common scenarios can affect the total restore time:

- The total restore time depends on the Input/output operations per second (IOPS) and the throughput of the storage account.
- The total restore time can be affected if the target storage account is loaded with other application read and write operations. To improve restore operation, select a storage account that isn't loaded with other application data.

Best practices

When you're configuring VM backups, we suggest following these practices:

- Modify the default schedule times that are set in a policy. For example, if the default time in the policy is 12:00 AM, increment the timing by several minutes so that resources are optimally used.
- If you're restoring VMs from a single vault, we highly recommend that you use different general-purpose v2 storage accounts to ensure that the target storage account doesn't get throttled. For example, each VM must have a different storage account. For example, if 10 VMs are restored, use 10 different storage accounts.
- For backup of VMs that are using premium storage with Instant Restore, we recommend allocating *50%* free space of the total allocated storage space, which is required **only** for the first backup. The 50% free space isn't a requirement for backups after the first backup is complete
- The limit on the number of disks per storage account is relative to how heavily the disks are being accessed by applications that are running on an infrastructure as a service (IaaS) VM. As a general practice, if 5 to 10 disks or more are present on a single storage account, balance the load by moving some disks to separate storage accounts.
- To restore VMs with managed disks using PowerShell, provide the additional parameter ***TargetResourceGroupName*** to specify the resource group to which managed disks will be restored, Learn more here.

Backup costs

Azure VMs backed up with Azure Backup are subject to Azure Backup pricing.

Billing doesn't start until the first successful backup finishes. At this point, the billing for both storage and protected VMs begins. Billing continues as long as any backup data for the VM is stored in a vault. If you stop protection for a VM, but backup data for the VM exists in a vault, billing continues.

Billing for a specified VM stops only if the protection is stopped and all backup data is deleted. When protection stops and there are no active backup jobs, the size of the last successful VM backup becomes the protected instance size used for the monthly bill.

If you have opted for agent-based application consistent or file-system consistent backups, the protected-instance size calculation is based on the *actual* size of the VM. The VM's size is the sum of all the data in the VM, excluding the temporary storage. Pricing is based on the actual data that's stored on the data disks, not on the maximum supported size for each data disk that's attached to the VM.

> **Note**

The agentless multi-disk crash consistent VM backup feature is generally available. This release includes changes to billing; see the pricing details.

Similarly, the backup storage bill is based on the amount of data that's stored in Azure Backup, which is the sum of the actual data in each recovery point.

For example, take an A2-Standard-sized VM that has two additional data disks with a maximum size of 32 TB each. The following table shows the actual data stored on each of these disks:

Expand table

Disk	Max size	Actual data present
OS disk	32 TB	17 GB
Local/temporary disk	135 GB	5 GB (not included for backup)
Data disk 1	32 TB	30 GB
Data disk 2	32 TB	0 GB

The actual size of the VM in this case is 17 GB + 30 GB + 0 GB = 47 GB. This protected-instance size (47 GB) becomes the basis for the monthly bill. As the amount of data in the VM grows, the protected-instance size used for billing changes to match.

Next steps

- Prepare for Azure VM backup.

Last updated on 01/08/2026

Quickstart: Back up a virtual machine in Azure with the Azure CLI

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. You can protect your data by taking backups at regular intervals. Azure Backup creates recovery points that can be stored in geo-redundant recovery vaults. This article details how to back up a virtual machine (VM) in Azure with the Azure CLI. You can also perform these steps with Azure PowerShell or in the Azure portal.

This quickstart enables backup on an existing Azure VM. If you need to create a VM, you can create a VM with the Azure CLI.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Get started with Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Authenticate to Azure using Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use and manage extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.
- This quickstart requires version 2.0.18 or later of the Azure CLI. If using Azure Cloud Shell, the latest version is already installed.

Create a Recovery Services vault for Azure VM backup

A Recovery Services vault is a logical container that stores the backup data for each protected resource, such as Azure VMs. When the backup job for a protected resource runs, it creates a

recovery point inside the Recovery Services vault. You can then use one of these recovery points to restore data to a given point in time.

Create a Recovery Services vault with az backup vault create. Specify the same resource group and location as the VM you wish to protect. If you used the VM quickstart, then you created:

- a resource group named *myResourceGroup*,
- a VM named *myVM*,
- resources in the *eastus* location.

Azure CLI

```
az backup vault create --resource-group myResourceGroup \
    --name myRecoveryServicesVault \
    --location eastus
```

By default, the Recovery Services vault is set for Geo-Redundant storage. Geo-Redundant storage ensures your backup data is replicated to a secondary Azure region that's hundreds of miles away from the primary region. If the storage redundancy setting needs to be modified, use az backup vault backup-properties set cmdlet.

Azure CLI

```
az backup vault backup-properties set \
    --name myRecoveryServicesVault  \
    --resource-group myResourceGroup \
    --backup-storage-redundancy "LocallyRedundant/GeoRedundant"
```

Enable backup for an Azure VM

Create a protection policy to define: when a backup job runs, and how long the recovery points are stored. The default protection policy runs a backup job each day and retains recovery points for 30 days. You can use these default policy values to quickly protect your VM. To enable backup protection for a VM, use az backup protection enable-for-vm. Specify the resource group and VM to protect, then the policy to use:

Azure CLI

```
az backup protection enable-for-vm \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --vm myVM \
    --policy-name DefaultPolicy
```

> **Note**
>
> If the VM isn't in the same resource group as that of the vault, then myResourceGroup refers to the resource group where vault was created. Instead of VM name, provide the VM ID as indicated below.

Azure CLI

```
az backup protection enable-for-vm \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --vm $(az vm show -g VMResourceGroup -n MyVm --query id | tr -d '"') \
    --policy-name DefaultPolicy
```

> **Important**
>
> While using CLI to enable backup for multiple VMs at once, ensure that a single policy doesn't have more than 100 VMs associated with it. This is a recommended best practice. Currently, the PowerShell client doesn't explicitly block if there are more than 100 VMs, but the check is planned to be added in the future.

Prerequisites to backup encrypted VMs

To enable protection on encrypted VMs (encrypted using BEK and KEK), you must provide the Azure Backup service permission to read keys and secrets from the key vault. To do so, set a *keyvault* access policy with the required permissions, as demonstrated below:

Azure CLI

```
# Enter the name of the resource group where the key vault is located on this
variable
AZ_KEYVAULT_RGROUP=TestKeyVaultRG

# Enter the name of the key vault on this variable
AZ_KEYVAULT_NAME=TestKeyVault

# Get the object id for the Backup Management Service on your subscription
AZ_ABM_OBJECT_ID=$( ad sp list --display-name "Backup Management Service" --query
'[].id' -o tsv --only-show-errors )

# This command will grant the permissions required by the Backup Management Service
to access the key vault
az keyvault set-policy --key-permissions get list backup --secret-permissions get
list backup \
```

```
  --resource-group $AZ_KEYVAULT_RGROUP --name $AZ_KEYVAULT_NAME --object-id
$AZ_ABM_OBJECT_ID
```

Trigger an Azure VM backup job

To trigger a backup job rather than wait for the default policy to run the job at the scheduled time, use az backup protection backup-now. This first backup job creates a full recovery point. Each backup job after this initial backup creates incremental recovery points. Incremental recovery points are storage and time-efficient, as they only transfer changes made since the last backup.

The following parameters are used to back up the VM:

- `--container-name` is the name of your VM
- `--item-name` is the name of your VM
- `--retain-until` value should be set to the last available date, in UTC time format (**dd-mm-yyyy**), that you wish the recovery point to be available

The following example backs up the VM named *myVM* and sets the expiration of the recovery point to October 18, 2017:

Azure CLI

```
az backup protection backup-now \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --container-name myVM \
    --item-name myVM \
    --backup-management-type AzureIaaSVM
    --retain-until 18-10-2017
```

Monitor the backup job for the Azure VM

To monitor the status of backup jobs, use az backup job list:

Azure CLI

```
az backup job list \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --output table
```

The output is similar to the following example, which shows the backup job is *InProgress*:

Output

```
Name      Operation         Status      Item Name    Start Time UTC       Duration
--------  ---------------   ----------  -----------  -------------------  -----------
---
a0a8e5e6  Backup            InProgress  myvm         2017-09-19T03:09:21
0:00:48.718366
fe5d0414  ConfigureBackup   Completed   myvm         2017-09-19T03:03:57
0:00:31.191807
```

When the *Status* of the backup job reports *Completed*, your VM is protected with Recovery Services and has a full recovery point stored.

Clean up deployment for the Azure VM backup

When no longer needed, you can disable protection on the VM, remove the restore points and Recovery Services vault, then delete the resource group and associated VM resources. If you used an existing VM, you can skip the final az group delete command to leave the resource group and VM in place.

If you want to try a Backup tutorial that explains how to restore data for your VM, go to Next steps.

Azure CLI

```azurecli
az backup protection disable \
    --resource-group myResourceGroup \
    --vault-name myRecoveryServicesVault \
    --container-name myVM \
    --item-name myVM \
    --backup-management-type AzureIaaSVM
    --delete-backup-data true
az backup vault delete \
    --resource-group myResourceGroup \
    --name myRecoveryServicesVault \
az group delete --name myResourceGroup
```

Next steps

In this quickstart, you created a Recovery Services vault, enabled protection on a VM, and created the initial recovery point. To learn more about Azure Backup and Recovery Services, continue to the tutorials.

- Back up multiple Azure VMs
- Restore an Azure VM using REST API

Last updated on 12/19/2025

Quickstart: Back up a virtual machine in Azure with PowerShell

The Azure PowerShell AZ module is used to create and manage Azure resources from the command line or in scripts.

Azure Backup backs up on-premises machines and apps, and Azure VMs. This article shows you how to back up an Azure VM with the AZ module. Alternatively, you can back up a VM using the Azure CLI, or in the Azure portal.

This quickstart enables backup on an existing Azure VM. If you need to create a VM, you can create a VM with Azure PowerShell.

This quickstart requires the Azure PowerShell AZ module version 1.0.0 or later. Run `Get-Module -ListAvailable Az` to find the version. If you need to install or upgrade, see Install Azure PowerShell module.

> **Note**
>
> We recommend that you use the Azure Az PowerShell module to interact with Azure. To get started, see Install Azure PowerShell. To learn how to migrate to the Az PowerShell module, see Migrate Azure PowerShell from AzureRM to Az.

Sign in and register

1. Sign in to your Azure subscription with the `Connect-AzAccount` command and follow the on-screen directions.

   PowerShell

   ```
   Connect-AzAccount
   ```

2. The first time you use Azure Backup, you must register the Azure Recovery Service provider in your subscription with Register-AzResourceProvider, as follows:

   PowerShell

   ```
   Register-AzResourceProvider -ProviderNamespace "Microsoft.RecoveryServices"
   ```

Create a Recovery Services vault for Azure VM backup

A Recovery Services vault is a logical container that stores backup data for protected resources, such as Azure VMs. When a backup job runs, it creates a recovery point inside the Recovery Services vault. You can then use one of these recovery points to restore data to a given point in time.

When you create the vault:

- For the resource group and location, specify the resource group and location of the VM you want to back up.
- If you used this sample script to create the VM, the resource group is **myResourceGroup**, the VM is ***myVM**, and the resources are in the **WestEurope** region.
- Azure Backup automatically handles storage for backed up data. By default the vault uses Geo-Redundant Storage (GRS). Geo-redundancy ensures that backed up data is replicated to a secondary Azure region, hundreds of miles away from the primary region.

Now create a vault:

1. Use the New-AzRecoveryServicesVault to create the vault:

   PowerShell

   ```
   New-AzRecoveryServicesVault `
       -ResourceGroupName "myResourceGroup" `
       -Name "myRecoveryServicesVault" `
   -Location "WestEurope"
   ```

2. Set the vault context with Set-AzRecoveryServicesVaultContext, as follows:

   PowerShell

   ```
   Get-AzRecoveryServicesVault `
       -Name "myRecoveryServicesVault" | Set-AzRecoveryServicesVaultContext
   ```

3. Change the storage redundancy configuration (LRS/GRS) of the vault with Set-AzRecoveryServicesBackupProperty, as follows:

   PowerShell

   ```
   Get-AzRecoveryServicesVault `
       -Name "myRecoveryServicesVault" | Set-AzRecoveryServicesBackupProperty -
   BackupStorageRedundancy LocallyRedundant/GeoRedundant
   ```

> **Note**
>
> Storage Redundancy can be modified only if there are no backup items protected to the vault.

Enable backup for an Azure VM

You enable backup for an Azure VM, and specify a backup policy.

- The policy defines when backups run, and how long recovery points created by the backups should be retained.
- The default protection policy runs a backup once a day for the VM, and retains the created recovery points for 30 days. You can use this default policy to quickly protect your VM.

Enable backup as follows:

1. First, set the default policy with Get-AzRecoveryServicesBackupProtectionPolicy:

   PowerShell

   ```
   $policy = Get-AzRecoveryServicesBackupProtectionPolicy     -Name
   "DefaultPolicy"
   ```

2. Enable VM backup with Enable-AzRecoveryServicesBackupProtection. Specify the policy, the resource group, and the VM name.

   PowerShell

   ```
   Enable-AzRecoveryServicesBackupProtection `
       -ResourceGroupName "myResourceGroup" `
       -Name "myVM" `
       -Policy $policy
   ```

Trigger an Azure VM backup job

Backups run according to the schedule specified in the backup policy. You can also run an on-demand backup:

- The first initial backup job creates a full recovery point.
- After the initial backup, each backup job creates incremental recovery points.

- Incremental recovery points are storage and time-efficient, as they only transfer changes made since the last backup.

To run an on-demand backup, you use the Backup-AzRecoveryServicesBackupItem.

- You specify a container in the vault that holds your backup data with Get-AzRecoveryServicesBackupContainer.
- Each VM to back up is treated as an item. To start a backup job, you obtain information about the VM with Get-AzRecoveryServicesBackupItem.

Run an on-demand backup job as follows:

1. Specify the container, obtain VM information, and run the backup.

   PowerShell

   ```
   $backupcontainer = Get-AzRecoveryServicesBackupContainer `
       -ContainerType "AzureVM" `
       -FriendlyName "myVM"

   $item = Get-AzRecoveryServicesBackupItem `
       -Container $backupcontainer `
       -WorkloadType "AzureVM"

   Backup-AzRecoveryServicesBackupItem -Item $item
   ```

2. You might need to wait up to 20 minutes, since the first backup job creates a full recovery point. Monitor the job as described in the next procedure.

Monitor the backup job for the Azure VM

To monitor the status of backup jobs, use Get-AzRecoveryservicesBackupJob cmdlet.

PowerShell

```
Get-AzRecoveryservicesBackupJob
```

Output is similar to the following example, which shows the job as **InProgress**:

Output

```
WorkloadName   Operation         Status       StartTime              EndTime
JobID
------------   ---------         ------       ---------              -------
-----
myvm           Backup            InProgress   9/18/2017 9:38:02 PM
9f9e8f14
```

```
myvm            ConfigureBackup    Completed    9/18/2017 9:33:18 PM    9/18/2017
9:33:51 PM    fe79c739
```

When the job status is **Completed**, the VM is protected and has a full recovery point stored.

Manage VM backups

If you want to perform more actions such as change policy, edit policy etc.. refer to the manage VM backups section.

Clean up the deployment for Azure VM backup

If you no longer need to back up the VM, you can clean it up.

- If you want to try out restoring the VM, skip the clean-up.
- If you used an existing VM, you can skip the final Remove-AzResourceGroup cmdlet to leave the resource group and VM in place.

Disable protection, remove the restore points and vault. Then delete the resource group and associated VM resources, as follows:

PowerShell

```powershell
Disable-AzRecoveryServicesBackupProtection -Item $item -RemoveRecoveryPoints
$vault = Get-AzRecoveryServicesVault -Name "myRecoveryServicesVault"
Remove-AzRecoveryServicesVault -Vault $vault
Remove-AzResourceGroup -Name "myResourceGroup"
```

Next steps

In this quickstart, you created a Recovery Services vault, enabled protection on a VM, and created the initial recovery point.

- Learn how to back up VMs in the Azure portal.
- Learn how to quickly restore a VM
- Learn how to restore an Azure VM using REST API

Last updated on 12/19/2025

Quickstart: Back up a virtual machine in Azure

Summarize this article for me

This quickstart describes how to enable backup on an existing Azure VM by using the Azure portal. If you need to create a VM, you can create a VM with the Azure portal.

Azure backups can be created through the Azure portal. This method provides a browser-based user interface to create and configure Azure backups and all related resources. You can protect your data by taking backups at regular intervals. Azure Backup creates recovery points that can be stored in geo-redundant recovery vaults. This article details how to back up a virtual machine (VM) with the Azure portal.

Sign in to Azure

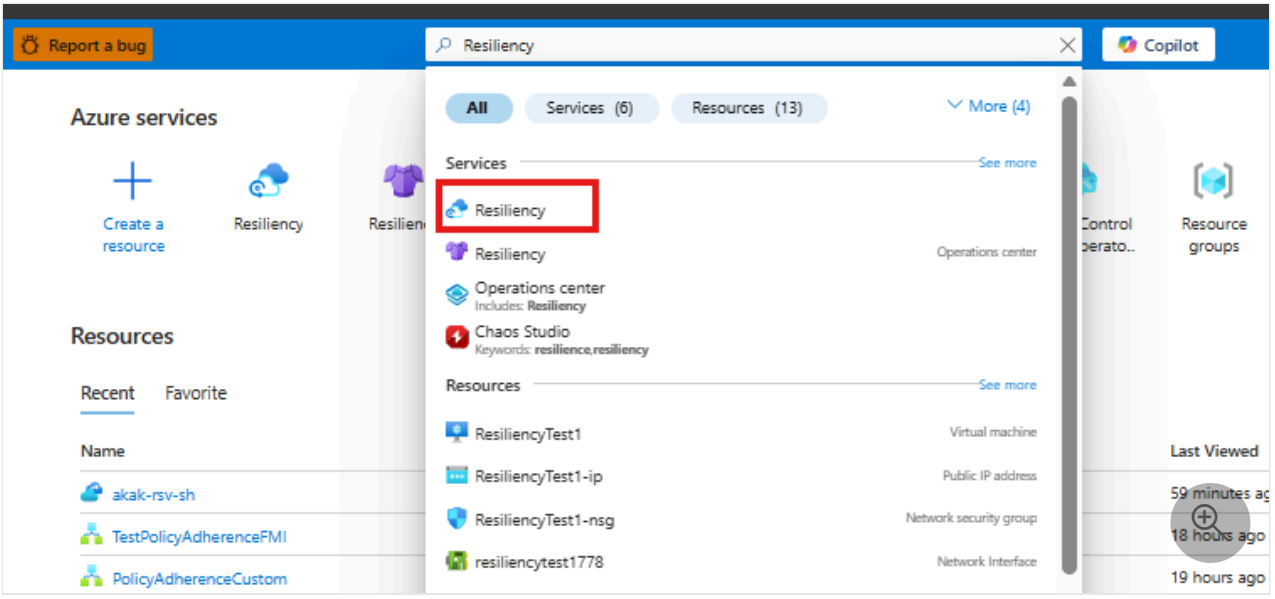
Sign in to the Azure portal.

Create a Recovery Services vault

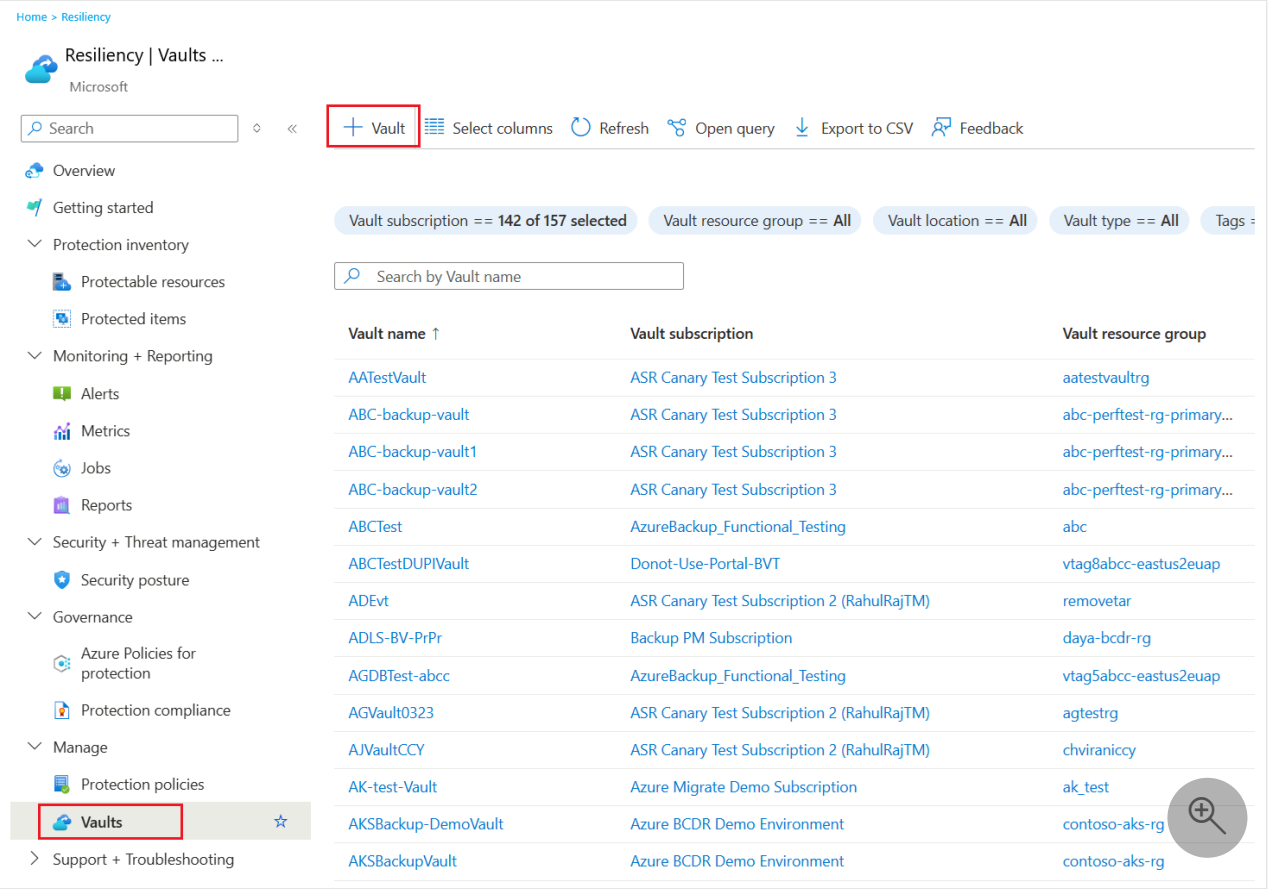
A Recovery Services vault is a management entity that stores recovery points that are created over time. It provides an interface to perform backup-related operations. These operations include taking on-demand backups, performing restores, and creating backup policies.

To create a Recovery Services vault:

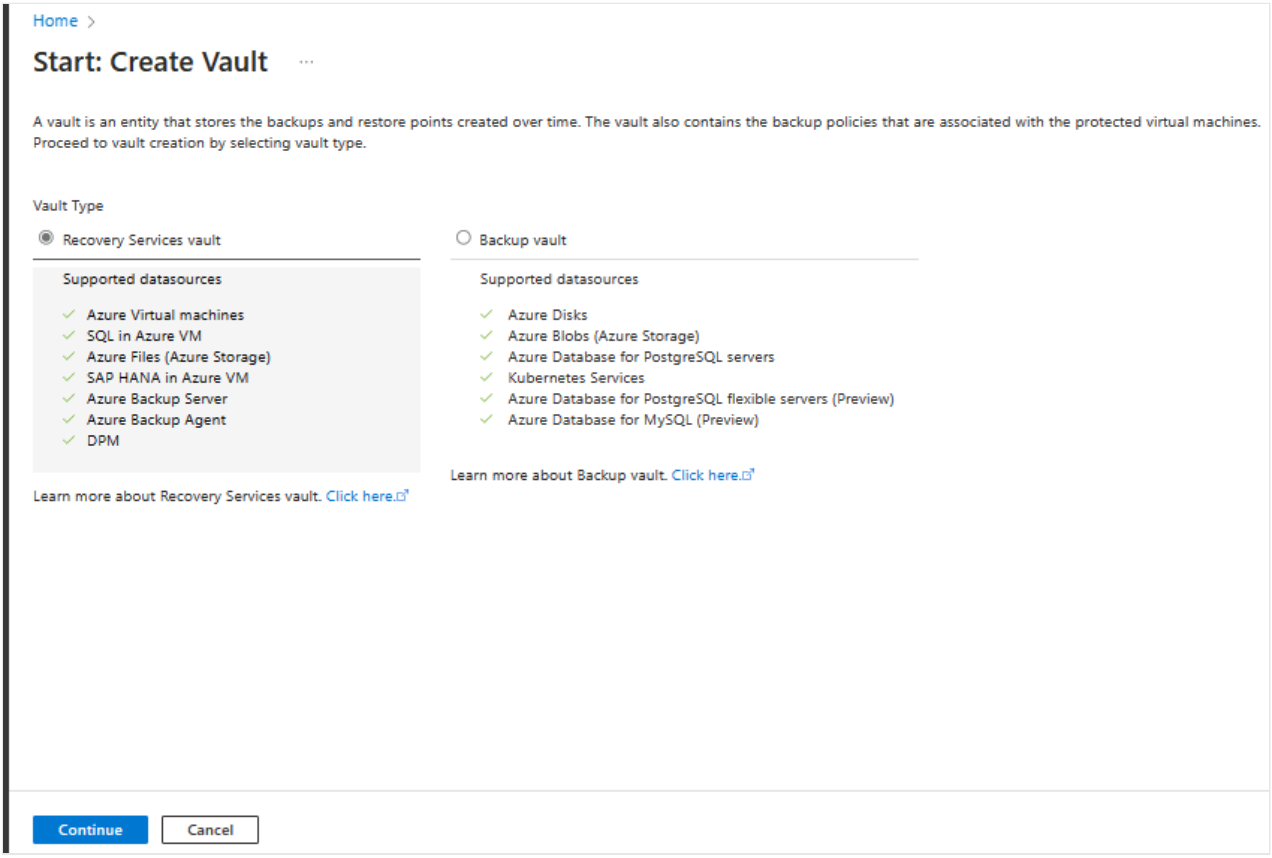
1. Sign in to the Azure portal.
2. Search for **Resiliency**, and then go to the **Resiliency** dashboard.



3. On the **Vault** pane, select **+ Vault**.



4. Select **Recovery Services vault** > **Continue**.



5. On the **Create Recovery Services vault** pane, enter the following values:

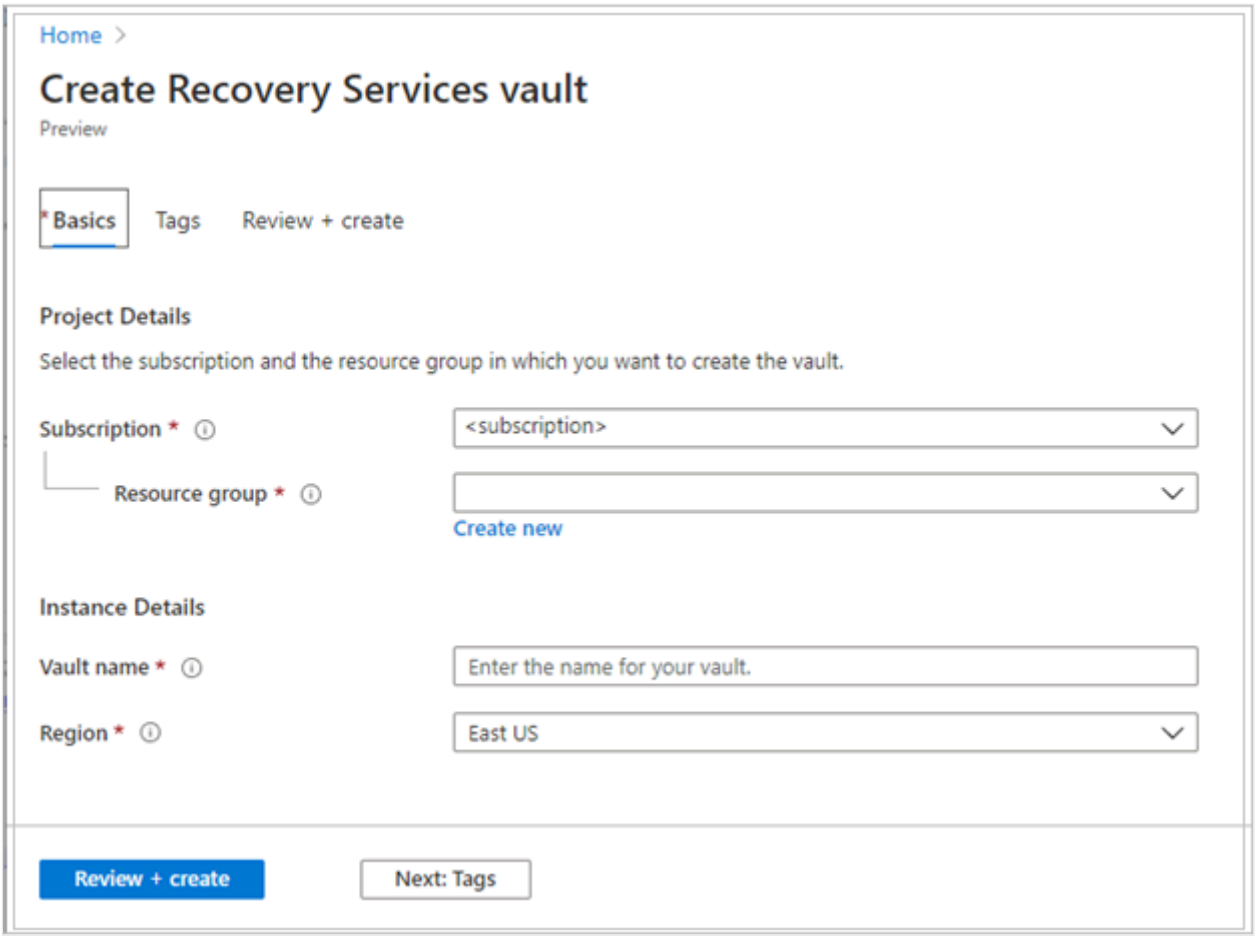
- **Subscription**: Select the subscription to use. If you're a member of only one subscription, you see that name. If you're not sure which subscription to use, use the default subscription. Multiple choices appear only if your work or school account is associated with more than one Azure subscription.

- **Resource group**: Use an existing resource group or create a new one. To view a list of available resource groups in your subscription, select **Use existing**. Then select a resource in the dropdown list. To create a new resource group, select **Create new**, and then enter the name. For more information about resource groups, see Azure Resource Manager overview.

- **Vault name**: Enter a friendly name to identify the vault. The name must be unique to the Azure subscription. Specify a name that has at least 2 but not more than 50 characters. The name must start with a letter and consist only of letters, numbers, and hyphens.

- **Region**: Select the geographic region for the vault. For you to create a vault to help protect any data source, the vault *must* be in the same region as the data source.

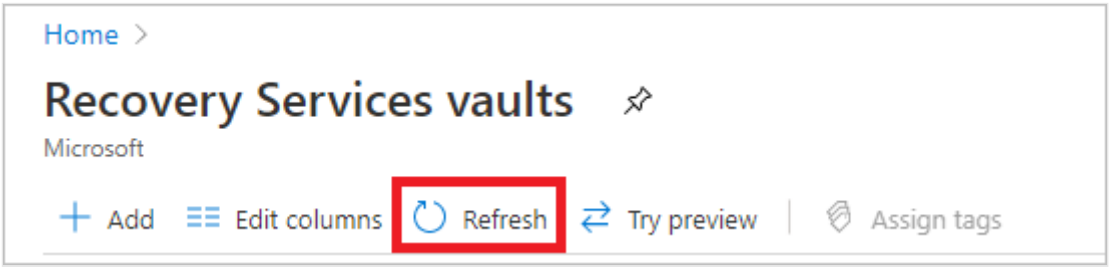
  > **Important**
  >
  > If you're not sure of the location of your data source, close the window. Go to the list of your resources in the portal. If you have data sources in multiple regions, create a Recovery Services vault for each region. Create the vault in the first location before you create a vault in another location. You don't need to specify storage accounts to store the backup data. The Recovery Services vault and Azure Backup handle that step automatically.



6. After you provide the values, select **Review + create**.

7. To finish creating the Recovery Services vault, select **Create**.

   It can take a while to create the Recovery Services vault. Monitor the status notifications in the **Notifications** area at the upper right. After the vault is created, it appears in the list of Recovery Services vaults. If the vault doesn't appear, select **Refresh**.



Azure Backup now supports immutable vaults that help you ensure that after recovery points are created, they can't be deleted before their expiry according to the backup policy. You can make the immutability irreversible to help protect your backup data from various threats, including ransomware attacks and malicious actors. Learn more about Azure Backup immutable vaults.

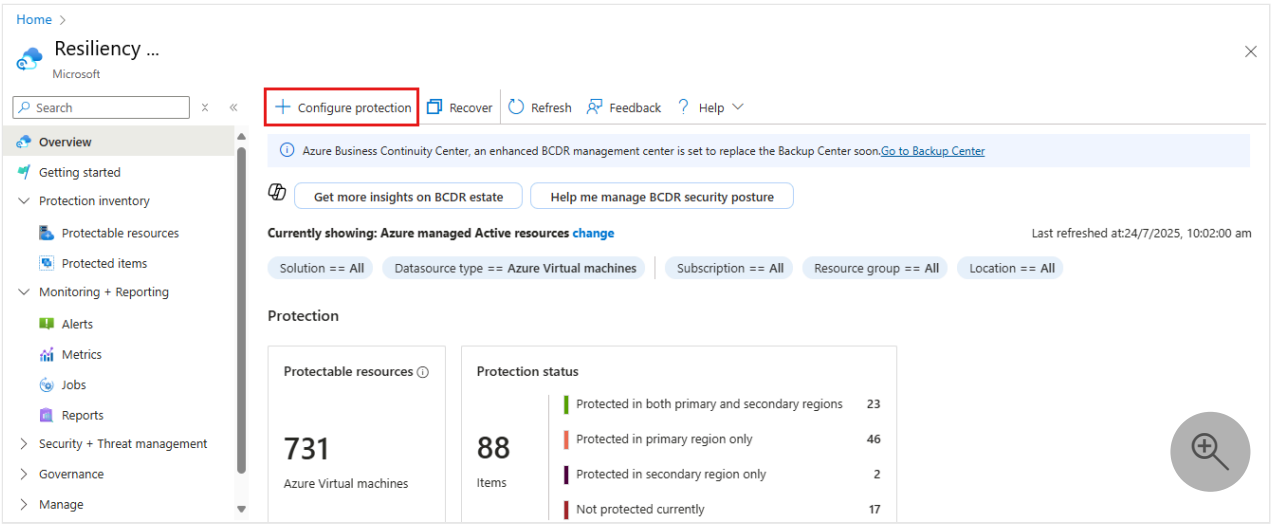
> **Important**

If you have Azure Files for protection, after vault creation, configure backup for Azure Files, and then initiate an on-demand backup. Learn more about the best practices for Azure Files backup.

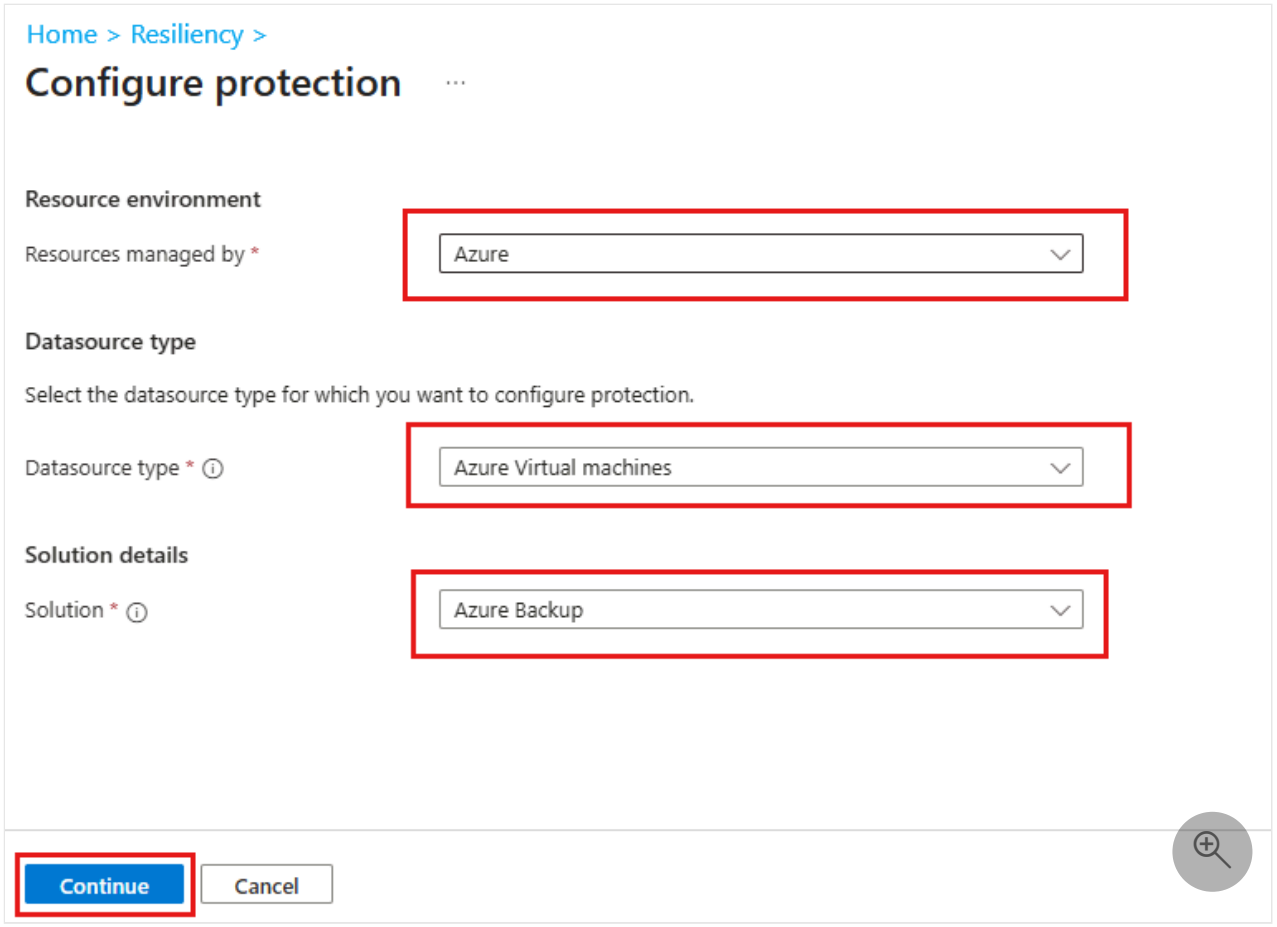
Apply a backup policy

To apply a backup policy to your Azure VMs, follow these steps:

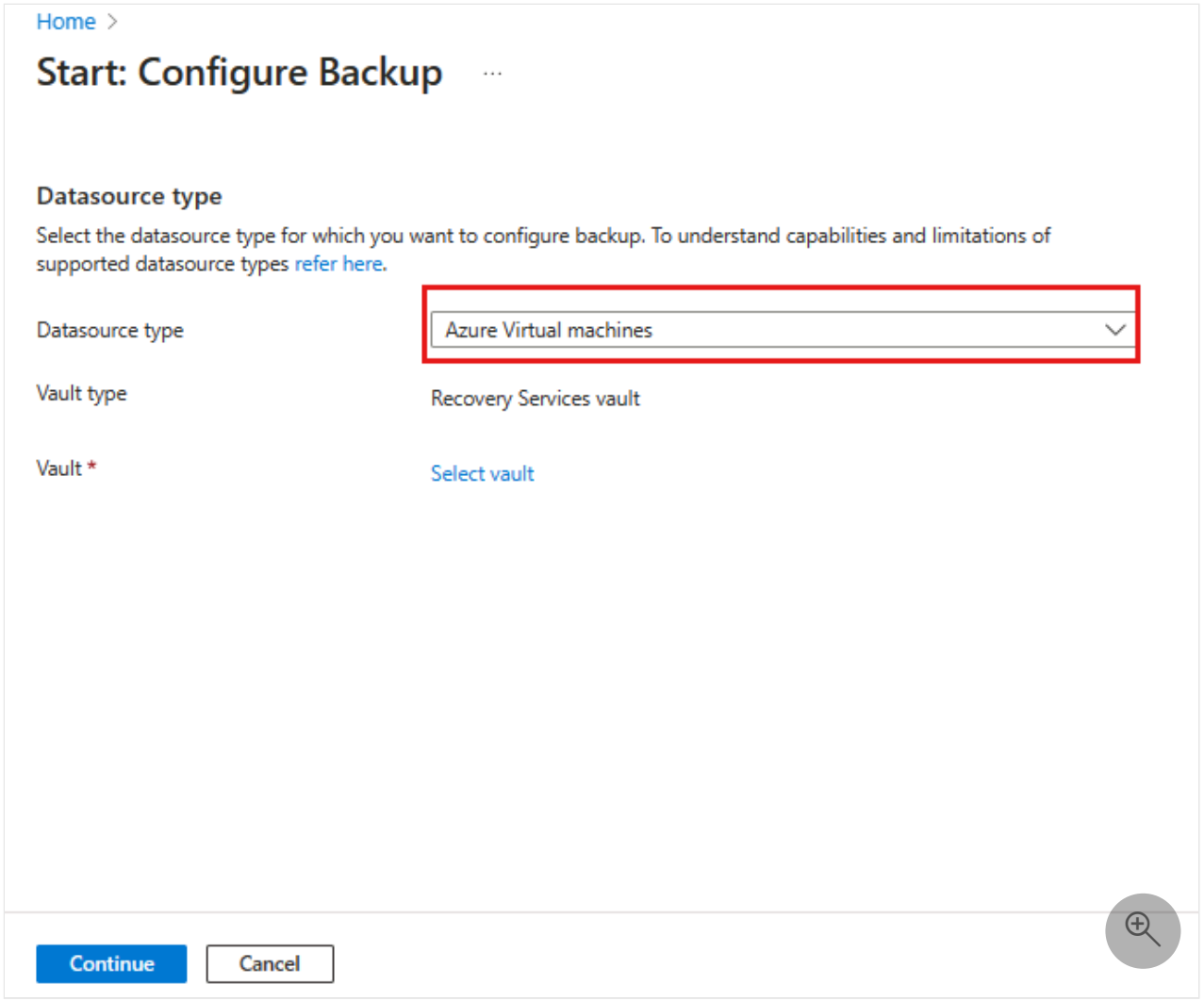
1. Go to **Resiliency** and select **+ Configure protection**.



2. On the **Configure protection** pane, select **Resource managed by** as **Azure**, **Datasource type** as **Azure Virtual machines**, **Solution** as **Azure Backup**, and then select **Continue**.

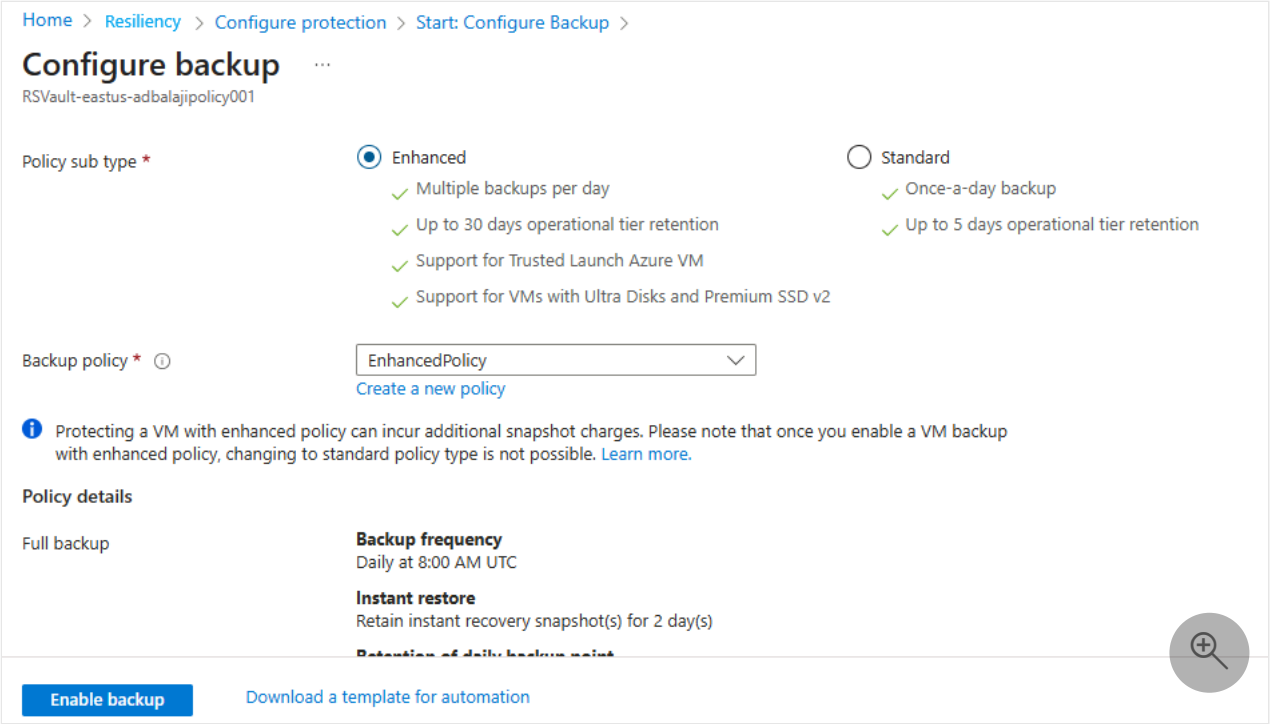


3. On the **Start: Configure Backup** pane, select **Azure Virtual machines** as the **Datasource type** and select the vault you have created. Then select **Continue**.



4. On the **Configure backup** pane, select the **Policy sub type** as **Enhanced**, **Standard**.

   - **Enhanced Backup policy**: This policy allows multiple daily backups, enabling hourly backups. To enable Azure Backup on Azure VMs in Azure Extended Zones, you can only use the Enhanced policy.
   - **Standard Backup policy**: This policy allows VM backup once a day. The daily backups are retained for 30 days. Instant recovery snapshots are retained for two days.

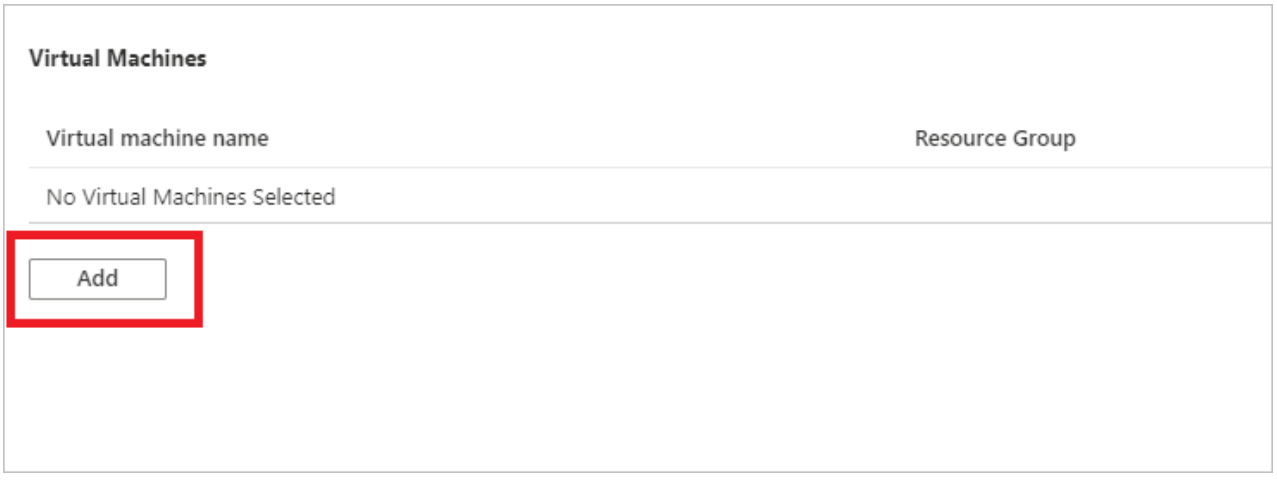


If you don't want to use the default policy, select **Create New**, and create a custom policy as described in the next procedure.

Select a VM to back up

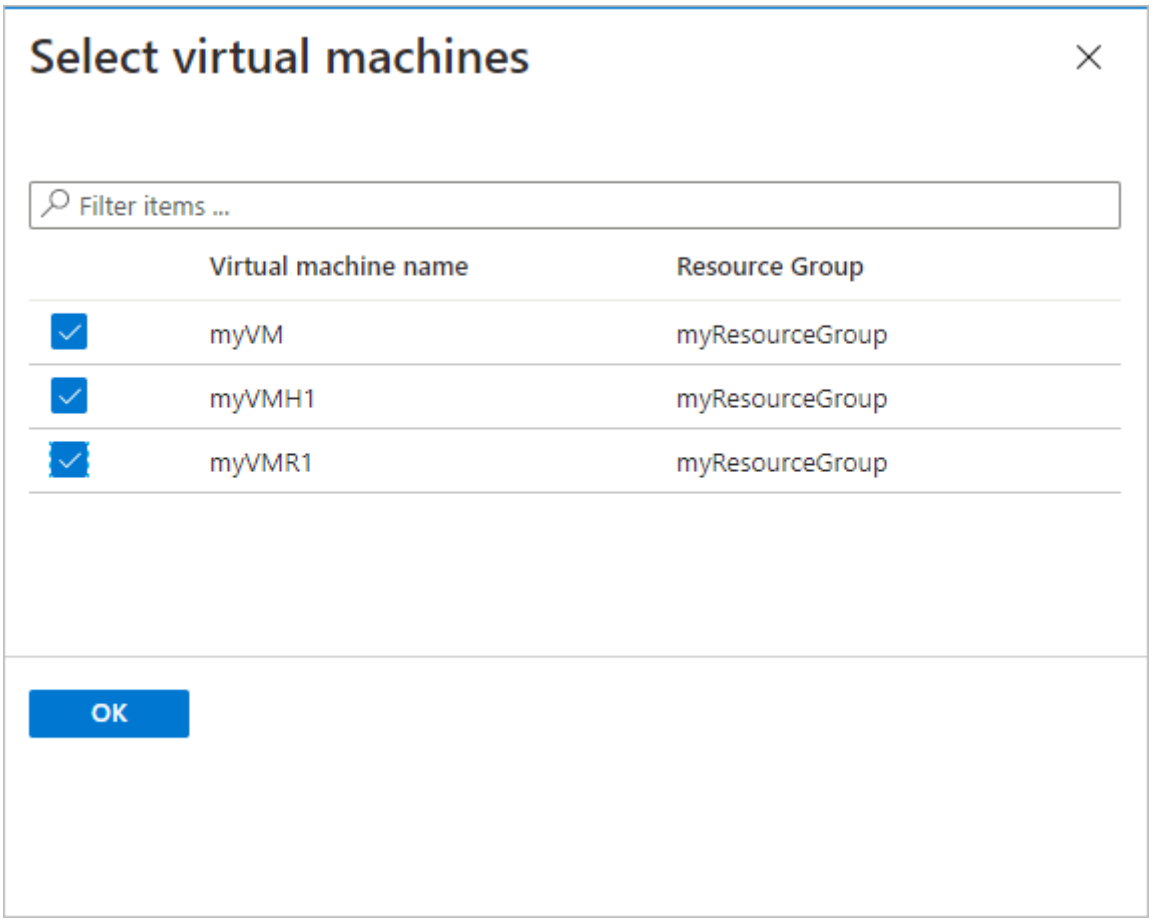
Create a scheduled daily backup to a Recovery Services vault.

1. Under **Virtual Machines**, select **Add**.



2. The **Select virtual machines** blade will open. Select the VMs you want to back up using the policy. Then select **OK**.

   - The selected VMs are validated.
   - You can only select VMs in the same region as the vault.
   - VMs can only be backed up in a single vault.



> ⓘ Note
>
> All the VMs in the same region and subscription as that of the vault are available to configure backup. When configuring backup, you can browse to the virtual machine name and its resource group, even though you don't have the required permission on those VMs. If your VM is in soft deleted state, then it won't be visible in this list. If you need to re-protect the VM, then you need to wait for the soft delete period to expire or undelete the VM from the soft deleted list. For more information, see the soft delete for VMs article.

Enable backup on a VM

A Recovery Services vault is a logical container that stores the backup data for each protected resource, such as Azure VMs. When the backup job for a protected resource runs, it creates a recovery point inside the Recovery Services vault. You can then use one of these recovery points to restore data to a given point in time.

To enable VM backup, in **Backup**, select **Enable backup**. This deploys the policy to the vault and to the VMs, and installs the backup extension on the VM agent running on the Azure VM.

After enabling backup:

- The Backup service installs the backup extension whether or not the VM is running.
- An initial backup will run in accordance with your backup schedule.
- When backups run, note that:
  - A VM that's running has the greatest chance for capturing an application-consistent recovery point.
  - However, even if the VM is turned off, it's backed up. Such a VM is known as an offline VM. In this case, the recovery point will be crash-consistent.
- Explicit outbound connectivity isn't required to allow backup of Azure VMs.

Create a custom policy

If you selected to create a new backup policy, fill in the policy settings.

1. In **Policy name**, specify a meaningful name.

2. In **Backup schedule**, specify when backups should be taken. You can take daily or weekly backups for Azure VMs.

3. In **Instant Restore**, specify how long you want to retain snapshots locally for instant restore.

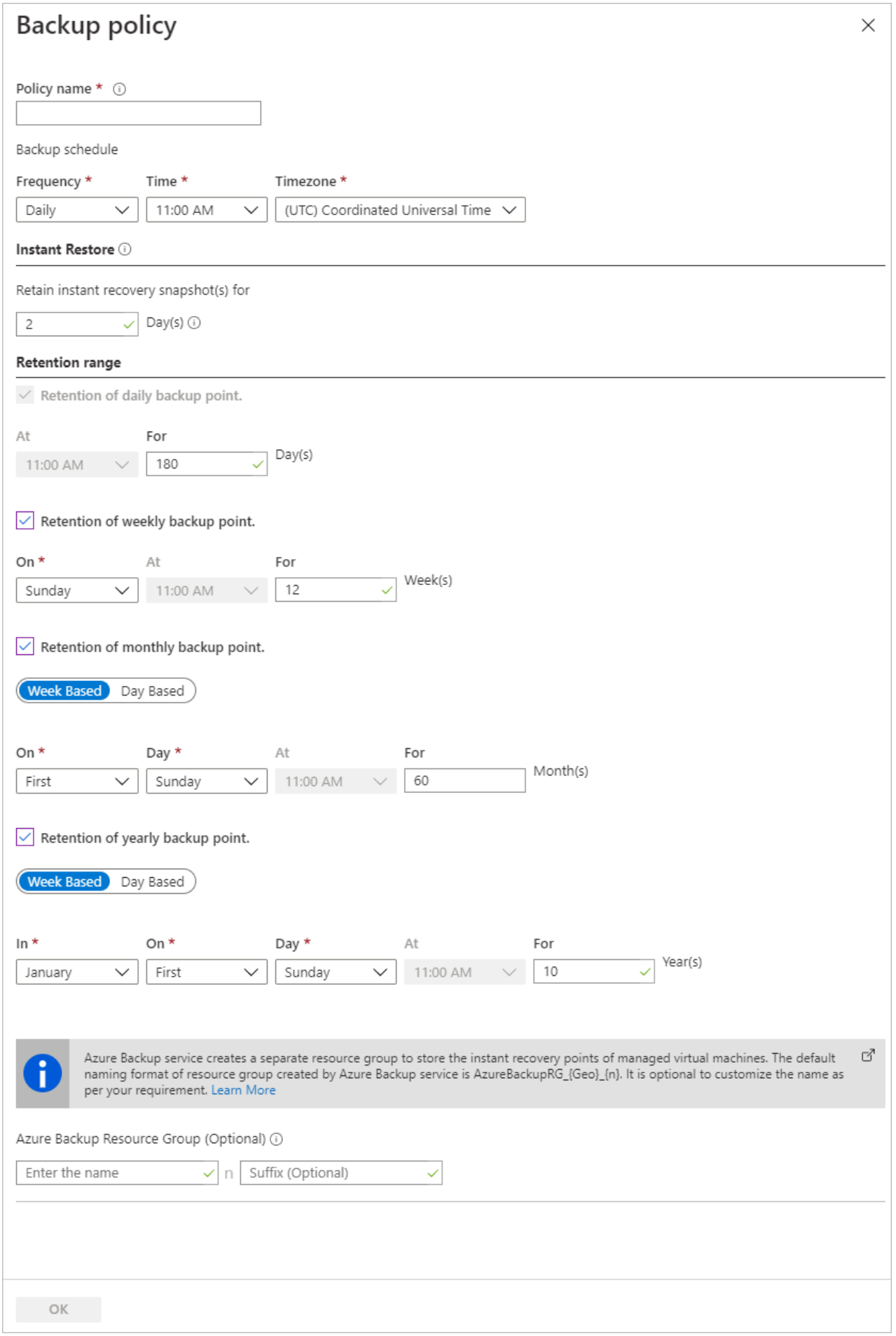
   - When you restore, backed up VM disks are copied from storage, across the network to the recovery storage location. With instant restore, you can leverage locally stored snapshots taken during a backup job, without waiting for backup data to be transferred to the vault.
   - You can retain snapshots for instant restore for between one to five days. The default value is two days.

4. In **Retention range**, specify how long you want to keep your daily or weekly backup points.

5. In **Retention of monthly backup point** and **Retention of yearly backup point**, specify whether you want to keep a monthly or yearly backup of your daily or weekly backups.

6. Select **OK** to save the policy.

   > ⓘ **Note**
   >
   > To store the restore point collection (RPC), the Backup service creates a separate resource group (RG). This RG is different than RG of the VM. Learn more.


Backup policy
Policy name *
Backup schedule
Frequency *
Time *
Timezone *
Daily
11:00 AM
(UTC) Coordinated Universal Time
Instant Restore
Retain instant recovery snapshot(s) for
2
Day(s)
Retention range
Retention of daily backup point.
At
For
11:00 AM
180
Day(s)
Retention of weekly backup point.
On *
At
For
Sunday
11:00 AM
12
Week(s)
Retention of monthly backup point.
Week Based
Day Based
On *
Day *
At
For
First
Sunday
11:00 AM
60
Month(s)
Retention of yearly backup point.
Week Based
Day Based
In *
On *
Day *
At
For
January
First
Sunday
11:00 AM
10
Year(s)
Azure Backup service creates a separate resource group to store the instant recovery points of managed virtual machines. The default naming format of resource group created by Azure Backup service is AzureBackupRG_{Geo}_{n}. It is optional to customize the name as per your requirement. Learn More
Azure Backup Resource Group (Optional)
Enter the name
n
Suffix (Optional)
OK

Note

Azure Backup doesn't support automatic clock adjustment for daylight-saving changes for Azure VM backups. As time changes occur, modify backup policies manually as required.

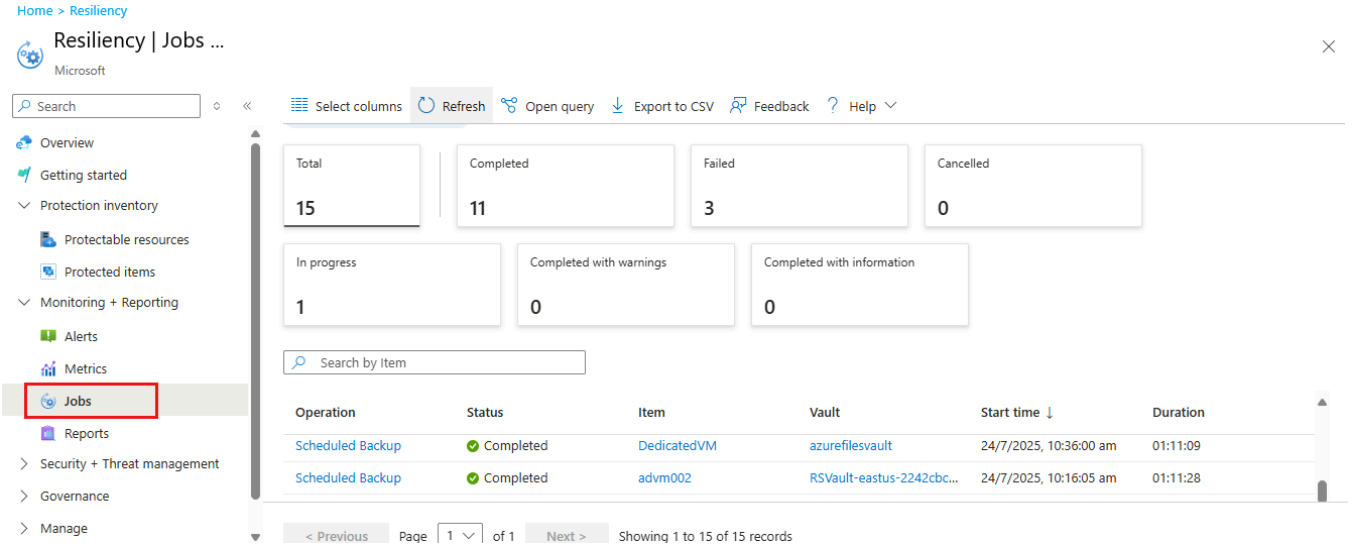
Start a backup job

The initial backup will run in accordance with the schedule, but you can run it immediately as follows:

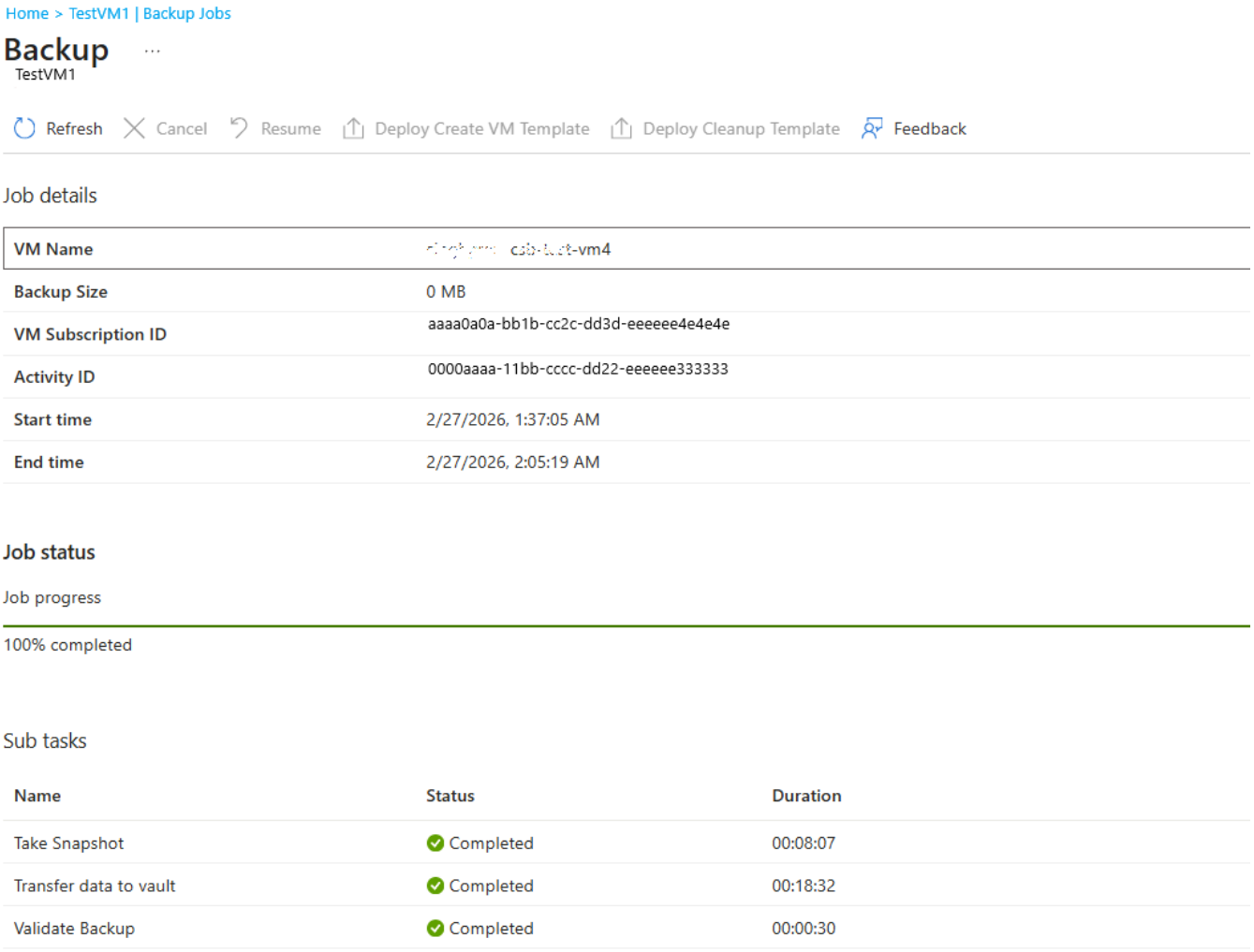
1. Go to **Resiliency** and then select **Protection Inventory** > **Protected items**.
2. On the **Protected items** pane, filter **Datasource type** by Virtual machines, and then select the **more icon** > **Details** corresponding to the VM instance you want to back up.
3. On the selected VM instance pane, right-click the relevant row or select the more icon (...), and then select **Backup Now**.
4. On the **Backup now** pane, use the calendar control to select the last day that the recovery point should be retained. Then select **OK**.

Monitor the backup job

Backup job details show **Snapshot** and **Transfer data to vault** subtasks. **Snapshot** ensures availability of a recovery point for Instant Restore. **Transfer data to vault** creates a recovery point in the vault for long-term retention and starts after **Snapshot** completes.



There are **Sub Tasks** running at the backend, one for front-end backup job that can be checked from the **Backup Job** details blade as given below:



For information on VM backup job phases and how to interpret the progress bar, see Verify the backup job status.

Depending on disk size and churn, **Transfer data to vault** can take time to complete. See the related article for job-status combinations, parallel backup behavior, and recovery-point guidance.

Optional steps

Install the VM agent

Azure Backup backs up Azure VMs by installing an extension to the Azure VM agent running on the machine. If your VM was created from an Azure Marketplace image, the agent is installed and running. If you create a custom VM, or you migrate an on-premises machine, you might need to install the agent manually, as summarized in the table.

Expand table

VM	Details
Windows	1. Download and install the agent MSI file. 2. Install with admin permissions on the machine. 3. Verify the installation. In *C:\WindowsAzure\Packages* on the VM, right-click **WaAppAgent.exe** > **Properties**. On the **Details** tab, **Product Version** should be 2.6.1198.718 or higher. If you're updating the agent, make sure that no backup operations are running, and reinstall the agent.
Linux	Install by using an RPM or a DEB package from your distribution's package repository. This is the preferred method for installing and upgrading the Azure Linux agent. All the endorsed distribution providers integrate the Azure Linux agent package into their images and repositories. The agent is available on GitHub, but we don't recommend installing from there. If you're updating the agent, make sure no backup operations are running, and update the binaries.

Clean up deployment

When no longer needed, you can disable protection on the VM, remove the restore points and Recovery Services vault, then delete the resource group and associated VM resources

Learn how to stop protection and delete VM backups.

Next steps

In this quickstart, you created a Recovery Services vault, enabled protection on a VM, and created the initial recovery point. To learn more about Azure Backup and Recovery Services, continue to the tutorials.

- Back up multiple Azure VMs
- Restore an Azure VM using REST API

Last updated on 03/02/2026

Back up a virtual machine in Azure with an ARM template

Azure Backup backs up on-premises machines and apps, and Azure VMs. This article shows you how to back up an Azure VM with an Azure Resource Manager template (ARM template) and Azure PowerShell. This quickstart focuses on the process of deploying an ARM template to create a Recovery Services vault. For more information on developing ARM templates, see the Azure Resource Manager documentation and the template reference.

An Azure Resource Manager template is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. You describe your intended deployment without writing the sequence of programming commands to create the deployment.

A Recovery Services vault is a logical container that stores backup data for protected resources, such as Azure VMs. When a backup job runs, it creates a recovery point inside the Recovery Services vault. You can then use one of these recovery points to restore data to a given point in time. Alternatively, you can back up a VM using Azure PowerShell, the Azure CLI, or in the Azure portal.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template will open in the Azure portal.



Review the template

The template used in this quickstart is from Azure quickstart Templates. This template allows you to deploy simple Windows VM and Recovery Services vault configured with the *DefaultPolicy* for *Protection*.

JSON

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-
01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "metadata": {
    "_generator": {
      "name": "bicep",
      "version": "0.5.6.12127",
      "templateHash": "12431143174203400392"
    }
  },
```

```
  "parameters": {
    "projectName": {
      "type": "string",
      "maxLength": 8,
      "metadata": {
        "description": "Specifies a name for generating resource names."
      }
    },
    "location": {
      "type": "string",
      "defaultValue": "[resourceGroup().location]",
      "metadata": {
        "description": "Specifies the location for all resources."
      }
    },
    "adminUsername": {
      "type": "string",
      "metadata": {
        "description": "Specifies the administrator username for the Virtual
Machine."
      }
    },
    "adminPassword": {
      "type": "secureString",
      "metadata": {
        "description": "Specifies the administrator password for the Virtual
Machine."
      }
    },
    "dnsLabelPrefix": {
      "type": "string",
      "metadata": {
        "description": "Specifies the unique DNS Name for the Public IP used to
access the Virtual Machine."
      }
    },
    "vmSize": {
      "type": "string",
      "defaultValue": "Standard_A2",
      "metadata": {
        "description": "Virtual machine size."
      }
    },
    "windowsOSVersion": {
      "type": "string",
      "defaultValue": "2016-Datacenter",
      "allowedValues": [
        "2008-R2-SP1",
        "2012-Datacenter",
        "2012-R2-Datacenter",
        "2016-Nano-Server",
        "2016-Datacenter-with-Containers",
        "2016-Datacenter",
        "2019-Datacenter",
        "2019-Datacenter-Core",
```

```
        "2019-Datacenter-Core-smalldisk",
        "2019-Datacenter-Core-with-Containers",
        "2019-Datacenter-Core-with-Containers-smalldisk",
        "2019-Datacenter-smalldisk",
        "2019-Datacenter-with-Containers",
        "2019-Datacenter-with-Containers-smalldisk"
      ],
      "metadata": {
        "description": "Specifies the Windows version for the VM. This will pick a
fully patched image of this given Windows version."
      }
    }
  },
  "variables": {
    "storageAccountName": "[format('{0}store', parameters('projectName'))]",
    "networkInterfaceName": "[format('{0}-nic', parameters('projectName'))]",
    "vNetAddressPrefix": "10.0.0.0/16",
    "vNetSubnetName": "default",
    "vNetSubnetAddressPrefix": "10.0.0.0/24",
    "publicIPAddressName": "[format('{0}-ip', parameters('projectName'))]",
    "vmName": "[format('{0}-vm', parameters('projectName'))]",
    "vNetName": "[format('{0}-vnet', parameters('projectName'))]",
    "vaultName": "[format('{0}-vault', parameters('projectName'))]",
    "backupFabric": "Azure",
    "backupPolicyName": "DefaultPolicy",
    "protectionContainer": "[format('iaasvmcontainer;iaasvmcontainerv2;{0};{1}',
resourceGroup().name, variables('vmName'))]",
    "protectedItem": "[format('vm;iaasvmcontainerv2;{0};{1}', resourceGroup().name,
variables('vmName'))]",
    "networkSecurityGroupName": "default-NSG"
  },
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2021-08-01",
      "name": "[variables('storageAccountName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "Standard_LRS"
      },
      "kind": "Storage",
      "properties": {}
    },
    {
      "type": "Microsoft.Network/publicIPAddresses",
      "apiVersion": "2021-05-01",
      "name": "[variables('publicIPAddressName')]",
      "location": "[parameters('location')]",
      "properties": {
        "publicIPAllocationMethod": "Dynamic",
        "dnsSettings": {
          "domainNameLabel": "[parameters('dnsLabelPrefix')]"
        }
      }
    },
```

```
    {
      "type": "Microsoft.Network/networkSecurityGroups",
      "apiVersion": "2021-05-01",
      "name": "[variables('networkSecurityGroupName')]",
      "location": "[parameters('location')]",
      "properties": {
        "securityRules": [
          {
            "name": "default-allow-3389",
            "properties": {
              "priority": 1000,
              "access": "Allow",
              "direction": "Inbound",
              "destinationPortRange": "3389",
              "protocol": "Tcp",
              "sourceAddressPrefix": "*",
              "sourcePortRange": "*",
              "destinationAddressPrefix": "*"
            }
          }
        ]
      }
    },
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2021-05-01",
      "name": "[variables('vNetName')]",
      "location": "[parameters('location')]",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
            "[variables('vNetAddressPrefix')]"
          ]
        },
        "subnets": [
          {
            "name": "[variables('vNetSubnetName')]",
            "properties": {
              "addressPrefix": "[variables('vNetSubnetAddressPrefix')]",
              "networkSecurityGroup": {
                "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
              }
            }
          }
        ]
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('networkSecurityGroupName'))]"
      ]
    },
    {
      "type": "Microsoft.Network/networkInterfaces",
      "apiVersion": "2021-05-01",
```

```
      "name": "[variables('networkInterfaceName')]",
      "location": "[parameters('location')]",
      "properties": {
        "ipConfigurations": [
          {
            "name": "ipconfig1",
            "properties": {
              "privateIPAllocationMethod": "Dynamic",
              "publicIPAddress": {
                "id": "[resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName'))]"
              },
              "subnet": {
                "id": "[format('{0}/subnets/{1}',
resourceId('Microsoft.Network/virtualNetworks', variables('vNetName')),
variables('vNetSubnetName'))]"
              }
            }
          }
        ]
      },
      "dependsOn": [
        "[resourceId('Microsoft.Network/publicIPAddresses',
variables('publicIPAddressName'))]",
        "[resourceId('Microsoft.Network/virtualNetworks', variables('vNetName'))]"
      ]
    },
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2021-11-01",
      "name": "[variables('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "hardwareProfile": {
          "vmSize": "[parameters('vmSize')]"
        },
        "osProfile": {
          "computerName": "[variables('vmName')]",
          "adminUsername": "[parameters('adminUsername')]",
          "adminPassword": "[parameters('adminPassword')]"
        },
        "storageProfile": {
          "imageReference": {
            "publisher": "MicrosoftWindowsServer",
            "offer": "WindowsServer",
            "sku": "[parameters('windowsOSVersion')]",
            "version": "latest"
          },
          "osDisk": {
            "createOption": "FromImage"
          },
          "dataDisks": [
            {
              "diskSizeGB": 1023,
              "lun": 0,
```

```
            "createOption": "Empty"
          }
        ]
      },
      "networkProfile": {
        "networkInterfaces": [
          {
            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]"
          }
        ]
      },
      "diagnosticsProfile": {
        "bootDiagnostics": {
          "enabled": true,
          "storageUri": "
[reference(resourceId('Microsoft.Storage/storageAccounts',
variables('storageAccountName'))).primaryEndpoints.blob]"
        }
      }
    },
    "dependsOn": [
      "[resourceId('Microsoft.Network/networkInterfaces',
variables('networkInterfaceName'))]",
      "[resourceId('Microsoft.Storage/storageAccounts',
variables('storageAccountName'))]"
    ]
  },
  {
    "type": "Microsoft.RecoveryServices/vaults",
    "apiVersion": "2022-01-01",
    "name": "[variables('vaultName')]",
    "location": "[parameters('location')]",
    "sku": {
      "name": "RS0",
      "tier": "Standard"
    },
    "properties": {}
  },
  {
    "type":
"Microsoft.RecoveryServices/vaults/backupFabrics/protectionContainers/protectedItems
",
    "apiVersion": "2022-01-01",
    "name": "[format('{0}/{1}/{2}/{3}', variables('vaultName'),
variables('backupFabric'), variables('protectionContainer'),
variables('protectedItem'))]",
    "properties": {
      "protectedItemType": "Microsoft.Compute/virtualMachines",
      "policyId": "[format('{0}/backupPolicies/{1}',
resourceId('Microsoft.RecoveryServices/vaults', variables('vaultName')),
variables('backupPolicyName'))]",
      "sourceResourceId": "[resourceId('Microsoft.Compute/virtualMachines',
variables('vmName'))]"
    },
```

```
      "dependsOn": [
        "[resourceId('Microsoft.RecoveryServices/vaults', variables('vaultName'))]",
        "[resourceId('Microsoft.Compute/virtualMachines', variables('vmName'))]"
      ]
    }
  ]
}
```

The resources defined in the template are:

- Microsoft.Storage/storageAccounts
- Microsoft.Network/publicIPAddresses
- Microsoft.Network/networkSecurityGroups
- Microsoft.Network/virtualNetworks
- Microsoft.Network/networkInterfaces
- Microsoft.Compute/virtualMachines
- Microsoft.RecoveryServices/vaults
- Microsoft.RecoveryServices/vaults/backupFabrics/protectionContainers/protectedItems

Deploy the template

To deploy the template, select **Try it** to open the Azure Cloud Shell, and then paste the following PowerShell script into the shell window. To paste the code, right-click the shell window and then select **Paste**.

Azure PowerShell

```
$projectName = Read-Host -Prompt "Enter a project name (limited to eight characters) that is used to generate Azure resource names"
$location = Read-Host -Prompt "Enter the location (for example, centralus)"
$adminUsername = Read-Host -Prompt "Enter the administrator username for the virtual machine"
$adminPassword = Read-Host -Prompt "Enter the administrator password for the virtual machine" -AsSecureString
$dnsPrefix = Read-Host -Prompt "Enter the unique DNS Name for the Public IP used to access the virtual machine"

$resourceGroupName = "${projectName}rg"
$templateUri = "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/quickstarts/microsoft.recoveryservices/recovery-services-create-vm-and-configure-backup/azuredeploy.json"

New-AzResourceGroup -Name $resourceGroupName -Location $location
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName -TemplateUri $templateUri -projectName $projectName -adminUsername $adminUsername -adminPassword $adminPassword -dnsLabelPrefix $dnsPrefix
```

Azure PowerShell is used to deploy the ARM template in this quickstart. The Azure portal, Azure CLI, and REST API can also be used to deploy templates.

Validate the deployment

Start a backup job

The template creates a VM and enables backup on the VM. After you deploy the template, you need to start a backup job. For more information, see Start a backup job.

Monitor the backup job

To monitor the backup job, see Monitor the backup job.

Clean up resources

If you no longer need to back up the VM, you can clean it up.

- If you want to try out restoring the VM, skip the cleanup.
- If you used an existing VM, you can skip the final Remove-AzResourceGroup cmdlet to leave the resource group and VM in place.

Disable protection, remove the restore points and vault. Then delete the resource group and associated VM resources, as follows:

PowerShell

```
Disable-AzRecoveryServicesBackupProtection -Item $item -RemoveRecoveryPoints
$vault = Get-AzRecoveryServicesVault -Name "myRecoveryServicesVault"
Remove-AzRecoveryServicesVault -Vault $vault
Remove-AzResourceGroup -Name "myResourceGroup"
```

Next steps

In this quickstart, you created a Recovery Services vault, enabled protection on a VM, and created the initial recovery point.

- Learn how to back up VMs in the Azure portal.
- Learn how to quickly restore a VM
- Learn how to create ARM templates.
- Learn how to restore an Azure VM using REST API

Last updated on 12/19/2025

Tutorial: Back up multiple virtual machines by using the Azure portal

This tutorial describes how to back up multiple virtual machines (VMs) by using the Azure portal.

Azure stores backup data in a Recovery Services vault, which is accessible from the **Settings** menu of most services. This integration simplifies the backup process. However, managing each database or virtual machine separately can be tedious. To streamline backups for multiple virtual machines (for department or location), you can create a backup policy and apply it to the relevant machines.

In this tutorial, you:

- ✓ Create a Recovery Services vault.
- ✓ Set the vault to help protect virtual machines.
- ✓ Create a custom backup and retention policy.
- ✓ Assign the policy to help protect multiple virtual machines.
- ✓ Trigger an on-demand back up for virtual machines.

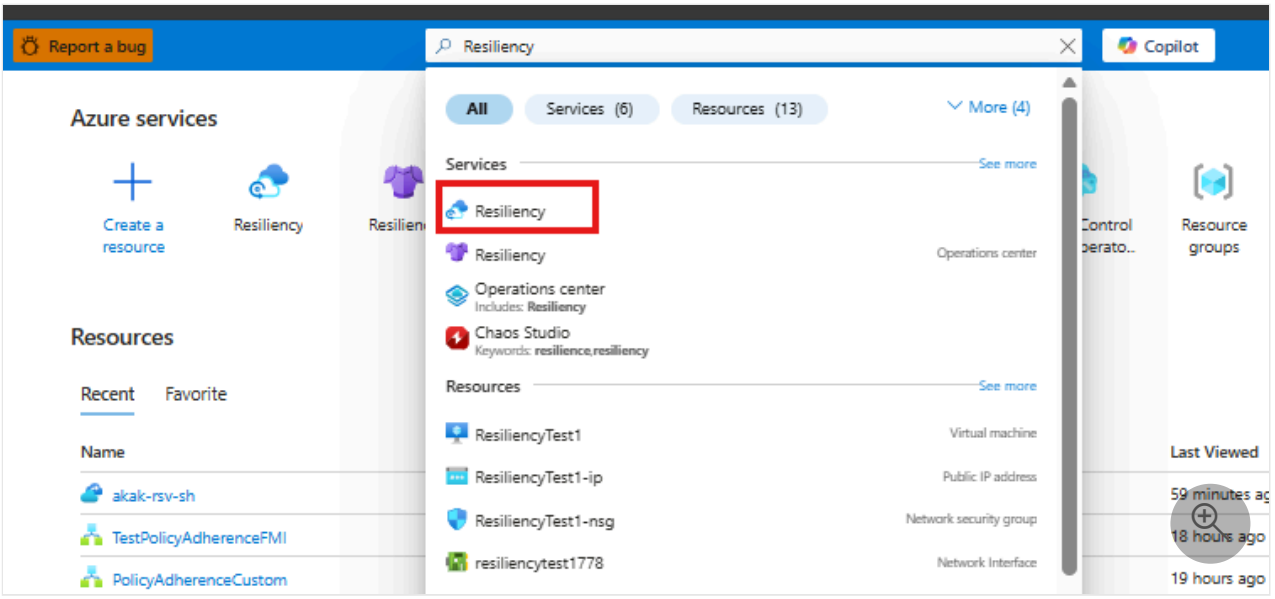
> **Note**
>
> The functionality that the following sections describe can also be accessed via Resiliency. Resiliency in Azure is a cloud-native unified experience for resiliency posture management that enables you to manage your protection estate across solutions and environments.

Create a Recovery Services vault

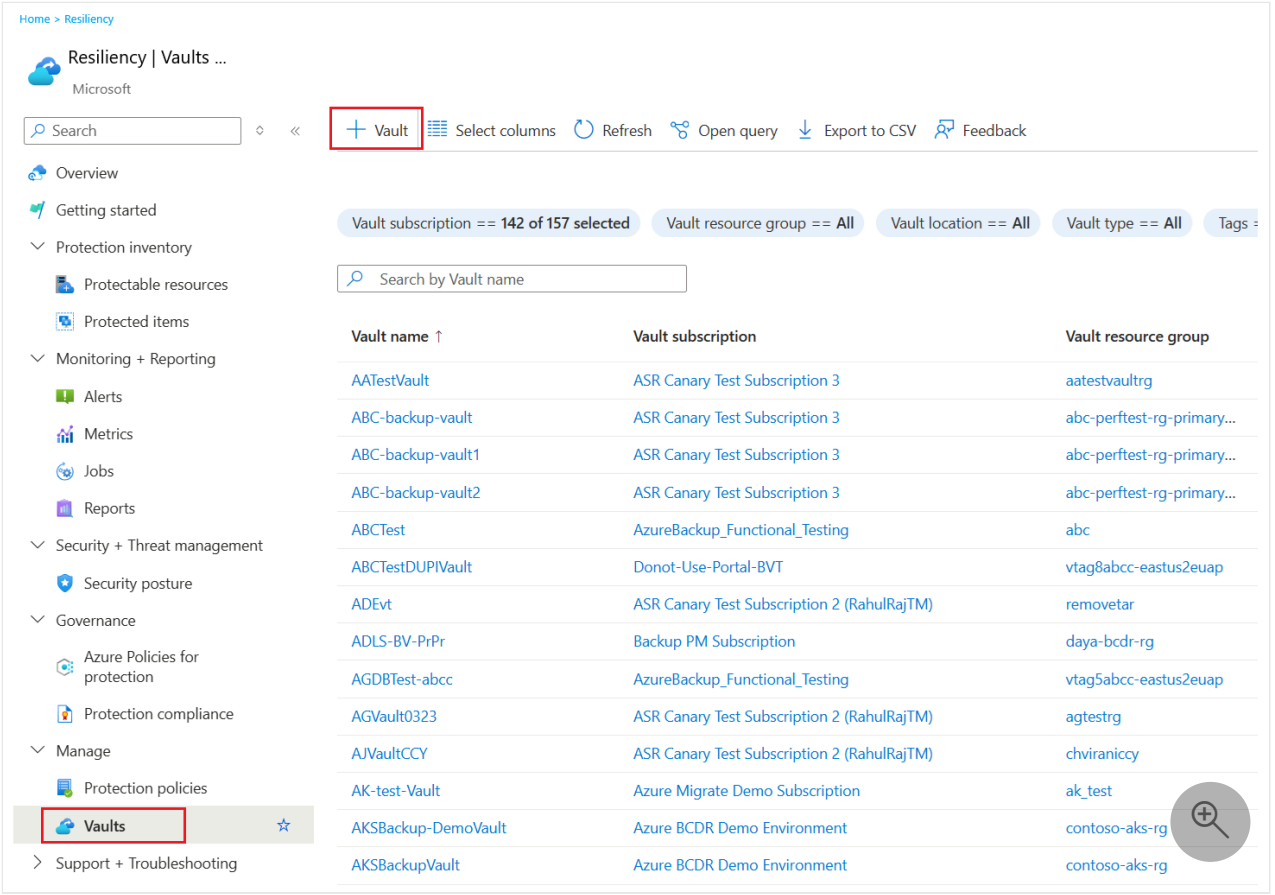
A Recovery Services vault is a management entity that stores recovery points that are created over time. It provides an interface to perform backup-related operations. These operations include taking on-demand backups, performing restores, and creating backup policies.

To create a Recovery Services vault:

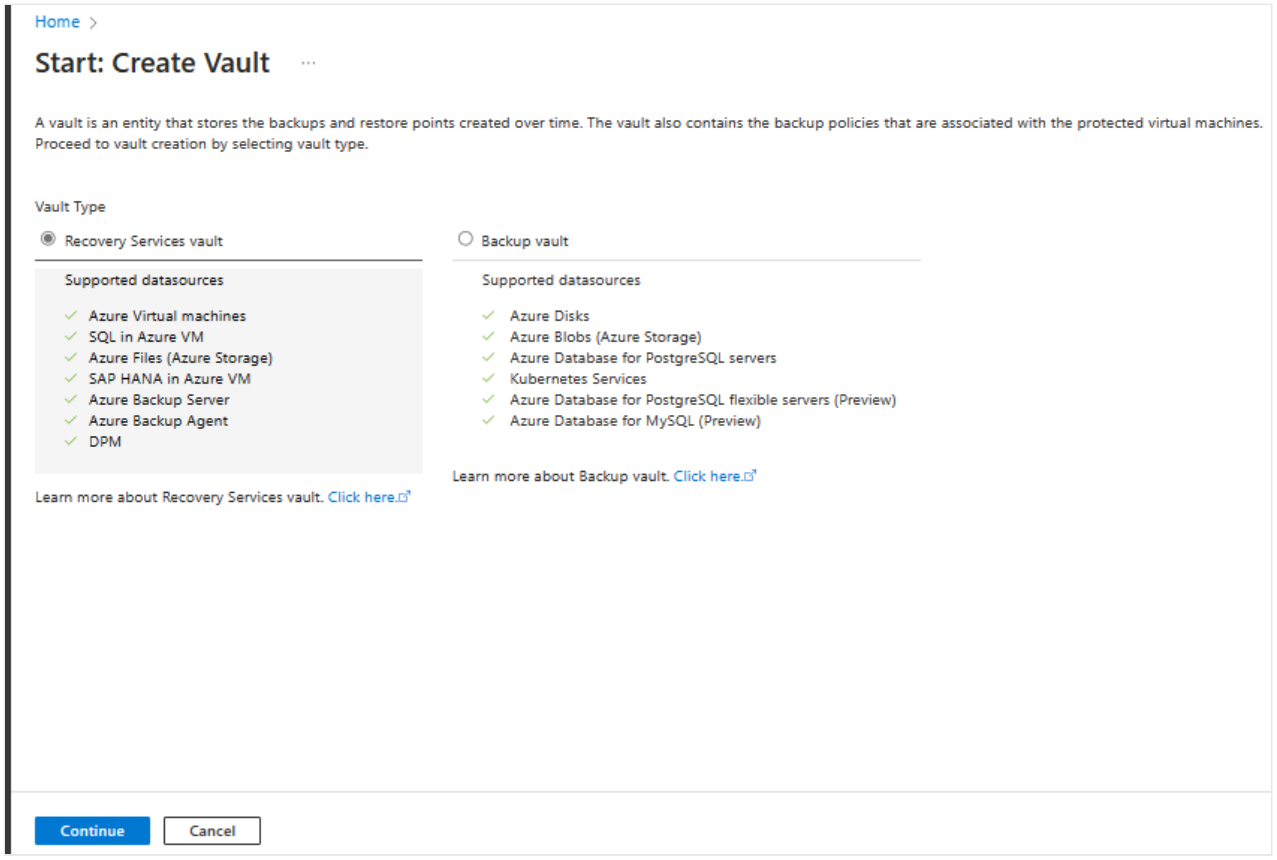
1. Sign in to the Azure portal.
2. Search for **Resiliency**, and then go to the **Resiliency** dashboard.



3. On the **Vault** pane, select **+ Vault**.



4. Select **Recovery Services vault** > **Continue**.



5. On the **Create Recovery Services vault** pane, enter the following values:

   - **Subscription**: Select the subscription to use. If you're a member of only one subscription, you see that name. If you're not sure which subscription to use, use the default subscription. Multiple choices appear only if your work or school account is associated with more than one Azure subscription.

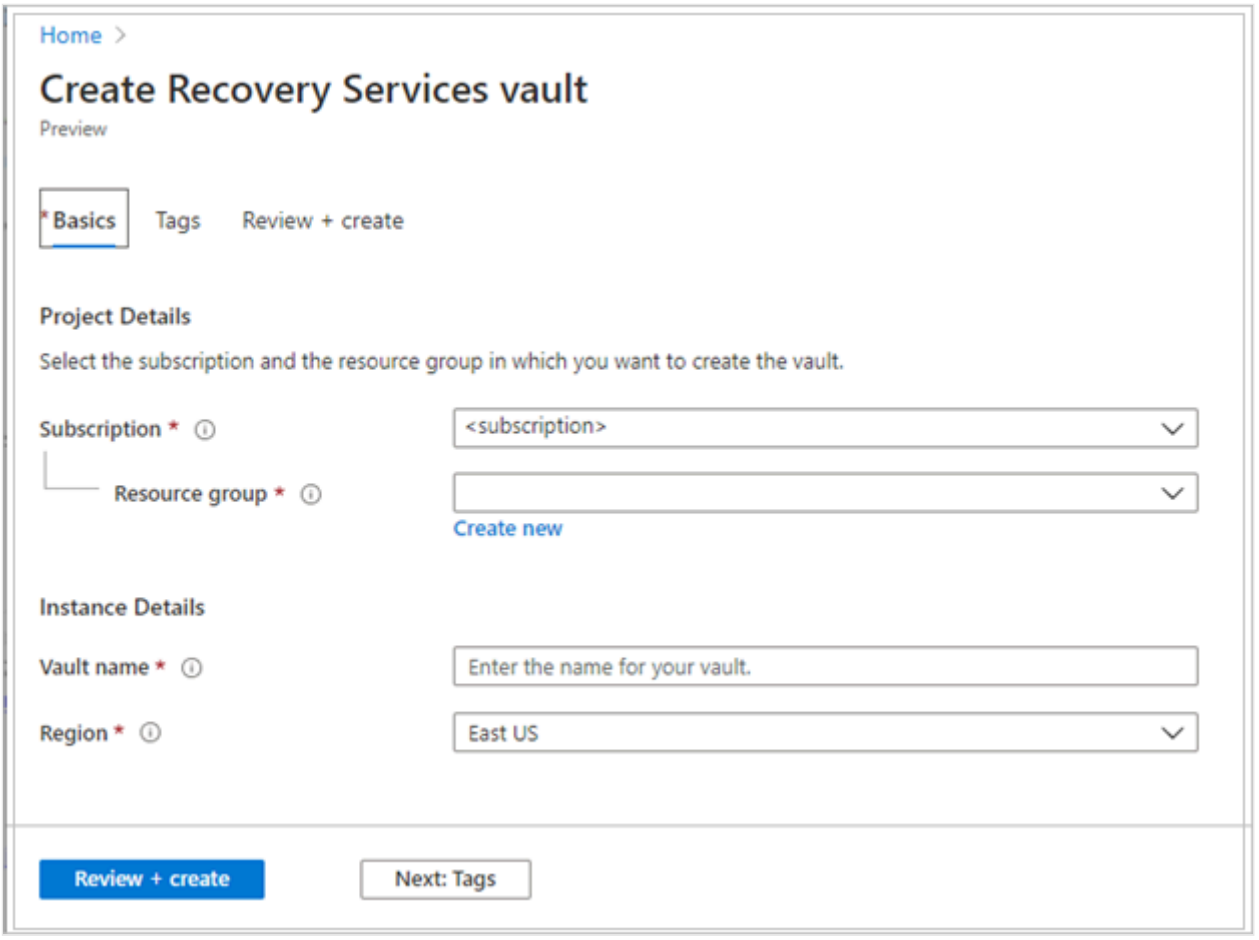
   - **Resource group**: Use an existing resource group or create a new one. To view a list of available resource groups in your subscription, select **Use existing**. Then select a resource in the dropdown list. To create a new resource group, select **Create new**, and then enter the name. For more information about resource groups, see Azure Resource Manager overview.

   - **Vault name**: Enter a friendly name to identify the vault. The name must be unique to the Azure subscription. Specify a name that has at least 2 but not more than 50 characters. The name must start with a letter and consist only of letters, numbers, and hyphens.

   - **Region**: Select the geographic region for the vault. For you to create a vault to help protect any data source, the vault *must* be in the same region as the data source.

     > **Important**
     >
     > If you're not sure of the location of your data source, close the window. Go to the list of your resources in the portal. If you have data sources in multiple

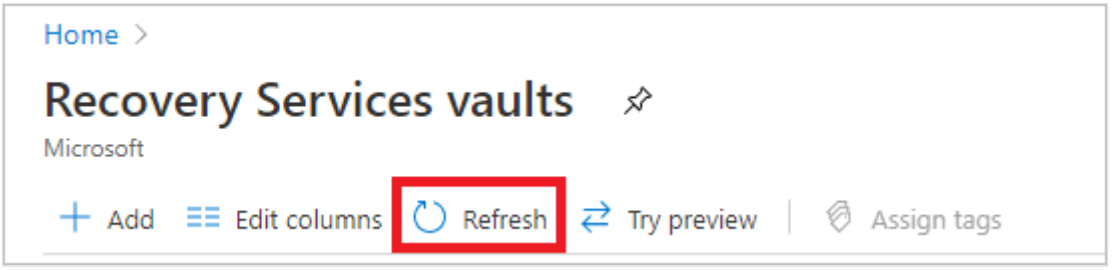
regions, create a Recovery Services vault for each region. Create the vault in the first location before you create a vault in another location. You don't need to specify storage accounts to store the backup data. The Recovery Services vault and Azure Backup handle that step automatically.



6. After you provide the values, select **Review + create**.

7. To finish creating the Recovery Services vault, select **Create**.

   It can take a while to create the Recovery Services vault. Monitor the status notifications in the **Notifications** area at the upper right. After the vault is created, it appears in the list of Recovery Services vaults. If the vault doesn't appear, select **Refresh**.



Azure Backup now supports immutable vaults that help you ensure that after recovery points are created, they can't be deleted before their expiry according to the backup policy. You can make the immutability irreversible to help protect your backup data from various threats,

including ransomware attacks and malicious actors. Learn more about Azure Backup immutable vaults.

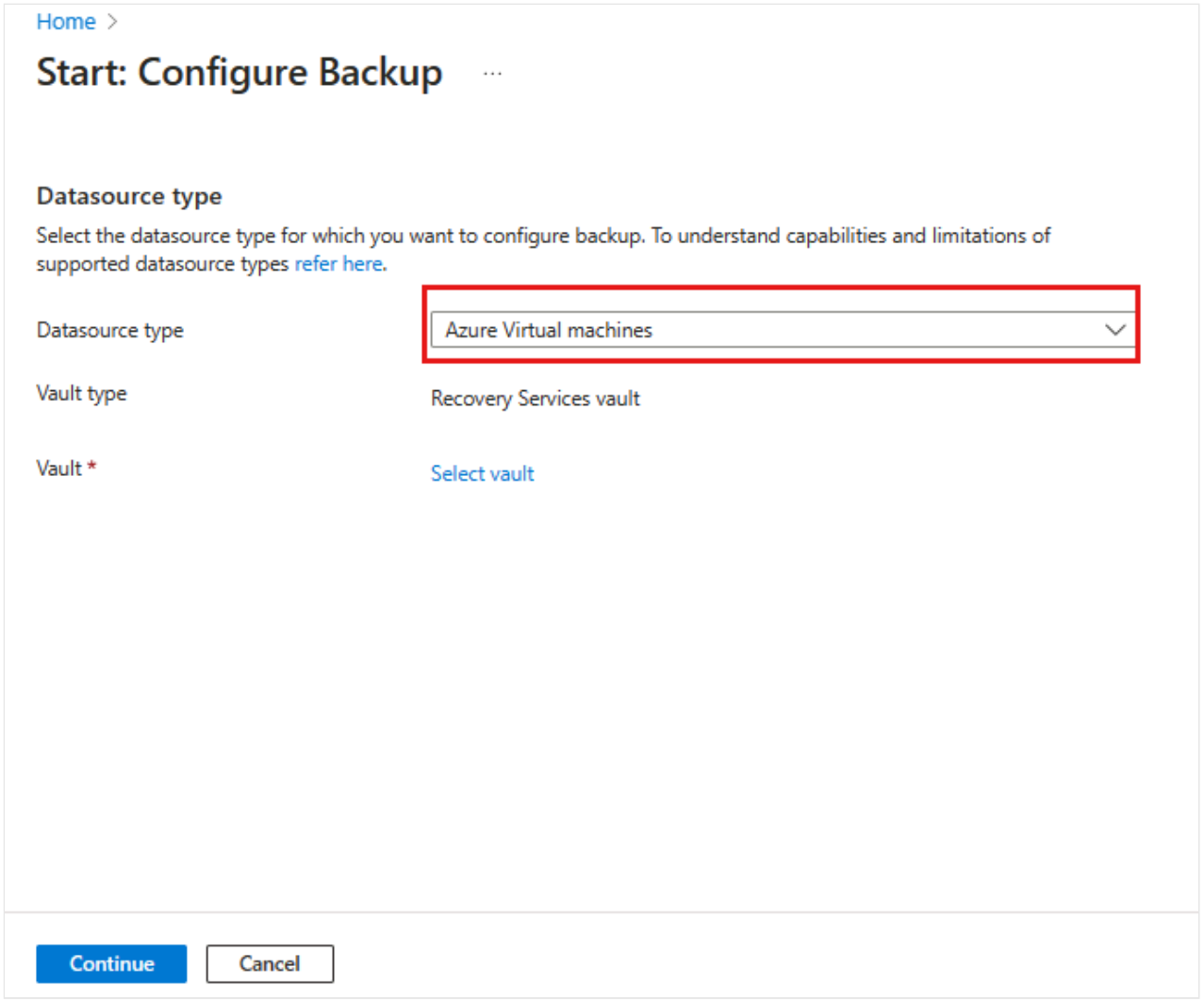
When you create a Recovery Services vault, the vault has geo-redundant storage by default. To provide data resiliency, geo-redundant storage replicates the data multiple times across two Azure regions.

Set a backup policy for VMs

After creation of the Recovery Services vault is complete, configure the vault for the data type backup and set the backup policy. The policy defines the schedule for recovery points and their retention period. In this tutorial, assume that you have a business (a sports complex with a hotel, stadium, and restaurants) that requires VM data protection.

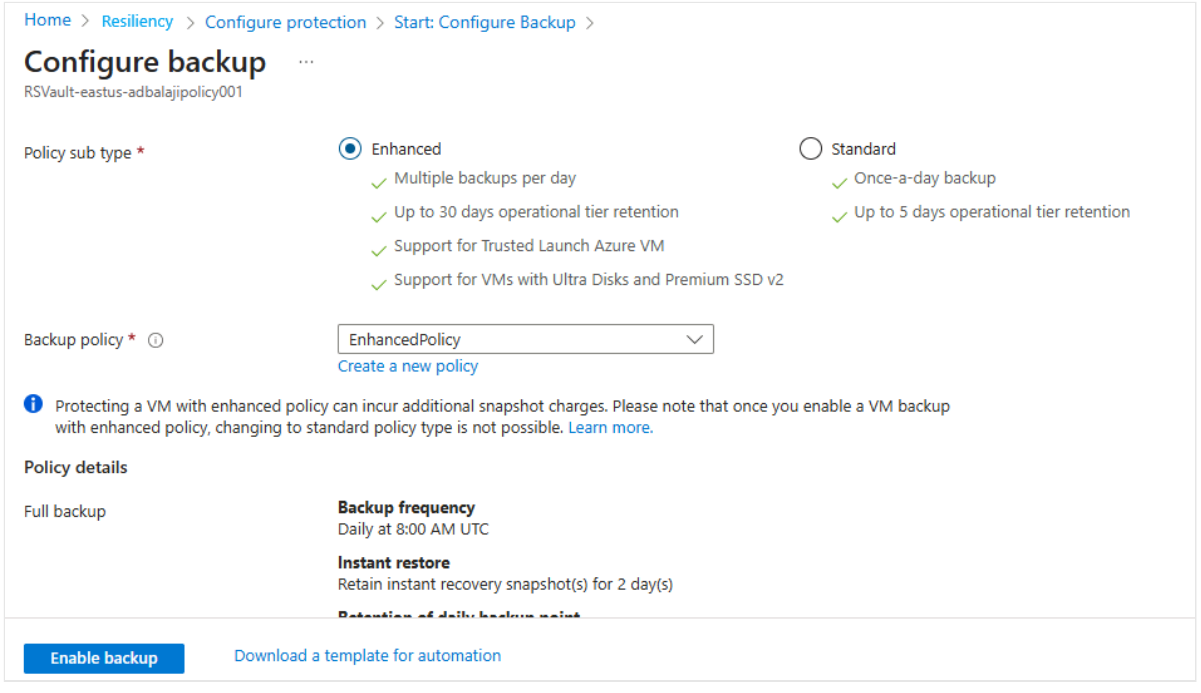
To set a backup policy for your Azure VMs, follow these steps:

1. Go to **Resiliency**. On the **Overview** tab, select **+ Configure protection**.
2. On the **Configure protection** pane, under **Where is your workload running?**, select **Azure**. Under **What do you want to back up?**, select **Virtual machine**. Then select **Continue**.
3. On the **Start: Configure Backup** pane, select **Azure Virtual machines** as the **Datasource type** value, and then select the vault that you created. Then select **Continue**.



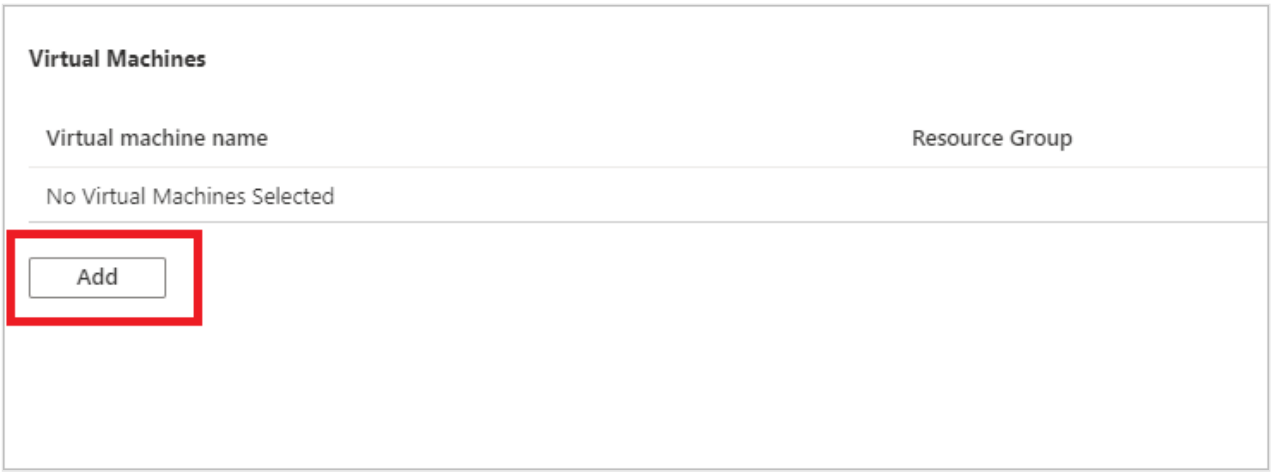
4. Assign a backup policy:

   - The default policy backs up the VM once a day. The daily backups are retained for 30 days. Instant recovery snapshots are retained for 2 days.



- If you don't want to use the default policy, select **Create a new policy**, and then create a custom policy.

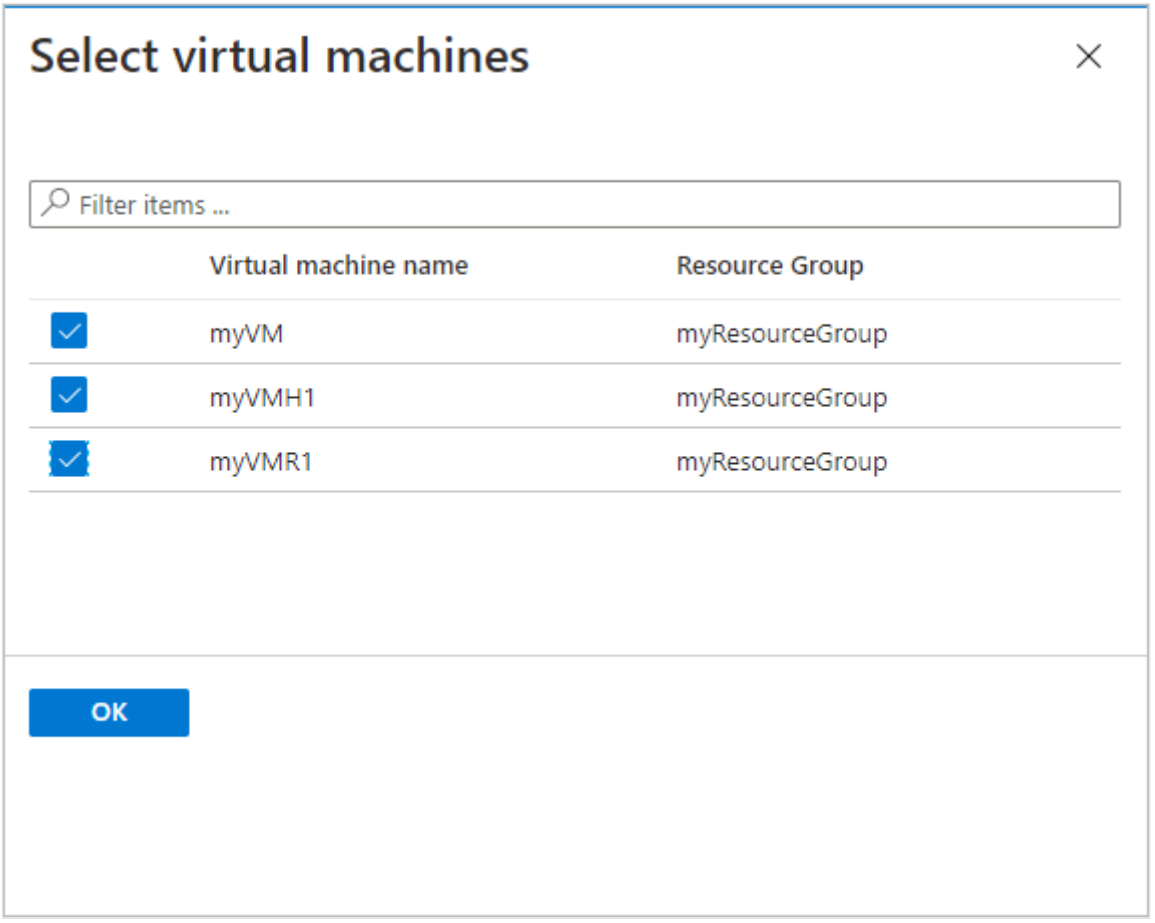
5. Under **Virtual Machines**, select **Add**.



6. The **Select virtual machines** pane opens. Select the VMs that you want to back up by using the policy. Then select **OK**.

   The following considerations apply:

   - The selected VMs are validated.
   - You can select only VMs in the same region as the vault.
   - VMs can be backed up only in a single vault.



> **Note**
>
> All the VMs in the same region and subscription as that of the vault are available for configuring a backup. When you configure backup, you can browse to each VM's name and its resource group, even though you don't have the required permission on those VMs.
>
> If your VM is in soft-deleted state, it isn't visible in this list. If you need to protect the VM again, wait for the soft-delete period to expire or restore the VM from the soft-deleted list. For more information, see the article about soft delete for VMs.

7. In **Backup**, select **Enable backup**. This selection deploys the policy to the vault and to the VMs. It also installs the backup extension on the VM agent that runs on the Azure VM.

After you enable a backup:

- The Azure Backup service installs the backup extension whether or not the VM is running.
- An initial backup runs in accordance with your backup schedule.

When backups run, note that:

- A VM that's running has the greatest chance for capturing an application-consistent recovery point.
- Even if a VM is offline, it's backed up. In this case, the recovery point is crash-consistent.
- Explicit outbound connectivity isn't required for backing up Azure VMs.

> **Note**
>
> You can also set an Enhanced policy to back up Azure VMs multiple times a day.

Run an initial backup

You enabled backups for the Recovery Services vaults, but you haven't created an initial backup. A best practice for disaster recovery is to trigger the initial backup so that your data is protected.

The initial backup runs in accordance with the schedule, but you can run it immediately as follows:

1. Go to **Resiliency** and select **Protected items**.

2. On the **Protected items** pane, for **Datasource type**, select **Azure Virtual machines**. Then search for the VM that you configured for backups.

3. Right-click the relevant row or select the more icon (...), and then select **Backup Now**.

4. In **Backup Now**, use the calendar control to select the last day that the recovery point should be retained. Then select **OK**.

5. Monitor the portal notifications. To monitor the job's progress, go to **Resiliency** > **Jobs** and filter the list for **In progress** jobs.

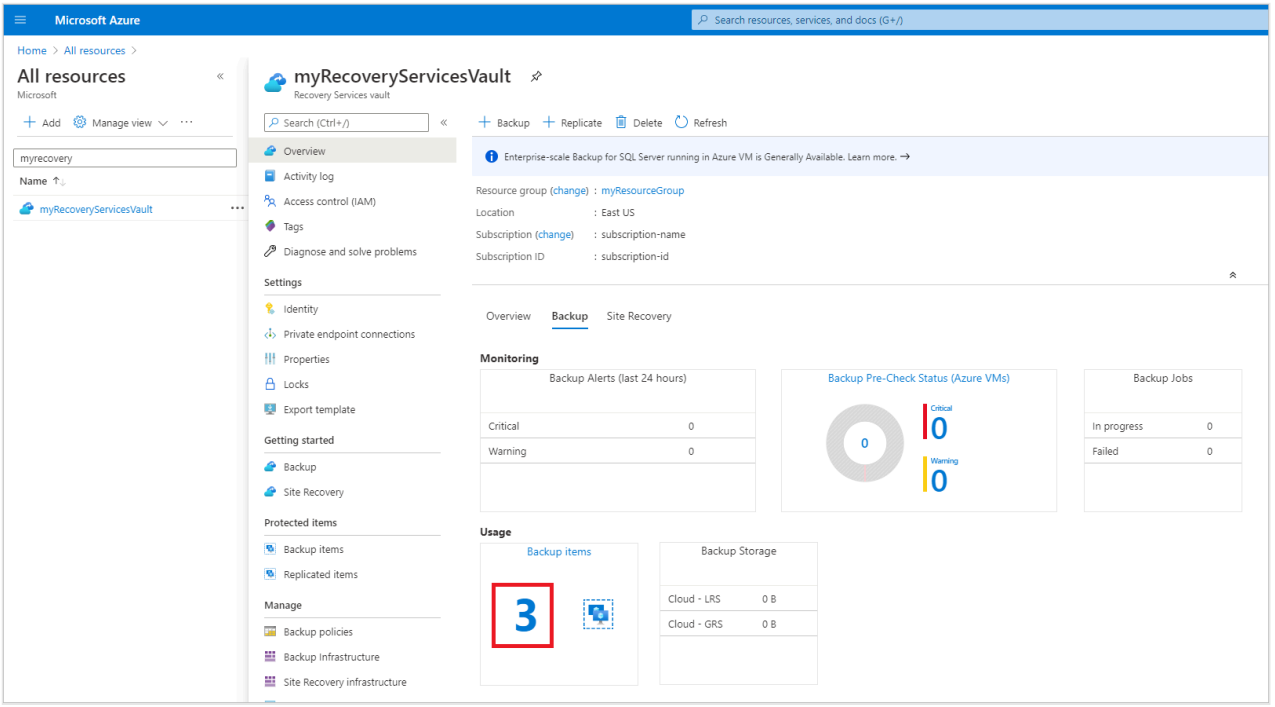
   Depending on the size of your VM, creating the initial backup might take a while.

Clean up resources

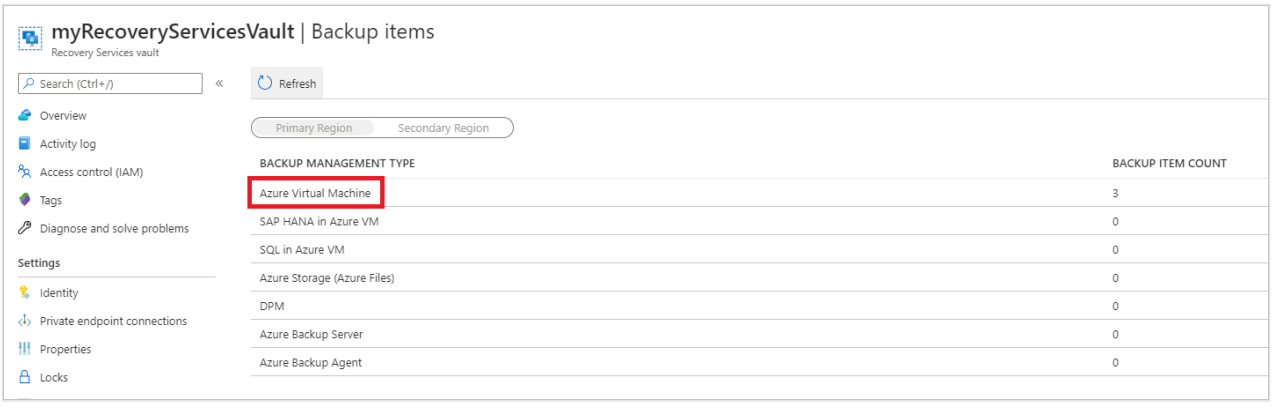
If you plan to work with subsequent tutorials, don't clean up the resources that you created in this tutorial.

If you don't plan to continue, delete all resources that you created in this tutorial. In the Azure portal, follow these steps:

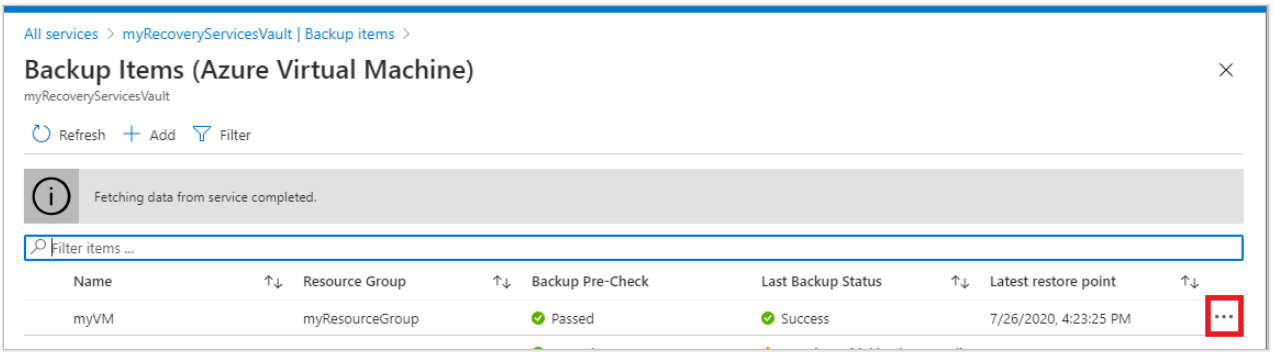
1. On the **myRecoveryServicesVault** dashboard, under **Backup items**. select **3**.



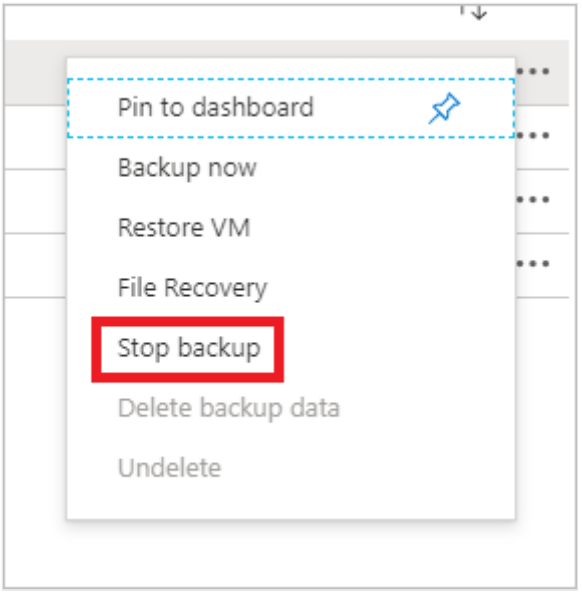
2. On the **Backup items** pane, select **Azure Virtual Machine** to open the list of virtual machines associated with the vault.



3. In the list of backup items, select the ellipsis for the **myVM** item.



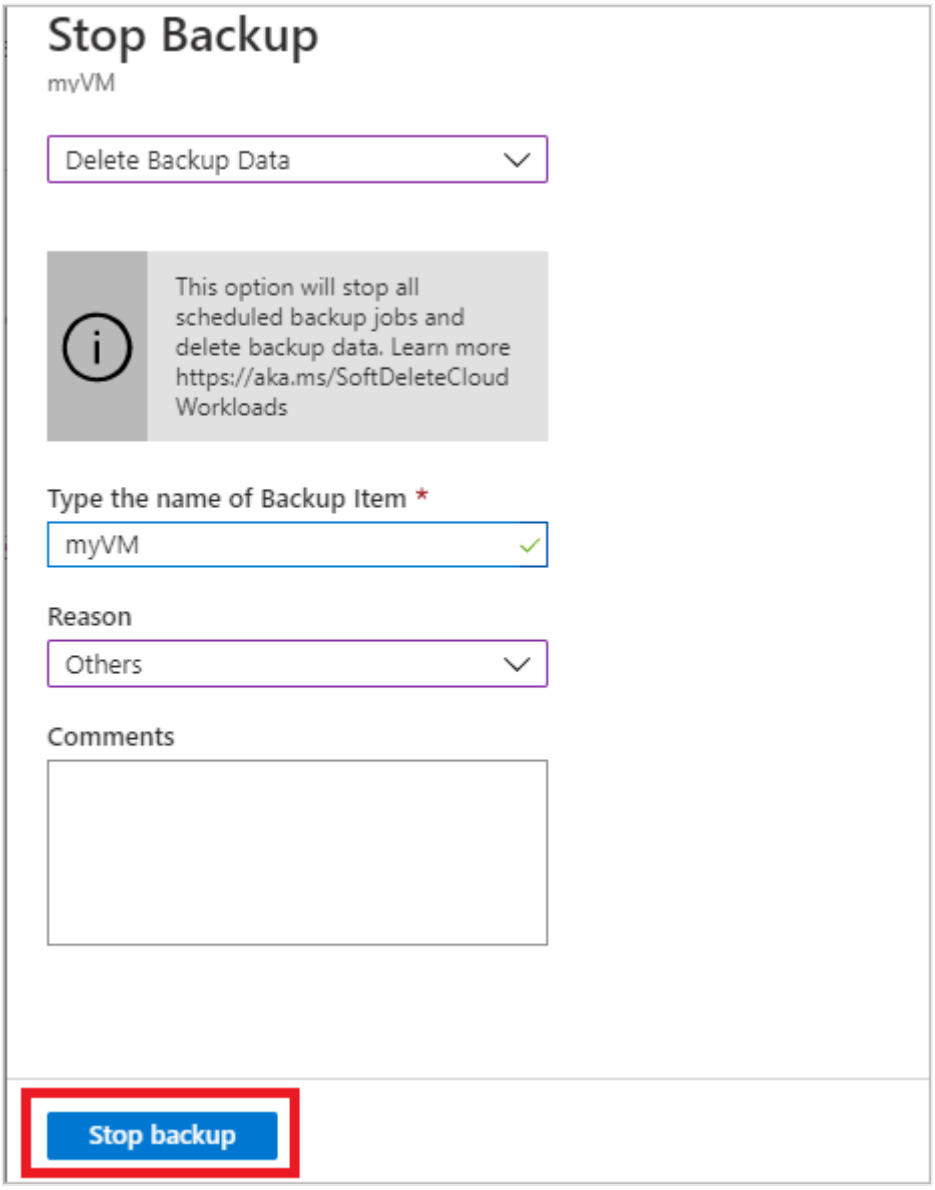
4. On the menu that opens, select **Stop backup**.



5. On the **Stop Backup** pane, in the uppermost box, select **Delete Backup Data**.

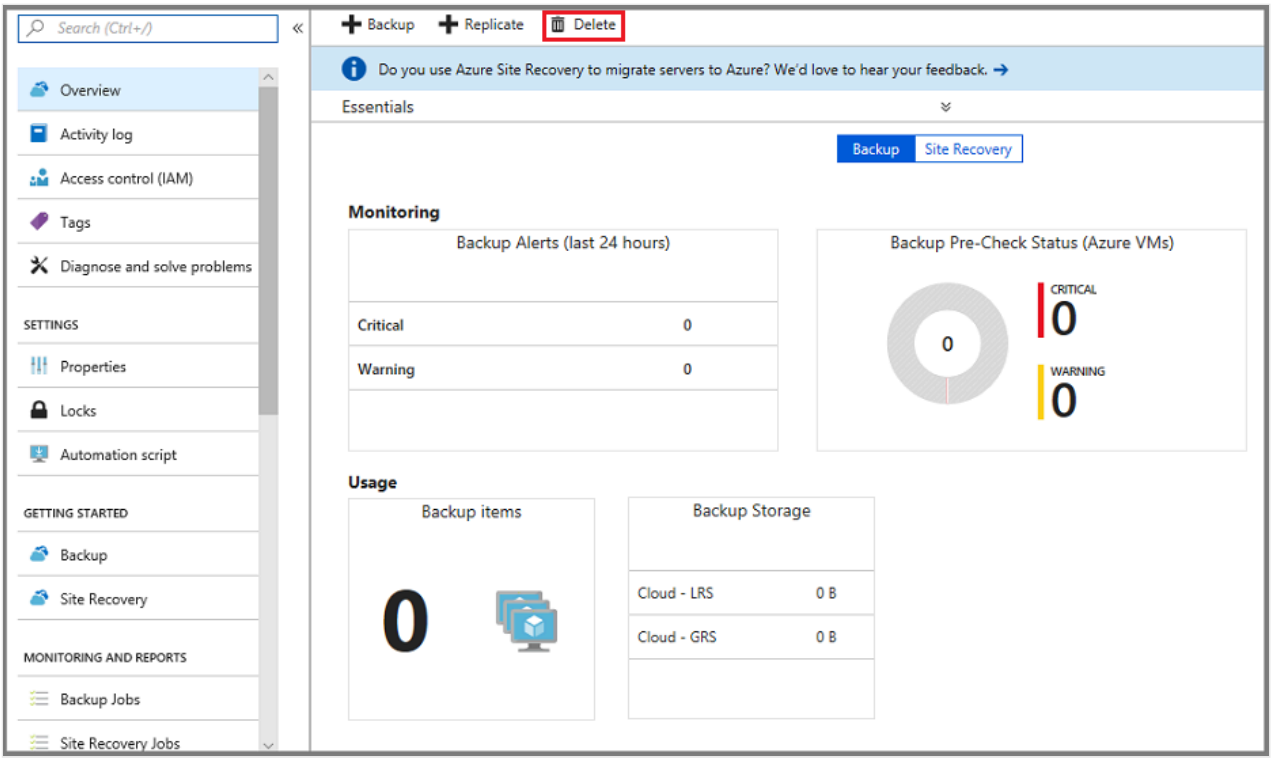
6. In the **Type the name of the Backup Item** box, enter **myVM**.

7. After the backup item is verified, a check mark appears. Select the **Stop backup** button to stop the policy and delete the restore points.



> ⓘ **Note**
>
> Deleted items stay in a soft-deleted state for 14 days. You can delete the vault after that period. For more information, see Delete an Azure Backup Recovery Services vault.

8. When no more items are in the vault, select **Delete**.



After you delete the vault, you return to the list of Recovery Services vaults.

Next step

Continue to the next tutorial to restore an Azure virtual machine from disk:

Restore a VM by using the Azure CLI

Last updated on 01/28/2026

How to restore Azure VM data in Azure portal

This article describes how to restore Azure VM data from the recovery points stored in Azure Backup Recovery Services vaults.

Restore options

Azure Backup provides several ways to restore a VM.

Expand table

Restore option	Details
Create a new VM	Quickly creates and gets a basic VM up and running from a restore point. You can specify a name for the VM and select the resource group and virtual network (VNet) in which it will be placed. The new VM must be created in the same region as the source VM. If a VM restore fails because an Azure VM SKU wasn't available in the specified region of Azure, or because of any other issues, Azure Backup still restores the disks in the specified resource group.
Restore disk	Restores a VM disk, which can then be used to create a new VM. Azure Backup provides a template to help you customize and create a VM. The restore job generates a template that you can download and use to specify custom VM settings, and create a VM. The disks are copied to the Resource Group you specify. Alternatively, you can attach the disk to an existing VM, or create a new VM using PowerShell. This option is useful if you want to customize the VM, add configuration settings that weren't there at the time of backup, or add settings that must be configured using the template or PowerShell.
Replace existing	You can restore a disk, and use it to replace a disk on the existing VM. The current VM must exist. If it's been deleted, this option can't be used. Azure Backup takes a snapshot of the existing VM before replacing the disk. The snapshot is copied to the vault and retained in accordance with the retention policy.

Restore option	Details
	When you choose a Vault-Standard recovery point, a VHD file with the content of the chosen recovery point is also created in the staging location you specify. Existing disks connected to the VM are replaced with the selected restore point. After the disk replacement operation is complete, the original disk is retained in the resource group. You can choose to manually delete the original disks if they aren't needed. Replace existing is supported for unencrypted managed VMs, including VMs created using custom images. It's unsupported for classic VMs, unmanaged VMs, and generalized VMs. If the restore point has more or less disks than the current VM, then the number of disks in the restore point will only reflect the VM configuration. Replace existing is also supported for VMs with linked resources, like user-assigned managed-identity or Key Vault.
Cross Region (secondary region)	Cross Region restore can be used to restore Azure VMs in the secondary region, which is an Azure paired region. You can restore all the Azure VMs for the selected recovery point if the backup is done in the secondary region. During the backup, snapshots aren't replicated to the secondary region. Only the data stored in the vault is replicated. So secondary region restores are only vault tier restores. The restore time for the secondary region will be almost the same as the vault tier restore time for the primary region. This feature is available for the options below: - Create a VM - Restore Disks If the source machine has more than 16 disks, VHDs won't be created in Cross Region Restore. We don't currently support the Replace existing disks option. Permissions The restore operation on secondary region can be performed by Backup Admins and App admins.
Cross Subscription Restore	Allows you to restore Azure Virtual Machines or disks to a different subscription within the same tenant as the source subscription (as per the Azure RBAC capabilities) from restore points.

Restore option	Details
	Additionally, Cross Zone Restore supports restoring Azure non-zone pinned VMs from a Recovery Services vault configured with Zonal-Redundant Storage (ZRS), enabling enhanced resiliency and flexibility in disaster recovery scenarios. Allowed only if the Cross Subscription Restore property is enabled for your Recovery Services vault. Works with Cross Region Restore and Cross Zonal Restore. You can trigger Cross Subscription Restore for managed virtual machines only. Cross Subscription Restore is supported for Restore with Managed System Identities (MSI). It's unsupported for snapshots tier recovery points. It's unsupported for unmanaged VMs and ADE encrypted VMs.
Cross Zonal Restore	Allows you to restore Azure Virtual Machines or disks pinned to any zone to different available zones (as per the Azure RBAC capabilities) from restore points. Note that when you select a zone to restore, it selects the logical zone (and not the physical zone) as per the Azure subscription you will use to restore to. You can trigger Cross Zonal Restore for managed virtual machines only. Cross Zonal Restore is supported for Restore with Managed System Identities (MSI). Cross Zonal Restore supports restore of an Azure zone pinned/non-zone pinned VM from a vault with Zonal-redundant storage (ZRS) enabled. Learn how to set Storage Redundancy. It's supported to restore an Azure zone pinned VM only from a vault with Cross Region Restore (CRR) (if the secondary region supports zones) or Zone Redundant Storage (ZRS) enabled. Cross Zonal Restore is supported from secondary regions. It's unsupported from snapshots restore point. It's unsupported for Encrypted Azure VMs.

Tip

To receive alerts/notifications when a restore operation fails, use Azure Monitor alerts for Azure Backup. This helps you to monitor such failures and take necessary actions to remediate the issues.

> **Note**
>
> You can also recover specific files and folders on an Azure VM. Learn more.

Storage accounts

Some details about storage accounts:

- **Create VM**: When you create a new VM, VHD files are also copied when restoring VMs with disks < 4 TB or VMs containing < 16 disks from a Vault-Standard recovery point. These files are then moved to Managed storage. To prevent extra charges, delete VHDs from the Staging Storage Account.
- **Restore disk**: The VM restore job generates a template, which you can download and use to specify custom VM settings. VHD files are also copied when restoring managed disks < **4 TB** or VMs containing < **16 disks** from a Vault-Standard recovery point, or when restoring unmanaged disks. These files are then moved to Managed storage. To prevent extra charges, delete VHDs from the Staging Storage Account.
- **Replace disk**: When replacing a managed disk from a Vault-Standard recovery point that's < **4 TB** or a VM containing < **16 disks**, a VHD file is created in the specified storage account. After replacement, source VM disks remain in the designated Resource Group, and VHDs stay in the storage account; you can delete or retain the source disk and the VHDs as needed.
- **Storage account location**: The storage account must be in the same region as the vault. Only these accounts are displayed. If there are no storage accounts in the location, you need to create one.
- **Storage type**: Blob Storage account isn't supported because the Premium Storage account it requires is excluded for cost optimization.
- **Storage redundancy**: Zone redundant storage (ZRS) is supported. The replication and redundancy information for the account is shown in parentheses after the account name.
- **Premium storage**:
  - When you restore non-premium VMs, premium storage accounts aren't supported.
  - When you restore managed VMs, premium storage accounts configured with network rules aren't supported.

Before you start

To restore a VM (create a new VM), make sure you have the correct Azure role-based access control (Azure RBAC) permissions for the Restore VM operation.

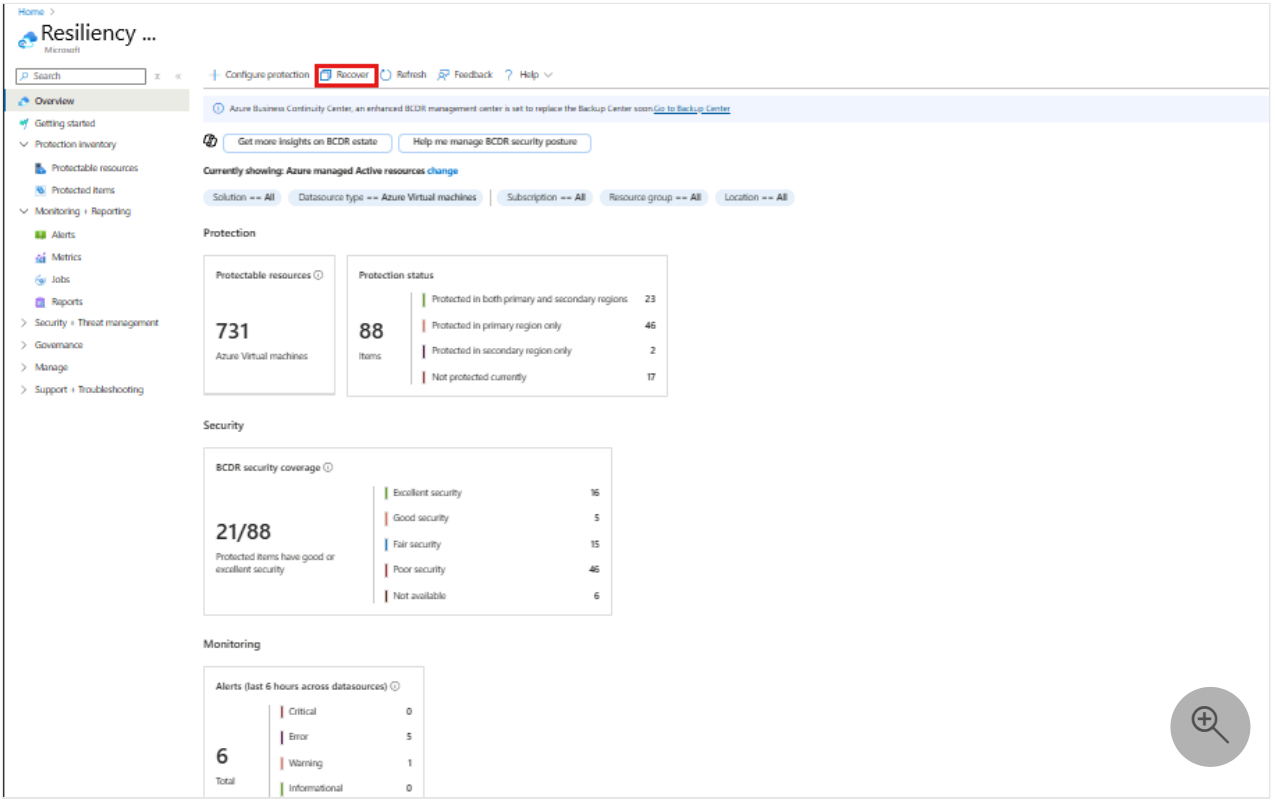
If you don't have permissions, you can restore a disk, and then after the disk is restored, you can use the template that was generated as part of the restore operation to create a new VM.

> **Note**
>
> The functionality that the following sections describe can also be accessed via Resiliency. Resiliency in Azure is a cloud-native unified experience for resiliency posture management that enables you to manage your protection estate across solutions and environments.

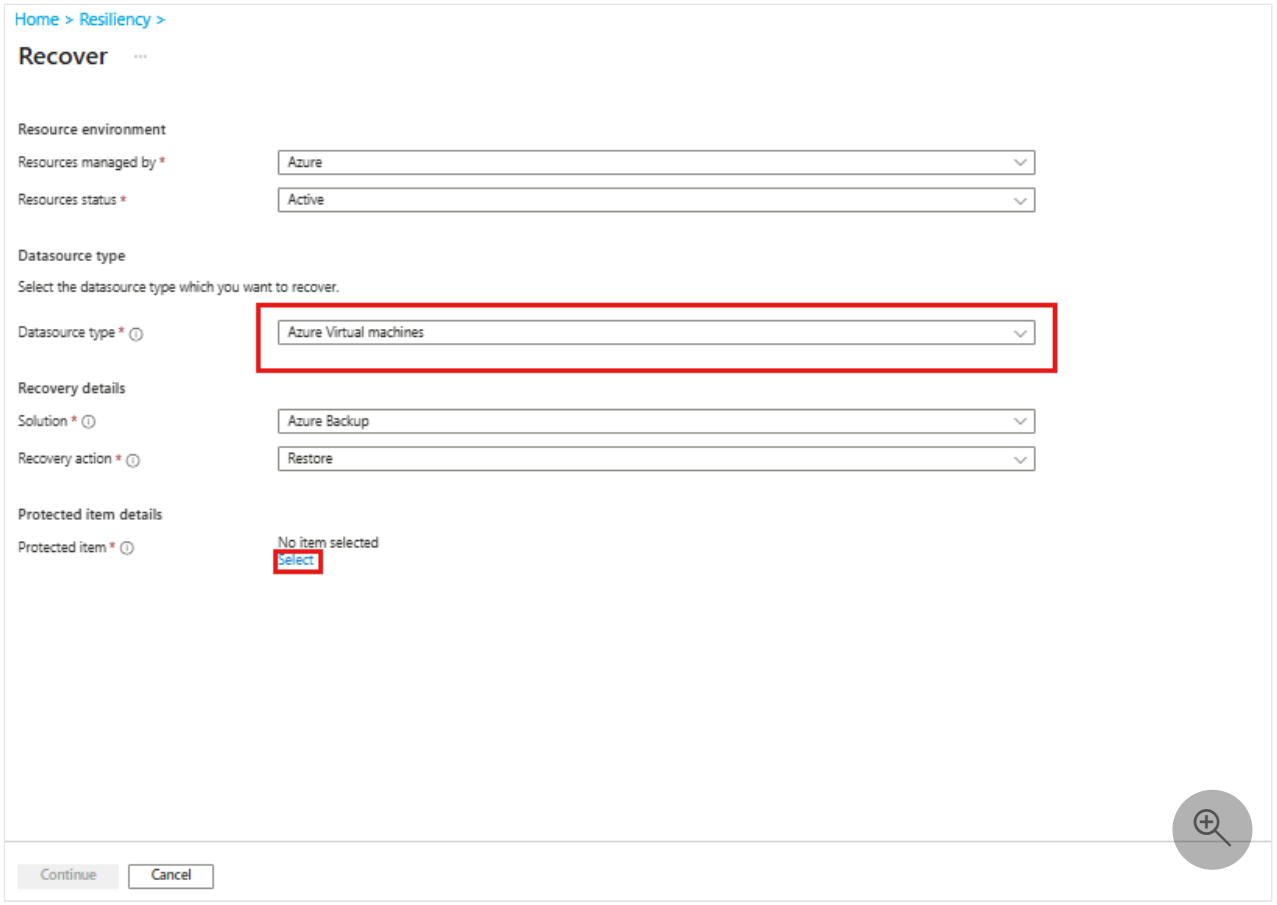
Select a restore point

To select a restore point for a VM restore, follow these steps:

1. Go to **Resiliency** in the Azure portal and select **Recover**.



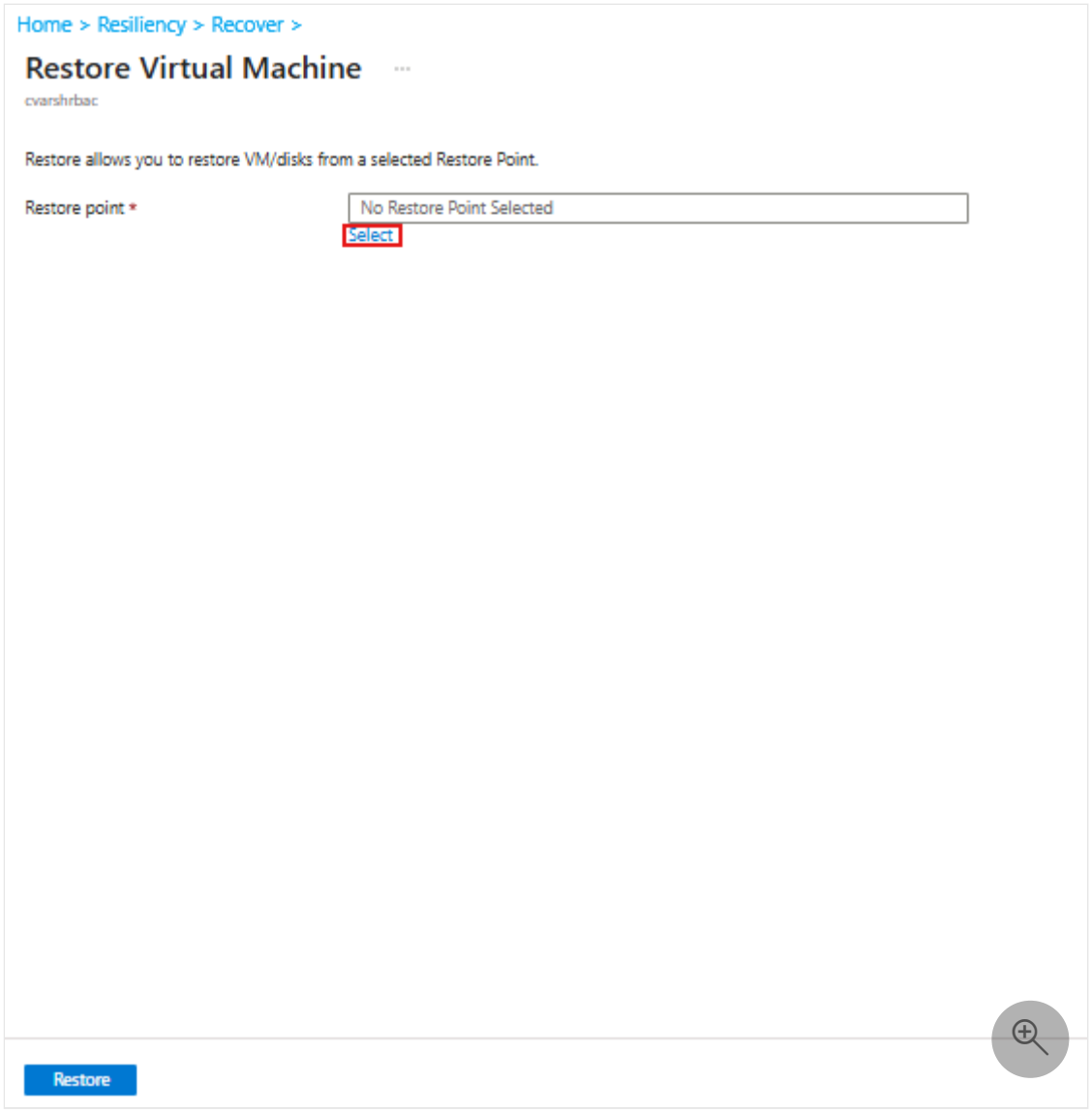
2. On the **Recover** pane, select **Datasource type** as **Azure Virtual machines**, and then under **Protected item**, click **Select**.



3. On the **Select Protected item** pane, select a protected VM from the list, and then click **Select**.

4. On the **Recover** pane, select **Continue**.

5. On the **Restore Virtual Machine** pane, under **Restore point**, click **Select**.

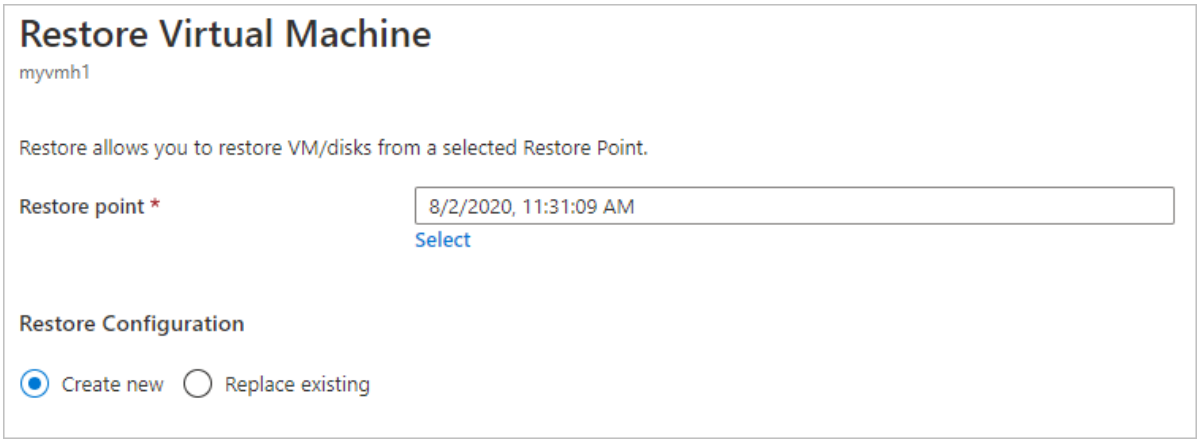


6. On the **Select restore point** pane, select a restore point from the list, and then select **OK** to use for recovery.

Choose a VM restore configuration

1. In **Restore Virtual Machine**, select a restore option:

   - **Create new**: Use this option if you want to create a new VM. You can create a VM with simple settings, or restore a disk and create a customized VM.
   - **Replace existing**: Use this option if you want to replace disks on an existing VM.



2. Specify settings for your selected restore option.

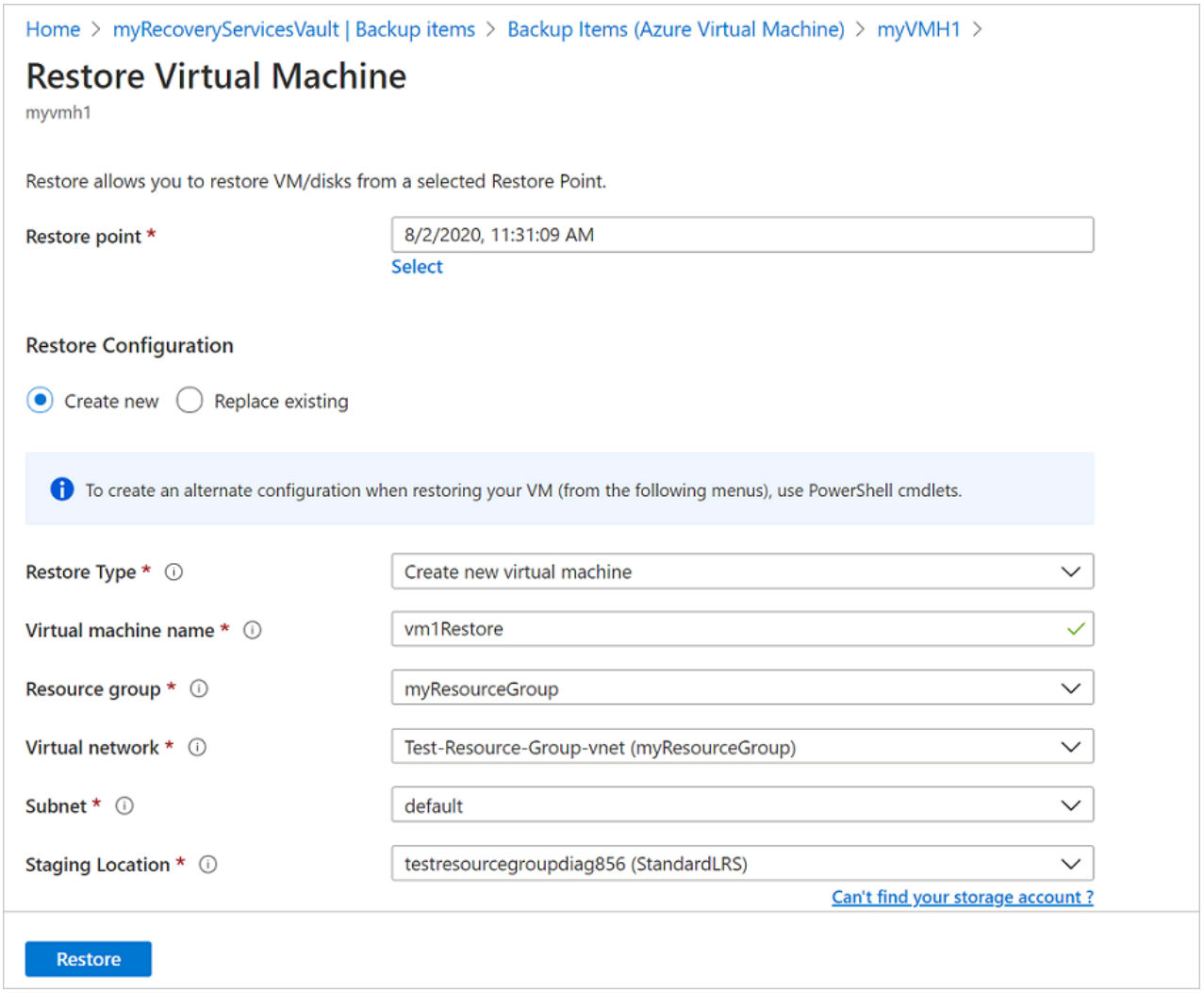
> **Note**
>
> Use the **Replace existing** option only when the **Transfer Data to Vault** subtask in the job details shows successfully completed. Otherwise, use the **Create New** option for the latest recovery point restoration.

Create a VM

As one of the restore options, you can create a VM quickly with basic settings from a restore point.

1. In **Restore Virtual Machine** > **Create new** > **Restore Type**, select **Create new virtual machine**.
2. In **Virtual machine name**, specify a VM that doesn't exist in the subscription.
3. In **Resource group**, select an existing resource group for the new VM, or create a new one with a globally unique name. If you assign a name that already exists, Azure assigns the group the same name as the VM.
4. In **Virtual network**, select the VNet in which the VM will be placed. All VNets associated with the subscription in the same location as the vault, which is active and not attached with any affinity group, are displayed. Select the subnet.

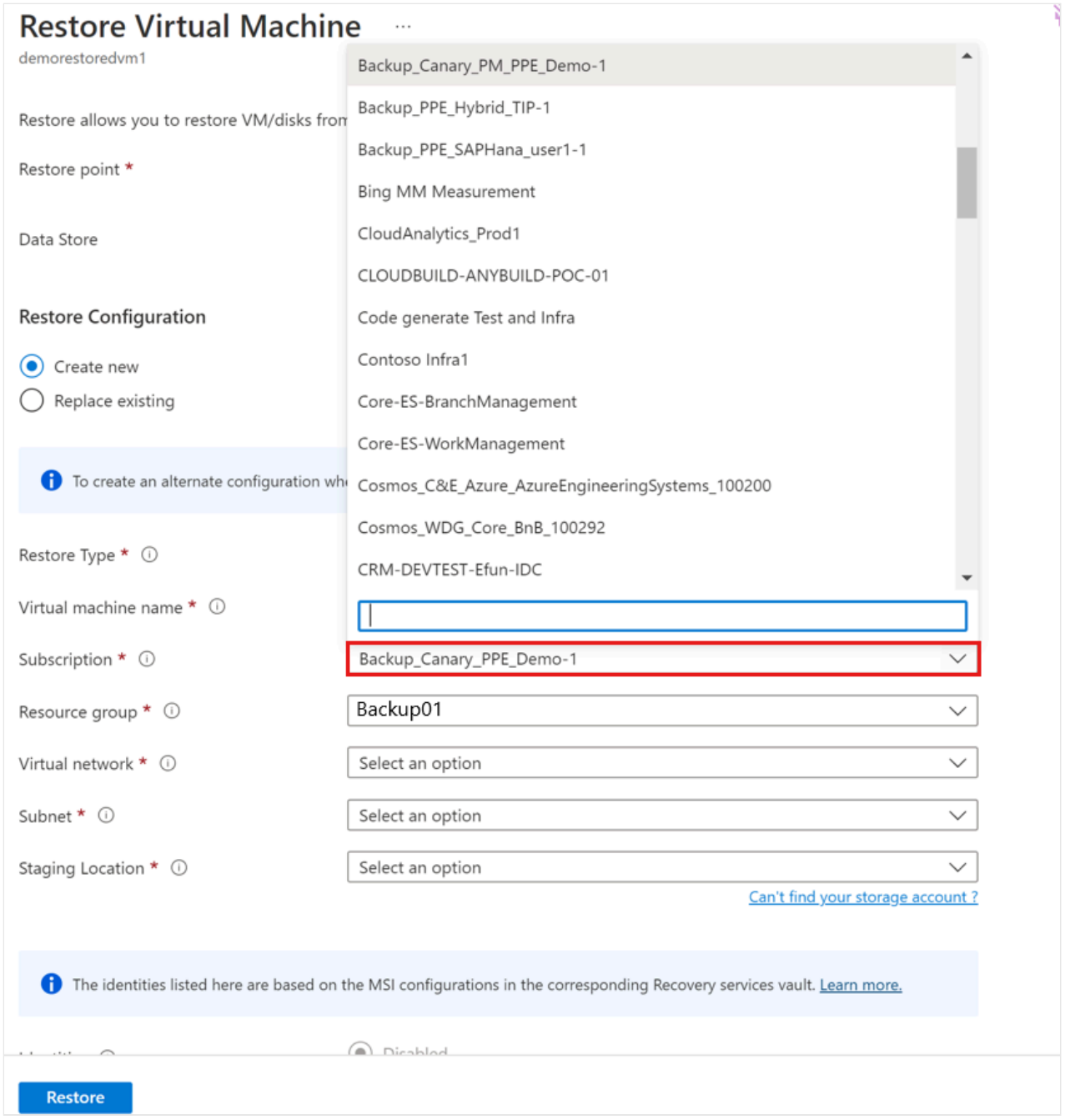
   The first subnet is selected by default.
5. In **Staging Location**, specify the storage account for the VM. Learn more.



6. Choose the required subscription from the **Subscription** drop-down list to restore an Azure VM to a different subscription.

   Azure Backup now supports Cross Subscription Restore (CSR), you can now restore an Azure VM using a recovery point from default subscription to another. Default subscription is the subscription where recovery point is available.

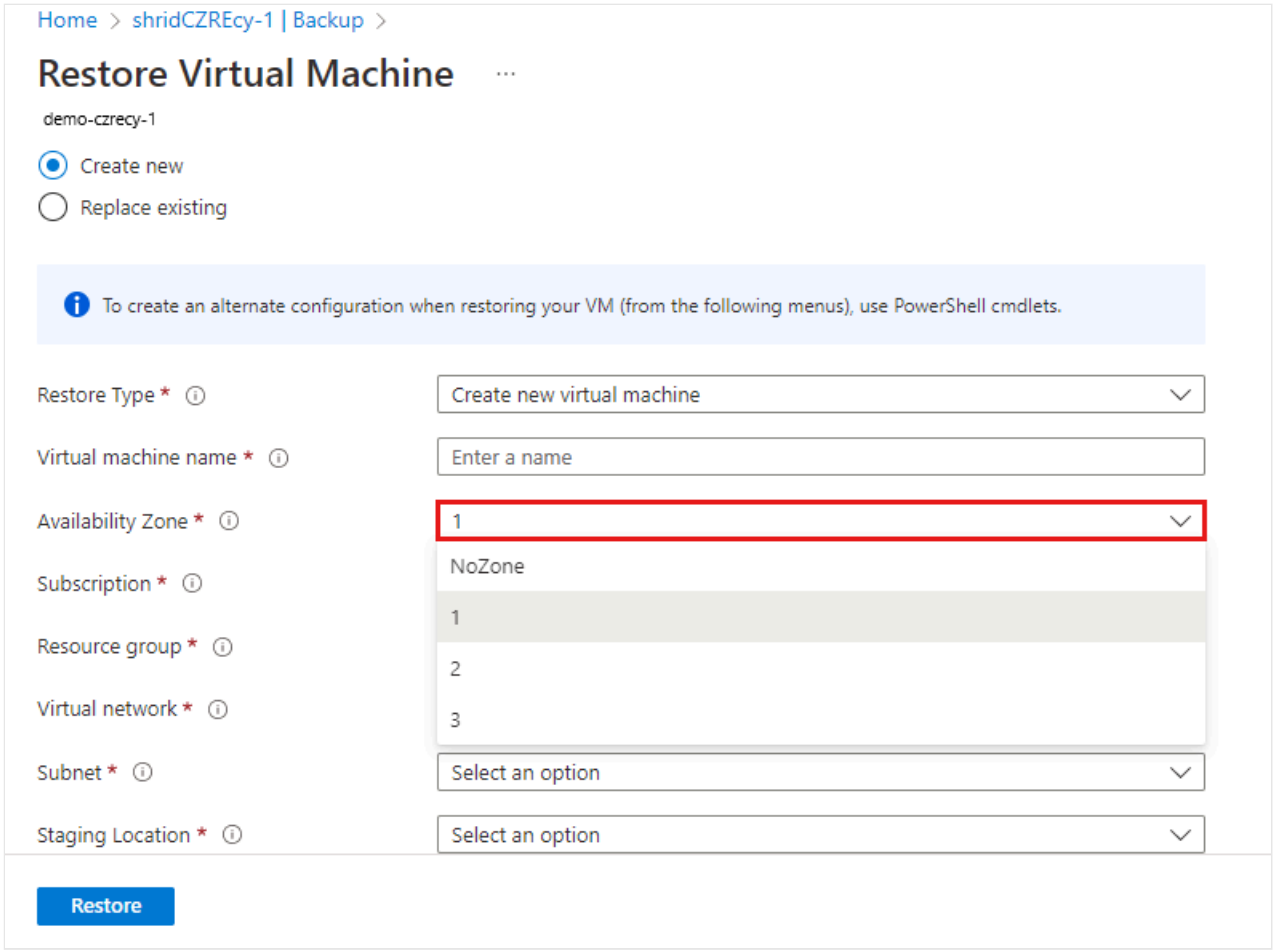
   The following screenshot lists all subscriptions under the tenant where you've permissions, which enable you to restore the Azure VM to another subscription.



7. Choose the required zone from the **Availability Zone** drop-down list to restore an Azure VM pinned to any zone to a different zone.

   Azure Backup now supports Cross Zonal Restore (CZR), you can now restore an Azure VM from the default zone to any available zones. Default zone is the zone in which Azure VM is running.

   The following screenshot lists all zones that enable you to restore Azure VM to another zone.



> ⓘ **Note**
>
> Azure Backup supports CZR only for vaults with ZRS or CRR redundancy.

8. Select **Restore** to trigger the restore operation.

> ⓘ **Note**
>
> - Before you modify any NSG settings, ensure the VM restore operation is complete. Learn about tracking the restore operation.
> - During an Alternate Location Restore (Create new VM), the restored VM retains the same SKU as the original VM at the time of the recovery point creation. Any SKU changes made after the backup date don't reflect in the restored VM.

Restore disks

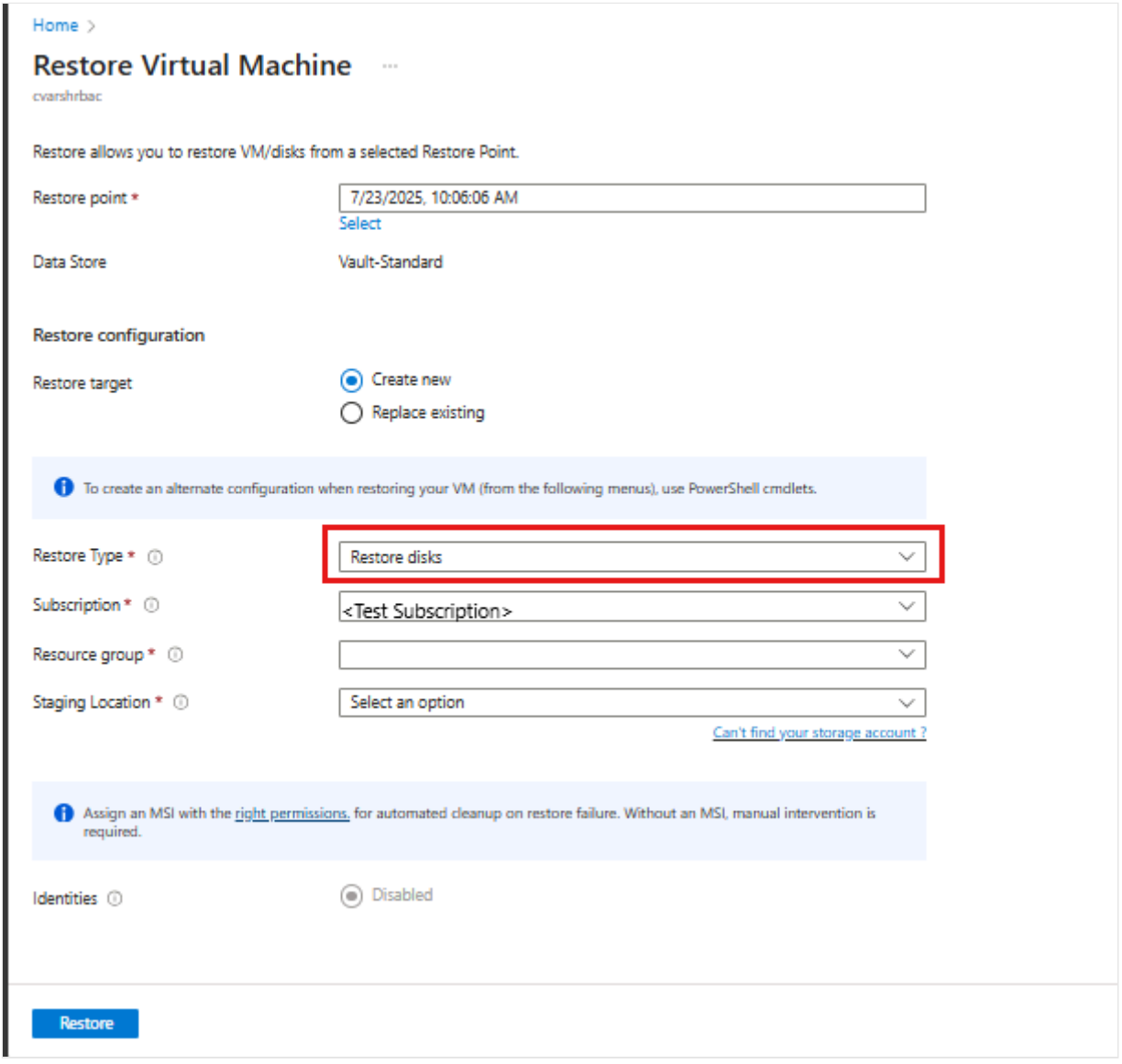
As one of the restore options, you can create a disk from a restore point. Then with the disk, you can do one of the following actions:

- Use the template that's generated during the restore operation to customize settings, and trigger VM deployment. You edit the default template settings, and submit the template for VM deployment.
- Attach restored disks to an existing VM.
- Create a new VM from the restored disks using PowerShell.

1. Under **Restore configuration** > **Create new** > **Restore Type**, select **Restore disks**.

2. Under **Resource group**, select an existing resource group for the restored disks, or create a new one with a globally unique name.

3. On **Staging location**, specify the storage account. The template file is stored here, and VHD files are also created in some scenarios. Learn more.



4. Choose the required subscription from the **Subscription** drop-down list to restore the VM disks to a different subscription.

   Azure Backup now supports Cross Subscription Restore (CSR). Like Azure VM, you can now restore Azure VM disks using a recovery point from default subscription to another.

Default subscription is the subscription where recovery point is available.

5. Choose the required zone from the **Availability Zone** drop-down list to restore the VM disks to a different zone.

   Azure Backup now supports Cross Zonal Restore (CZR). Like Azure VM, you can now restore Azure VM disks from the default zone to any available zones. Default zone is the zone in which the VM disks reside.

   > ⓘ **Note**
   >
   > Azure Backup supports CZR only for vaults with ZRS or CRR redundancy.

6. Select **Restore** to trigger the restore operation.

When your virtual machine uses managed disks and you select the **Create virtual machine** option, Azure Backup doesn't use the specified storage account. In the case of **Restore disks** and **Instant Restore**, the storage account is used only for storing the template. Managed disks are created in the specified resource group. When your virtual machine uses unmanaged disks, they're restored as blobs to the storage account.

While you restore disks for a Managed VM from a Vault-Standard recovery point, it restores the Managed disk and Azure Resource Manager (ARM) templates, along with the VHD files of the disks in staging location. If you restore disks from an Instant recovery point, it restores the Managed disks and ARM templates only.

> ⓘ **Note**
>
> - For restoring disk from a Vault-Standard recovery point that is/was greater than 4 TB, Azure Backup doesn't restore the VHD files.
> - For information on managed/premium disk performance after restored via Azure Backup, see the Latency section.

Use templates to customize a restored VM

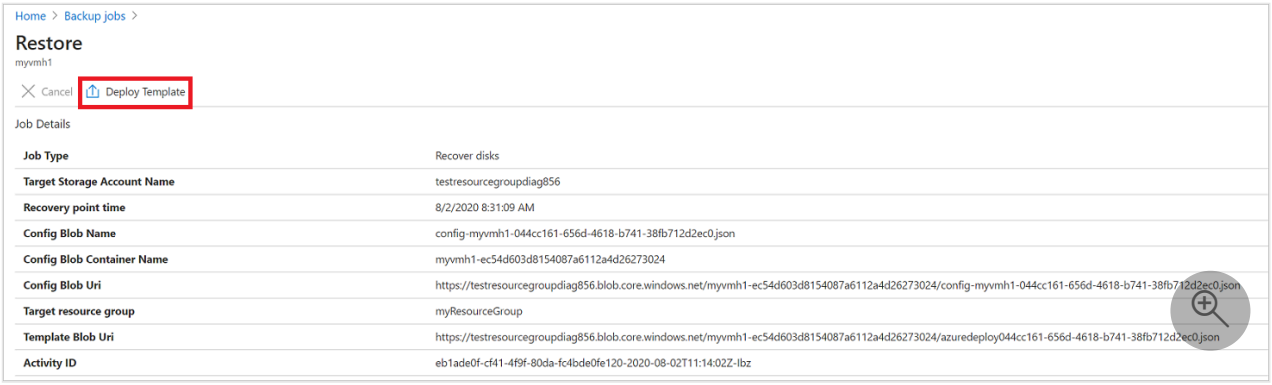
After the disk is restored, use the template that was generated as part of the restore operation to customize and create a new VM:

1. Go to the **Recovery Services vault**, and then select **Monitoring** > **Backup Jobs**.

   Alternatively, you can go to **Resiliency**, and then select **Monitoring + Reporting** > **Jobs**.

2. On the **Backup Jobs** pane, select the relevant restore job.

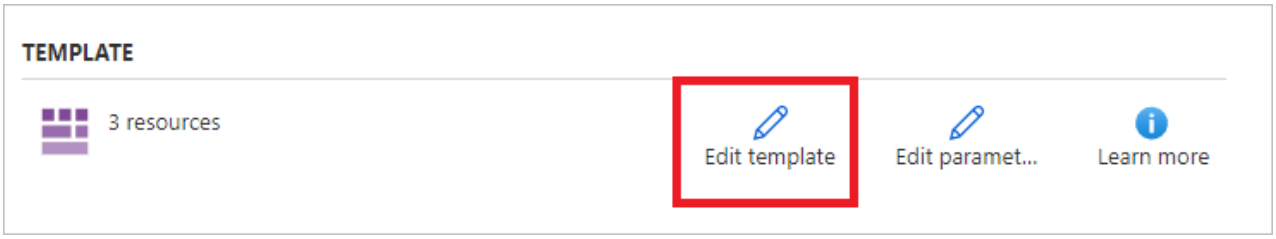
3. In **Restore**, select **Deploy Template** to initiate template deployment.



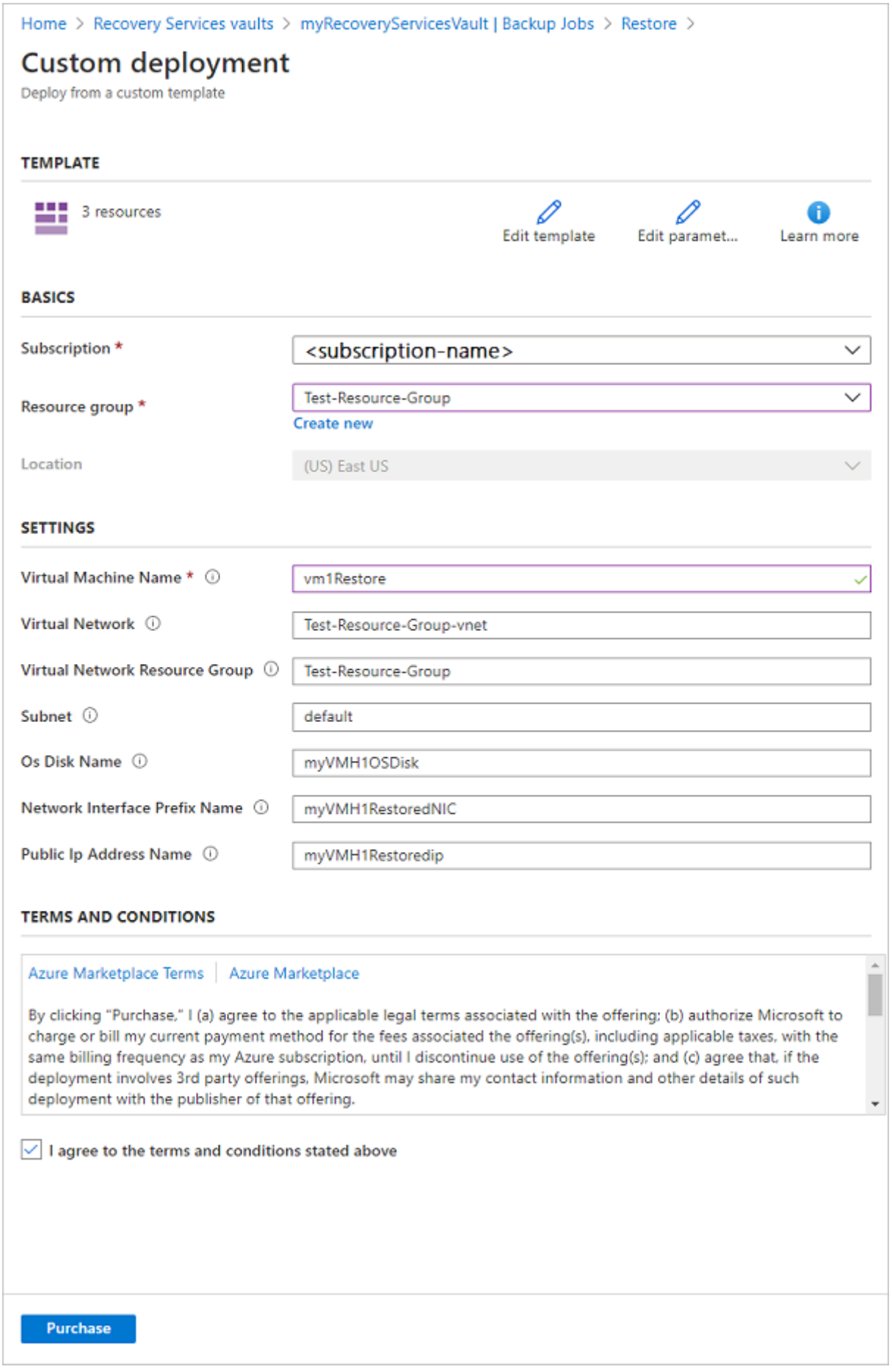
> **Note**
>
> For a shared access signature (SAS) that has **Allow storage account key access** set to disabled, the template won't deploy when you select **Deploy Template**.

4. To customize the VM setting provided in the template, select **Edit template**. If you want to add more customizations, select **Edit parameters**.

   - Learn more about deploying resources from a custom template.
   - Learn more about authoring templates.



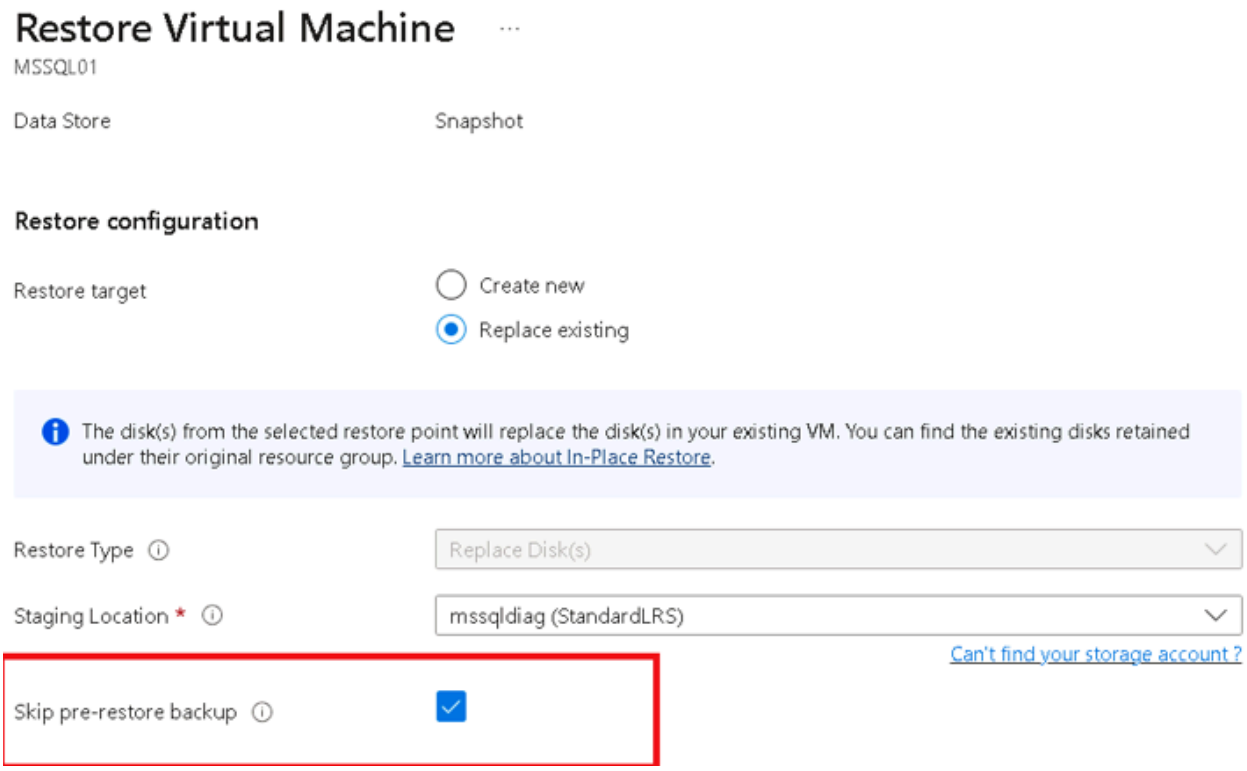
5. Enter the custom values for the VM, accept the **Terms and Conditions** and select **Purchase**.



Replace existing disks

As one of the restore options, you can replace an existing VM disk with the selected restore point. Review all restore options.

1. In **Restore configuration**, select **Replace existing**.
2. In **Restore Type**, select **Replace disk/s**. This is the restore point that will be used to replace existing VM disks.
3. In **Staging Location**, specify a storage account. VHD files are created here in some scenarios. Learn more.



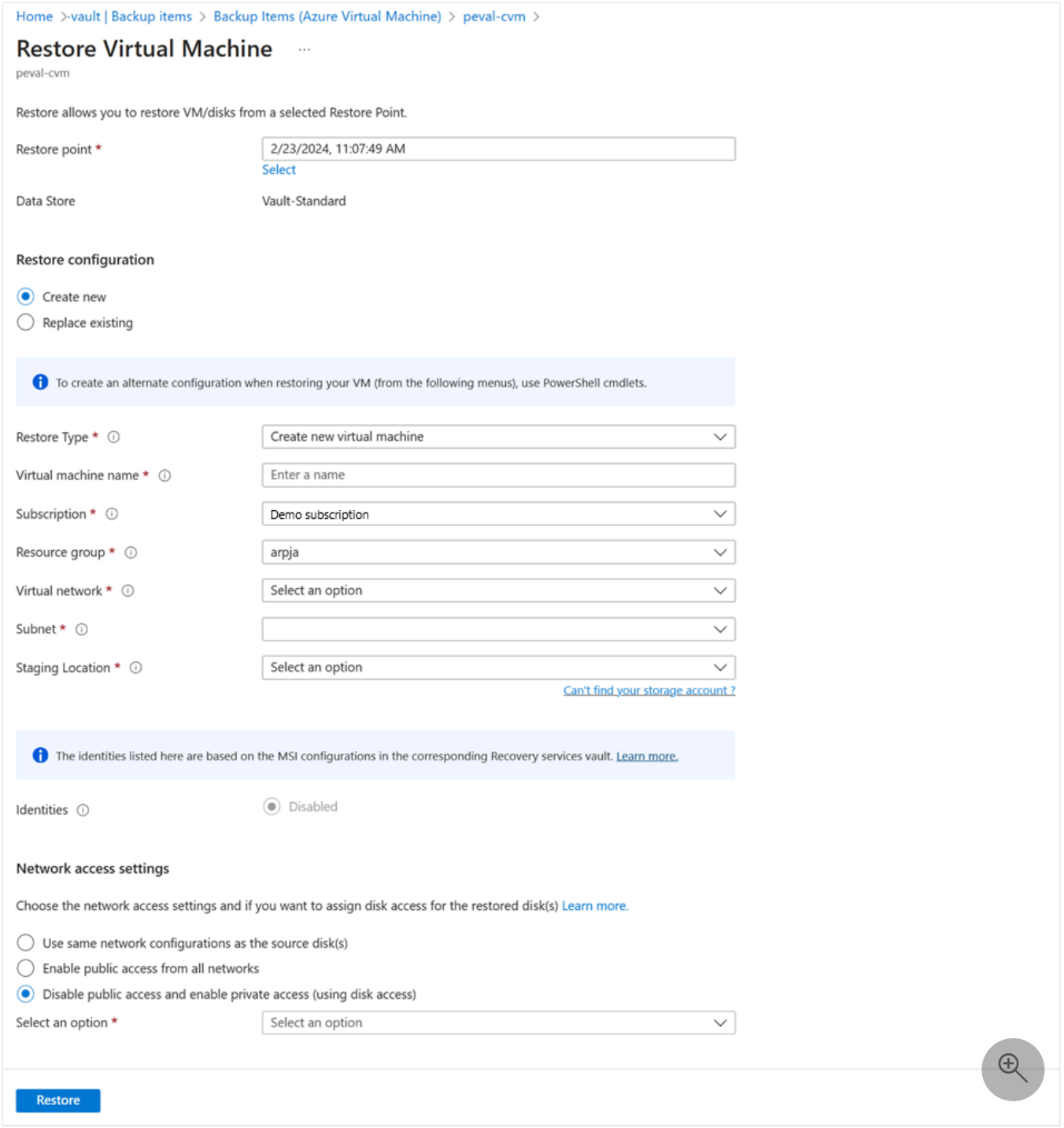
Assign network access settings during restore

Azure Backup also allows you to configure the access options for the restored disks once the restore operation is complete. You can set the disk access preferences at the time of initiating the restore.

> **Note**
>
> This feature is generally available for backed-up VMs that use private endpoint-enabled disks.

To enable disk access on restored disks during VM restore, choose one of the following options:

- **Use the same network configurations as the source disk(s)**: This option allows the restored disks to use the disk access and network configurations same as that of the source disks.
- **Enable public access from all networks**: This option allows the restored disk to be publicly accessible from all networks.
- **Disable public access and enable private access (using disk access)**: This option allows you to disable the public access and assign disk access to the restored disks for private access.



> **Note**

The option to choose the network configuration of the restored disks the same as that of the source disks or specify the access from specific networks only is currently not available from Azure PowerShell/ Azure CLI.

Cross Region Restore

As one of the restore options, Cross Region Restore (CRR) allows you to restore Azure VMs in a secondary region, which is an Azure paired region.

To begin using the feature, read the Before You Begin section.

To see if CRR is enabled, follow the instructions in Configure Cross Region Restore.

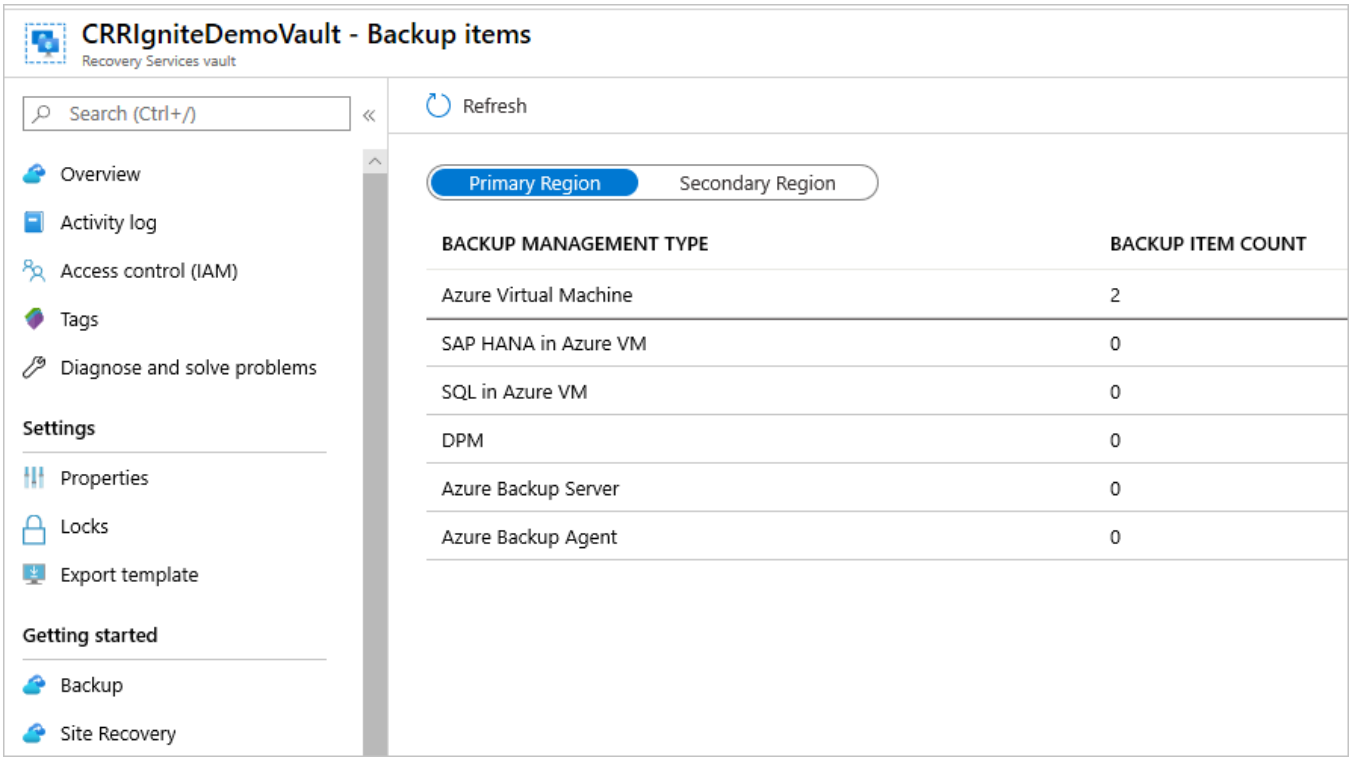
> **Note**
>
> Cross-region restore is currently not supported for machines running on Ultra disks. **Learn more about Ultra disk backup supportability**.

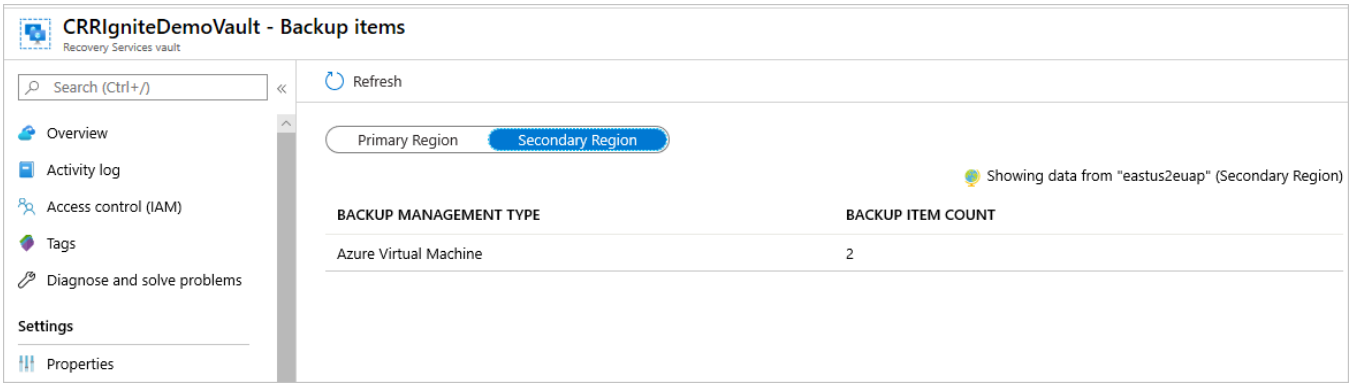
View backup items in secondary region

If CRR is enabled, you can view the backup items in the secondary region.

1. From the portal, go to **Recovery Services vault** > **Backup items**.
2. Select **Secondary Region** to view the items in the secondary region.

> **Note**
>
> Only Backup Management Types supporting the CRR feature will be shown in the list. Currently, only support for restoring secondary region data to a secondary region is allowed.
>
> CRR for Azure VMs is supported for Azure Managed VMs (including encrypted Azure VMs). See the **management types that support Cross Region Restore**.

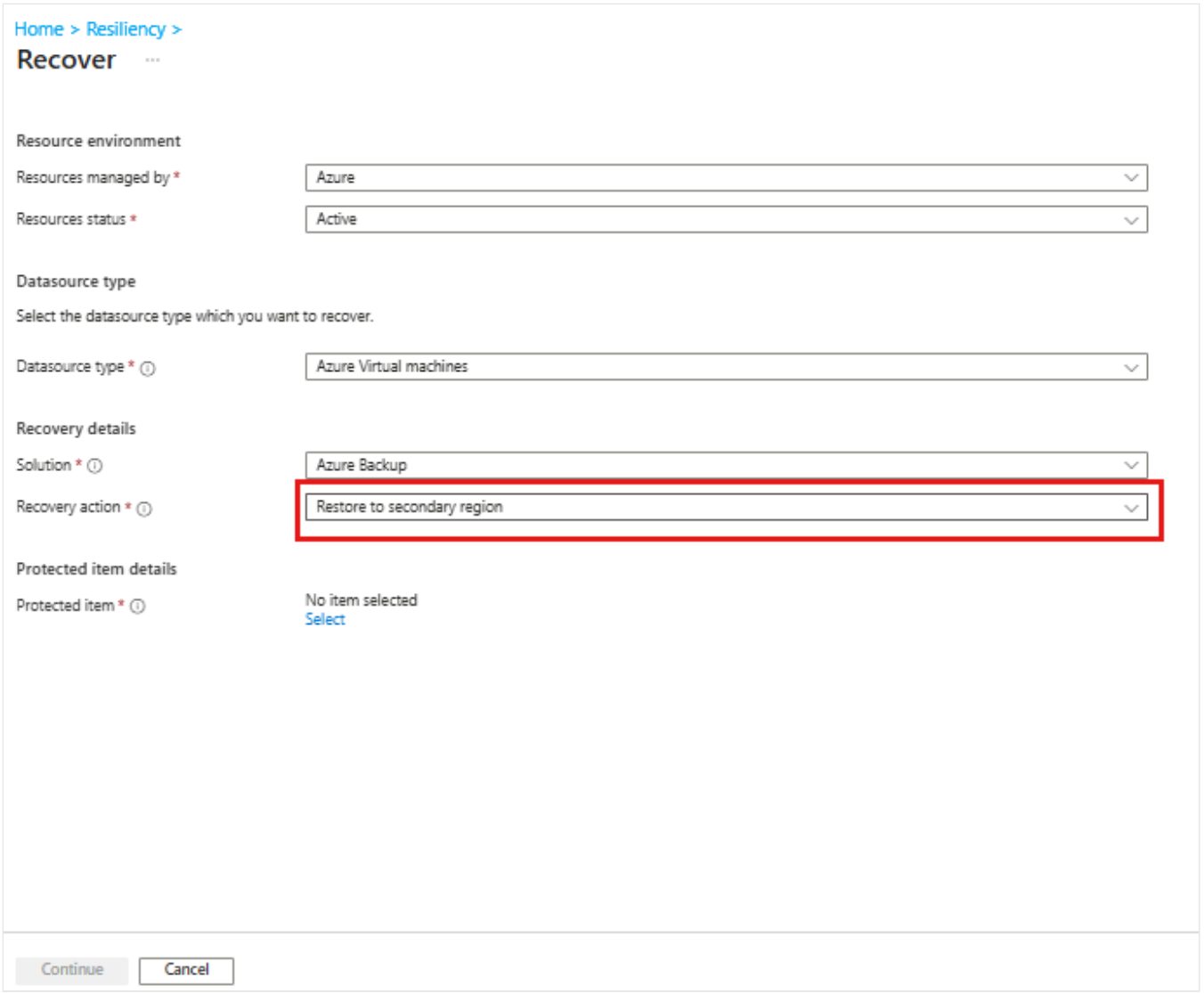




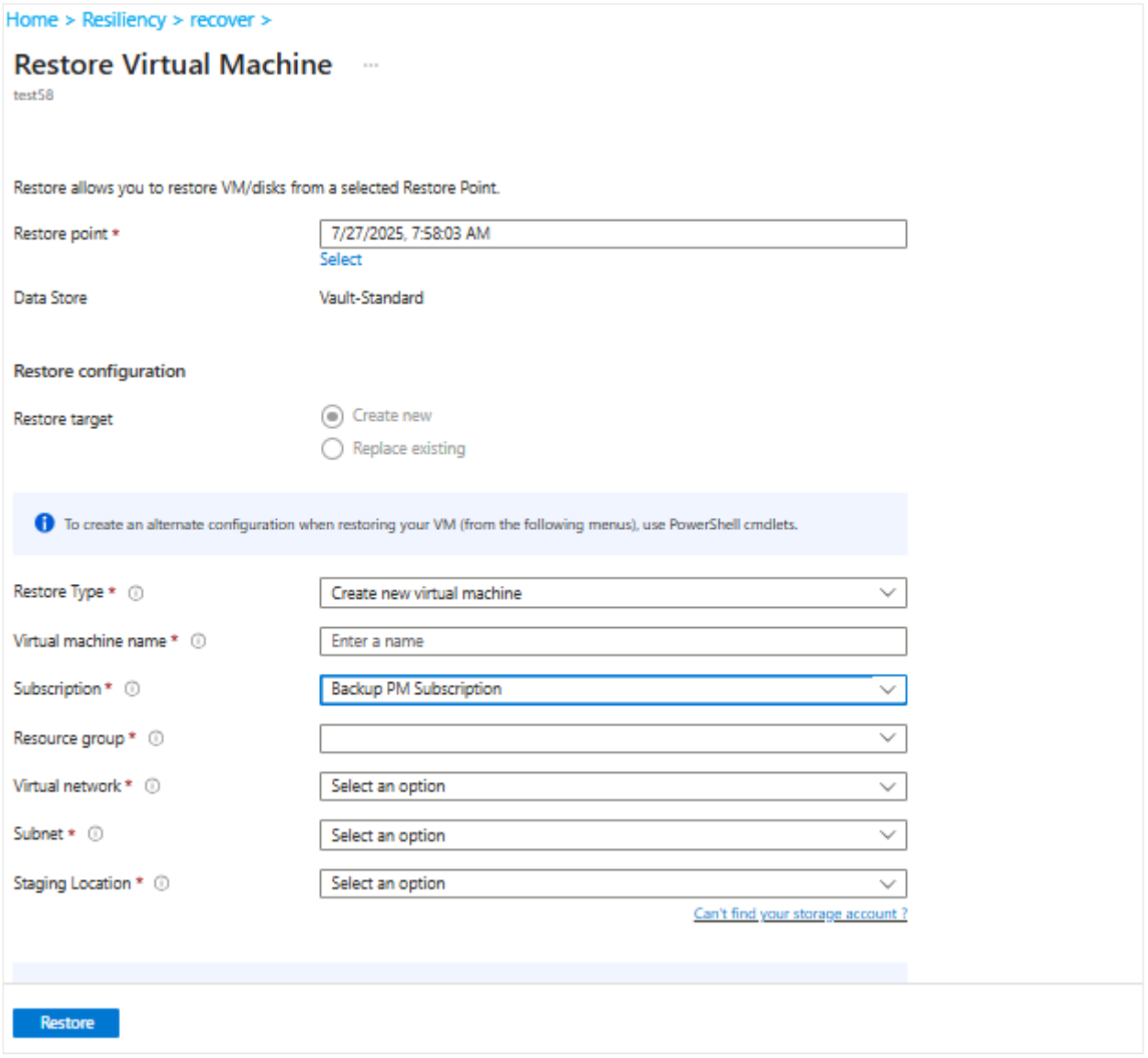
Restore in secondary region

The secondary region restore user experience is similar to the primary region restore user experience. When configuring details in the Restore Configuration pane to configure your restore, you're prompted to provide only secondary region parameters.

Currently, secondary region RPO is *36 hours*. This is because the RPO in the primary region is *24 hours* and can take up to *12 hours* to replicate the backup data from the primary to the secondary region.



Screenshot shows how to start secondary region restore of a VM. ":::



- To restore and create a VM, refer to Create a VM.
- To restore as a disk, refer to Restore disks.

> **Note**
>
> - The Cross Region Restore feature restores CMK (customer-managed keys) enabled Azure VMs, which aren't backed-up in a CMK enabled Recovery Services vault, as non-CMK enabled VMs in the secondary region.
> - The Azure roles needed to restore in the secondary region are the same as those in the primary region.
> - When you restore an Azure VM, Azure Backup configures the virtual network settings in the secondary region automatically. If you are restoring disks while deploying the template, ensure to provide the virtual network settings, corresponding to the secondary region.
> - If VNet/Subnet is not available in the primary region or is not configured in the secondary region, Azure portal doesn't auto-populate any default values during

restore operation.

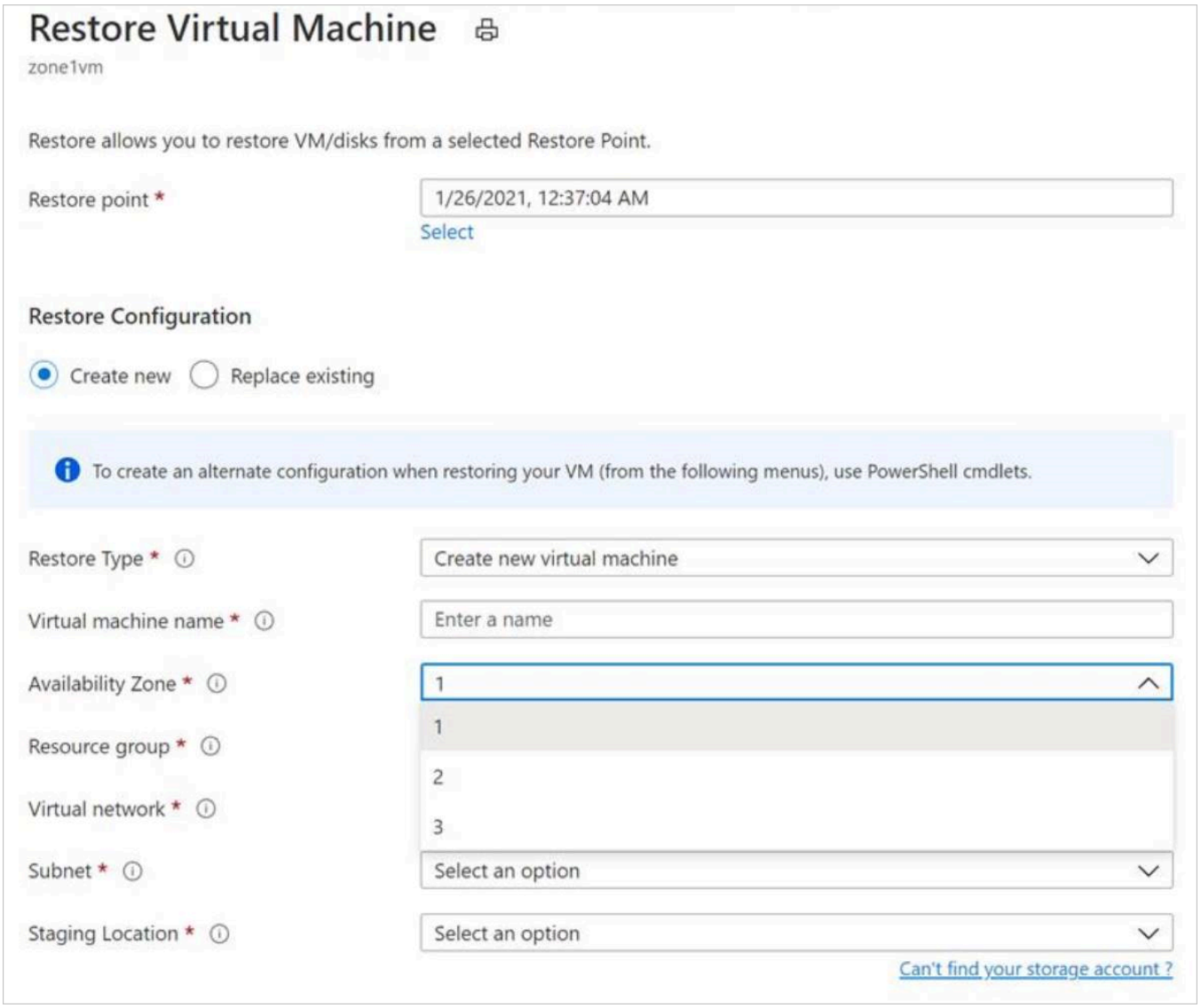
- For Cross Region Restores, the **Staging Location** (that is the storage account location) must be in the region that the Recovery Services vault treats as the *secondary* region. For example, a Recovery Services vault is located in East US 2 region (with Geo-Redundancy and Cross Region Restore enabled). This means that the *secondary* region would be *Central US*. Therefore, you need to create a storage account in *Central US* to perform a Cross Region Restore of the VM.
  Learn more about **Azure cross-region replication pairings for all geographies**.

Azure zone pinned VMs can be restored in any availability zones of the same region.

In the restore process, you'll see the option **Availability Zone.** You'll see your default zone first. To choose a different zone, choose the number of the zone of your choice. If the pinned zone is unavailable, you won't be able to restore the data to another zone because the backed-up data isn't zonally replicated. The restore in availability zones is possible from recovery points in vault tier only.

In summary, the **Availability Zone** will only appear when

- The source VM is zone pinned and is NOT encrypted
- The recovery point is present in vault tier only (Snapshots only or snapshot and vault tier are not supported)
- The recovery option is to either create a new VM or to restore disks (replace disks option replaces source data and hence the availability zone option is not applicable)
- Creating VM/disks in the same region when vault's storage redundancy is ZRS (Doesn't work when vault's storage redundancy is GRS even though the source VM is zone pinned)
- Creating VM/disks in the paired region when vault's storage redundancy is enabled for Cross-Region-Restore AND if the paired region supports zones

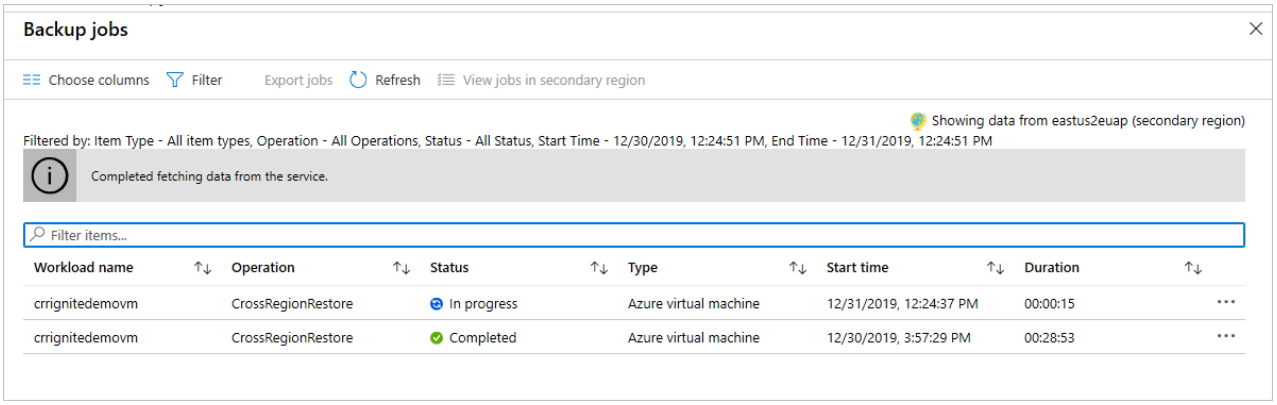


> ⓘ **Note**
>
> Cross region restore jobs once triggered, can't be canceled.

Monitoring secondary region restore jobs

1. From the portal, go to **Recovery Services vault** > **Backup Jobs**

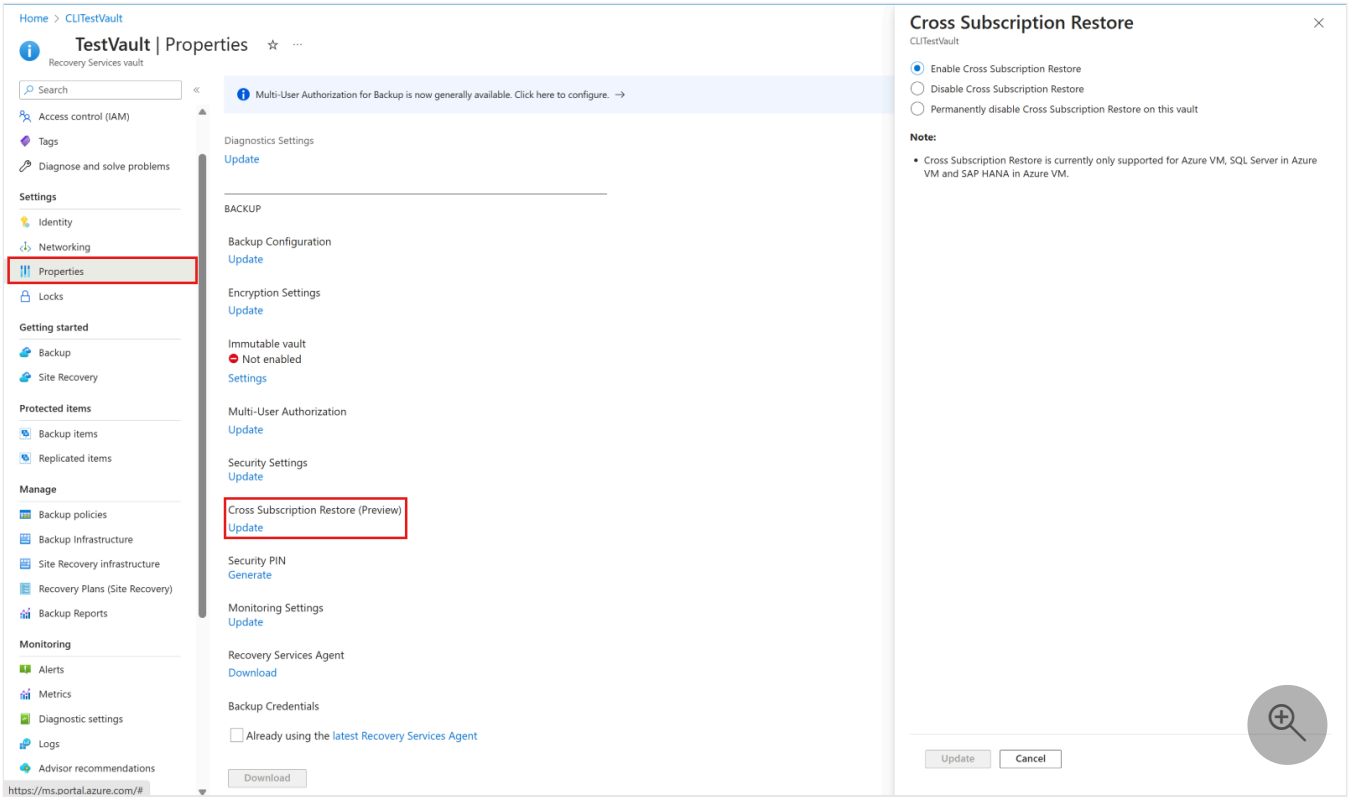
2. Select **Secondary Region** to view the items in the secondary region.



Cross Subscription Restore for Azure VM

Azure Backup now allows you to perform Cross Subscription Restore (CSR), which helps you to restore Azure VMs in a subscription that is different from the default one. Default subscription contains the recovery points.

This feature is enabled for Recovery Services vault by default. However, there may be instances when you may need to block Cross Subscription Restore based on your cloud infrastructure. So, you can enable, disable, or permanently disable Cross Subscription Restore for the existing vaults by going to *Vault* > **Properties** > **Cross Subscription Restore**.

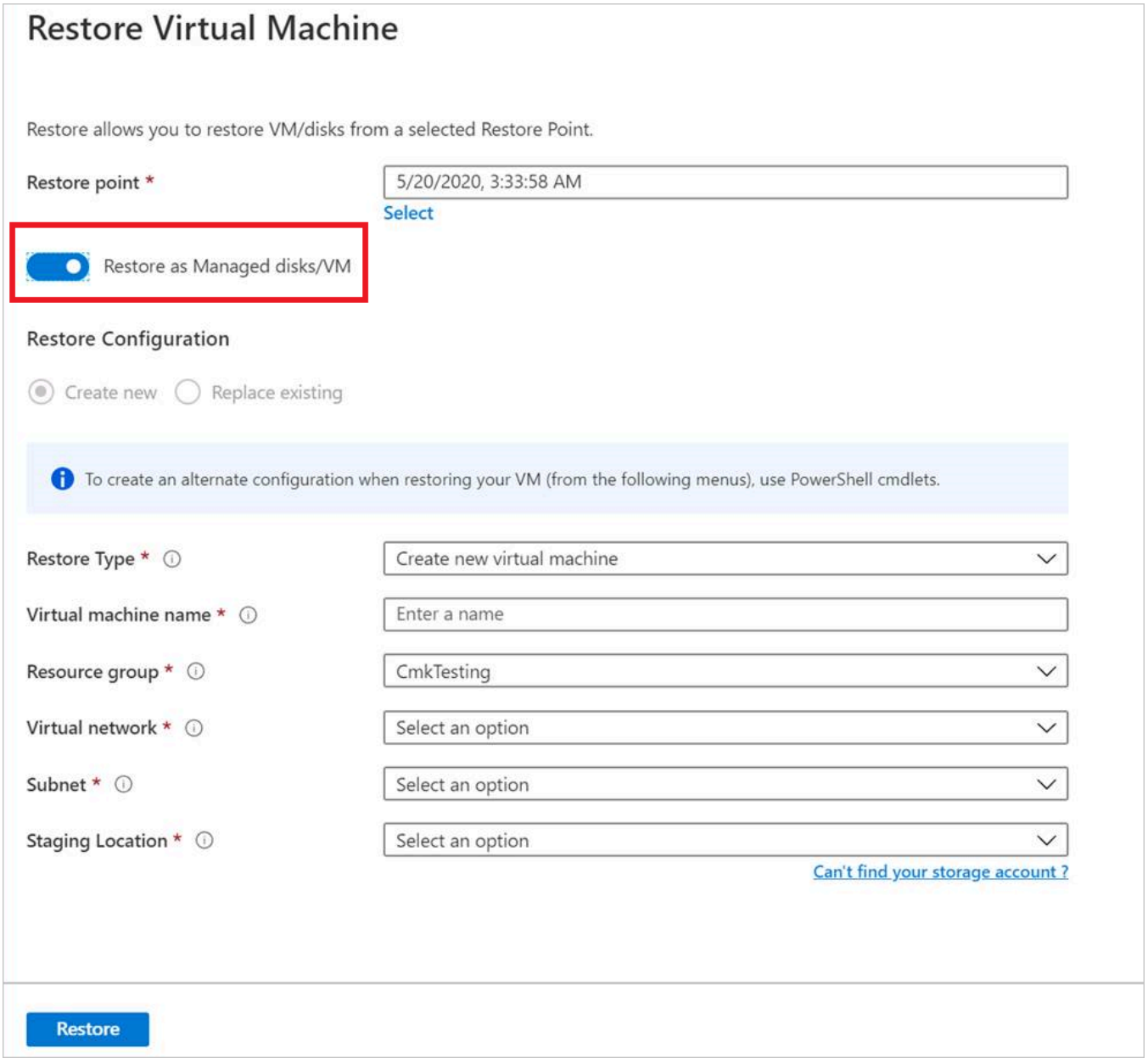


> **Note**
>
> - CSR once permanently disabled on a vault can't be re-enabled because it's an irreversible operation.

- If CSR is disabled but not permanently disabled, then you can reverse the operation by selecting *Vault* > **Properties** > **Cross Subscription Restore** > **Enable**.
- If a Recovery Services vault is moved to a different subscription when CSR is disabled or permanently disabled, restore to the original subscription fails.

Restoring unmanaged VMs and disks as managed

You're provided with an option to restore unmanaged disks as managed disks during restore. By default, the unmanaged VMs / disks are restored as unmanaged VMs / disks. However, if you choose to restore as managed VMs / disks, it's now possible to do so. These restore operations aren't triggered from the snapshot phase but only from the vault phase. This feature isn't available for unmanaged encrypted VMs.



Restore VMs with special configurations

There are many common scenarios in which you might need to restore VMs.

Expand table

Scenario	Guidance
Restore VMs using Hybrid Use Benefit	If a Windows VM uses Hybrid Use Benefit (HUB) licensing, restore the disks, and create a new VM using the provided template (with **License Type** set to **Windows_Server**), or PowerShell. This setting can also be applied after creating the VM.
Restore VMs during an Azure datacenter disaster	If the vault uses GRS and the primary datacenter for the VM goes down, Azure Backup supports restoring backed-up VMs to the paired datacenter. You select a storage account in the paired datacenter, and restore as normal. Azure Backup uses the compute service in the paired region to create the restored VM. Learn more about datacenter resiliency. If the vault uses GRS, you can choose the new feature, Cross Region Restore. This lets you restore to a second region in either full or partial outage scenarios, or even if there's no outage at all.
Bare-metal restore	The major difference between Azure VMs and on-premises hypervisors is that there's no VM console available in Azure. A console is required for certain scenarios, such as recovering by using a bare-metal recovery (BMR)-type backup. However, VM restore from the vault is a full replacement for BMR.
Restore VMs with special network configurations	Special network configurations include VMs using internal or external load balancing, using multiple NICS, or multiple reserved IP addresses. You restore these VMs by using the restore disk option. This option makes a copy of the VHDs into the specified storage account, and you can then create a VM with an internal or external load balancer, multiple NICS, or multiple reserved IP addresses, in accordance with your configuration. To restore VMs with special network configurations, you should use the **Restore Disk** option. This option creates a copy of the virtual hard disks (VHDs) into the specified storage account. Once the VHDs are copied, you can proceed to create a VM that matches your required network configuration, whether it involves internal or external load balancing, multiple NICs, or multiple reserved IP addresses. Note that VM NICs won't be created if the machine has Public IP and NAT gateway. Consider this limitation when you plan to restore VMs with special network configurations.
Network Security Group (NSG) on NIC/Subnet	Azure VM backup supports backing up and restoring NSG information at vnet, subnet, and NIC level.

Scenario	Guidance
Zone Pinned VMs	If you back up an Azure VM that's pinned to a zone (with Azure Backup), then you can restore it in the same zone where it was pinned. Learn more
Restore VM in any availability set	When you restore a VM from the portal, there's no option to choose an availability set. A restored VM doesn't have an availability set. If you use the restore disk option, then you can specify an availability set when you create a VM from the disk using the provided template or PowerShell.
Restore special VMs such as SQL VMs	If you're backing up a SQL VM using Azure VM backup and then use the restore VM option or create a VM after restoring disks, then the newly created VM must be registered with the SQL provider as mentioned here. This will convert the restored VM into a SQL VM.

Restore domain controller VMs

Expand table

Scenario	Guidance
Restore a single domain controller VM in a single domain	Restore the VM like any other VM. Note that: From an Active Directory perspective, the Azure VM is like any other VM. Directory Services Restore Mode (DSRM) is also available, so all Active Directory recovery scenarios are viable. Learn more about backup and restore considerations for virtualized domain controllers.
Restore multiple domain controller VMs in a single domain	If other domain controllers in the same domain can be reached over the network, the domain controller can be restored like any VM. If it's the last remaining domain controller in the domain, or a recovery in an isolated network is performed, use a forest recovery.
Restore a single domain controller VM in a multiple domain configuration	Restore the disks and create a VM by using PowerShell
Restore multiple domains in one forest	We recommend a forest recovery.

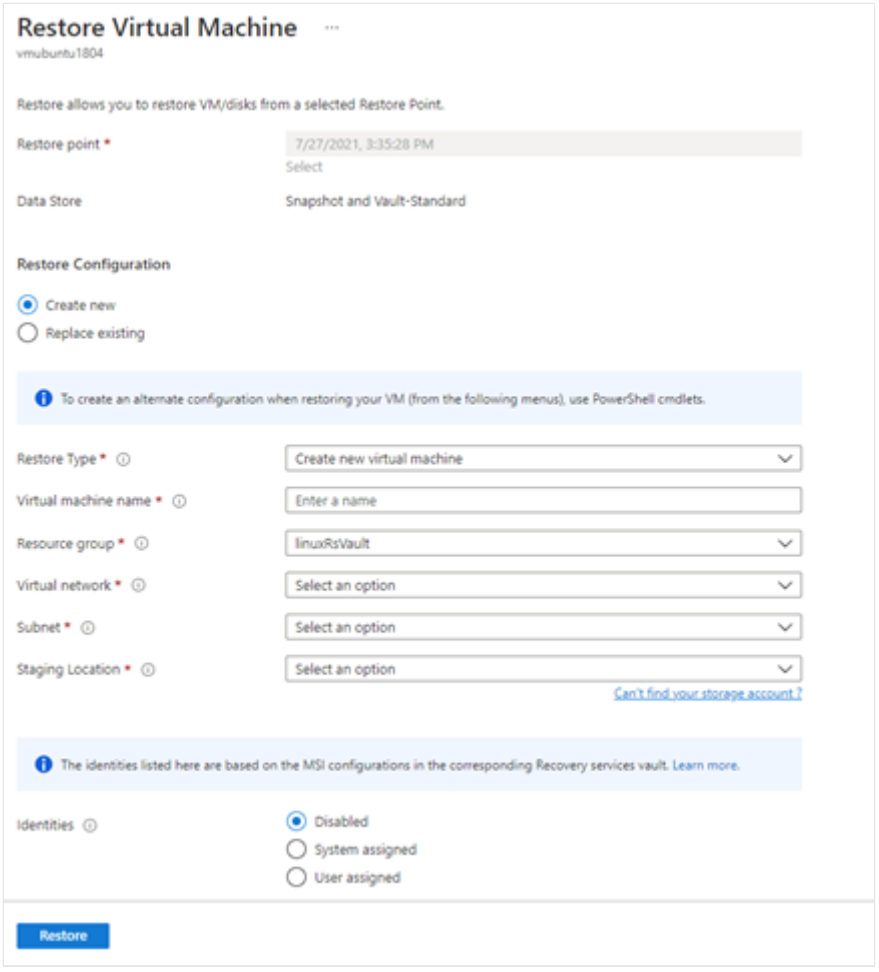
For more information, see Back up and restore Active Directory domain controllers.

Restore VMs with managed identities

Managed identities eliminate the need for the user to maintain the credentials. Managed identities provide an identity for applications to use when connecting to resources that support

Microsoft Entra authentication.

Azure Backup offers the flexibility to restore the managed Azure VM with managed identities. You can choose to select system-managed identities or user-managed identities as shown in the figure below. This is introduced as one of the input parameters in the **Restore configuration** pane of Azure VM. Managed identities are used for accessing the storage accounts and automated cleanup of any resources created during restore process. These managed identities have to be associated to the vault.



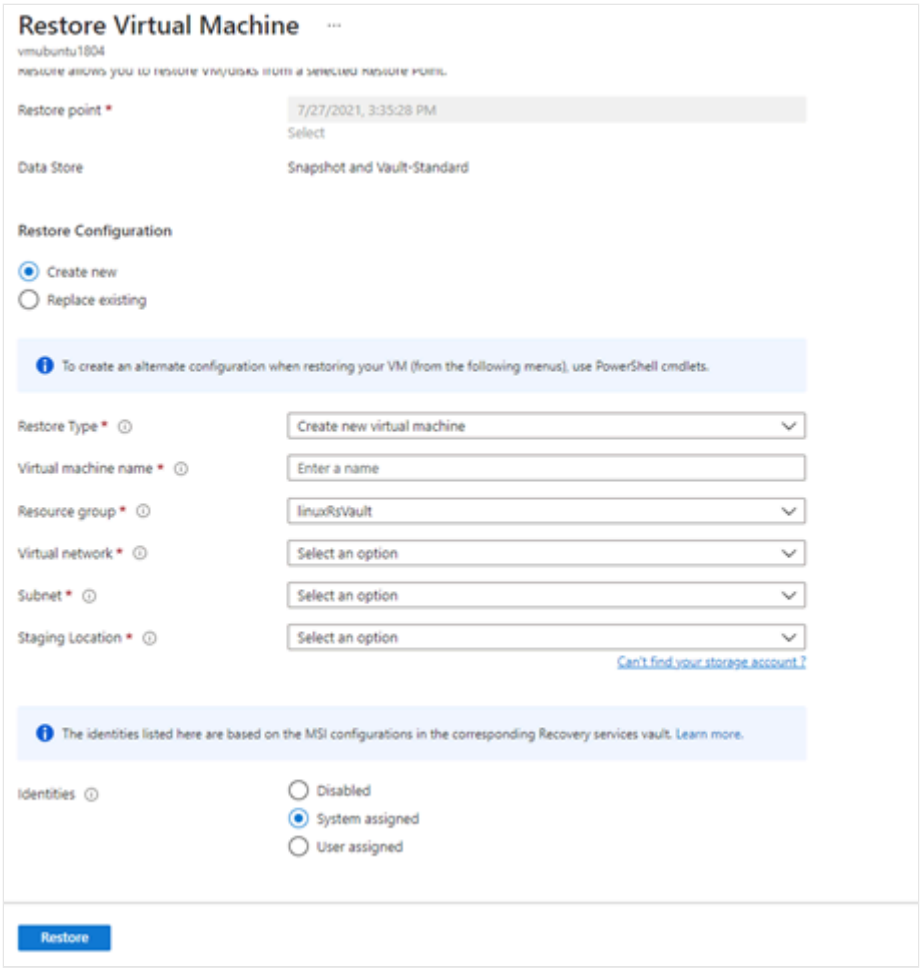
If you choose to select system-assigned or user-assigned managed identities, check for the below actions for managed identity on the target staging Storage Account and Resource Group.

JSON

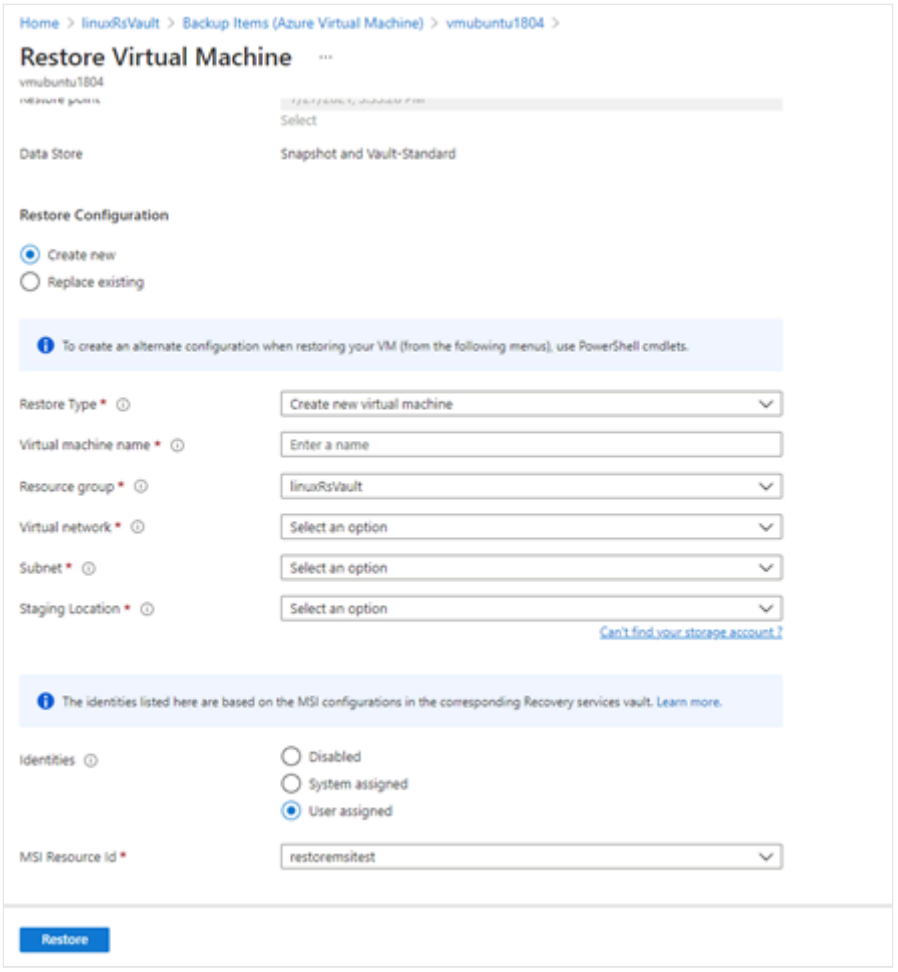
```
"permissions": [
            {
                "actions": [
                    "Microsoft.Authorization/*/read",
                    "Microsoft.Compute/disks/read",
                    "Microsoft.Compute/disks/write",
                    "Microsoft.Compute/disks/delete",
                    "Microsoft.Compute/disks/beginGetAccess/action",
                    "Microsoft.Compute/disks/endGetAccess/action",
                    "Microsoft.Compute/locations/diskOperations/read",
```

```
                    "Microsoft.Compute/virtualMachines/read",
                    "Microsoft.Compute/virtualMachines/write",
                    "Microsoft.Compute/virtualMachines/delete",
                    "Microsoft.Compute/virtualMachines/instanceView/read",
                    "Microsoft.Compute/virtualMachines/extensions/read",
                    "Microsoft.Compute/virtualMachines/extensions/write",
                    "Microsoft.Compute/virtualMachines/extensions/delete",
                    "Microsoft.Insights/alertRules/*",
                    "Microsoft.Network/locations/operationResults/read",
                    "Microsoft.Network/locations/operations/read",
                    "Microsoft.Network/locations/usages/read",
                    "Microsoft.Network/networkInterfaces/delete",
                    "Microsoft.Network/networkInterfaces/ipconfigurations/read",
                    "Microsoft.Network/networkInterfaces/join/action",
                    "Microsoft.Network/networkInterfaces/read",
                    "Microsoft.Network/networkInterfaces/write",
                    "Microsoft.Network/networkSecurityGroups/read",
                    "Microsoft.Network/networkSecurityGroups/securityRules/read",
                    "Microsoft.Network/publicIPAddresses/delete",
                    "Microsoft.Network/publicIPAddresses/join/action",
                    "Microsoft.Network/publicIPAddresses/read",
                    "Microsoft.Network/publicIPAddresses/write",
                    "Microsoft.Network/virtualNetworks/read",
                    "Microsoft.Network/virtualNetworks/subnets/join/action",
                    "Microsoft.Network/virtualNetworks/subnets/read",
                    "Microsoft.Resources/deployments/*",
                    "Microsoft.Resources/subscriptions/resourceGroups/read",
                    "Microsoft.Storage/checkNameAvailability/read",

"Microsoft.Storage/storageAccounts/blobServices/containers/delete",

"Microsoft.Storage/storageAccounts/blobServices/containers/read",

"Microsoft.Storage/storageAccounts/blobServices/containers/write",
                    "Microsoft.Storage/storageAccounts/read",
                    "Microsoft.Storage/storageAccounts/write"
                ],
                "notActions": [],
                "dataActions": [

"Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete",

"Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read",

"Microsoft.Storage/storageAccounts/blobServices/containers/blobs/write",

"Microsoft.Storage/storageAccounts/blobServices/containers/blobs/add/action"
                ],
```

Or, add the **VM restore operator** role assignment on the staging location (Storage Account) and target Resource Group for the successful restore operation.



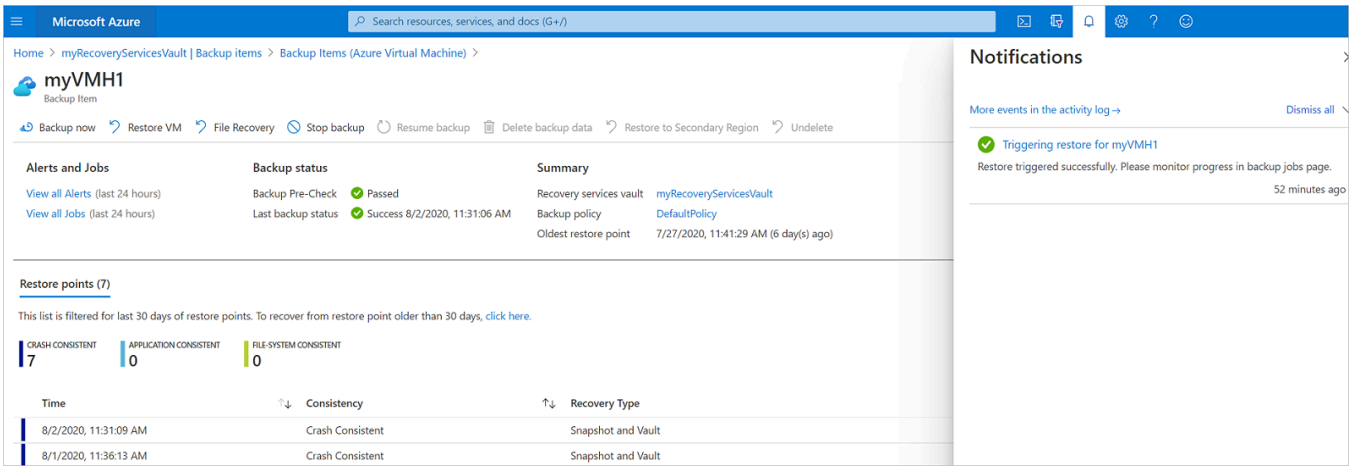
You can also select the user-managed identity by providing the input as their MSI Resource ID as provided in the figure below.



> **Note**
>
> The support is available for only managed VMs, and not supported for classic VMs and unmanaged VMs. For the storage accounts that are restricted with firewalls, system MSI is only supported.
>
> Cross Region Restore isn't supported with managed identities.
>
> Currently, this is available in all Azure public and national cloud regions.

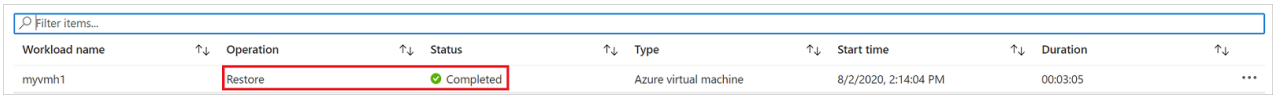
Track the restore operation

After you trigger the restore operation, the backup service creates a job for tracking. Azure Backup displays notifications about the job in the portal. If they aren't visible, select the **Notifications** symbol, and then select **More events in the activity log** to see the Restore Process Status.



Track restore as follows:

1. To view operations for the job, select the notifications hyperlink. Alternatively, in the vault, select **Backup jobs**, and then select the relevant VM.



2. To monitor restore progress, select any restore job with a status of **In-progress**. This displays the progress bar, which displays information about the restore progress:

    - **Estimated time of restore**: Initially provides the time taken to complete the restore operation. As the operation progresses, the time taken reduces and reaches zero when the restore operation finishes.
    - **Percentage of restore**. Shows the percentage of restore operation that's done.
    - **Number of bytes transferred**: If you're restoring by creating a new VM, it shows the bytes that were transferred against the total number of bytes to be transferred.

Post-restore steps

There are a few things to note after restoring a VM:

- Extensions present during the backup configuration are installed, but not enabled. If you see an issue, reinstall the extensions. In the case of disk replacement, reinstallation of extensions is not required.

- When you choose Create a new VM (or restore disks and then create a VM), the restored VM is a new resource. RBAC role assignments that were scoped to the original VM do not carry over; reassign roles on the new VM as needed.

- If the backed-up VM had a static IP address, the restored VM will have a dynamic IP address to avoid conflict. You can add a static IP address to the restored VM.

- A restored VM doesn't have an availability set. If you use the restore disk option, then you can specify an availability set when you create a VM from the disk using the provided template or PowerShell.
- If you use a cloud-init-based Linux distribution, such as Ubuntu, for security reasons the password is blocked after the restore. Use the `VMAccess` extension on the restored VM to reset the password. We recommend using SSH keys on these distributions, so you don't need to reset the password after the restore.
- If you're unable to access a VM once restored because the VM has a broken relationship with the domain controller, then follow the steps below to bring up the VM:
  - Attach OS disk as a data disk to a recovered VM.
  - Manually install VM agent if Azure Agent is found to be unresponsive by following this link.
  - Enable Serial Console access on VM to allow command-line access to VM

    Windows Command Prompt

    ```
    bcdedit /store <drive letter>:\boot\bcd /enum
    bcdedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd /set
    {bootmgr} displaybootmenu yes
    bcdedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd /set
    {bootmgr} timeout 5
    bcdedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd /set
    {bootmgr} bootems yes
    bcdedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd /ems {<<BOOT
    LOADER IDENTIFIER>>} ON
    bcdedit /store <VOLUME LETTER WHERE THE BCD FOLDER IS>:\boot\bcd /emssettings
    EMSPORT:1 EMSBAUDRATE:115200
    ```

  - When the VM is rebuilt use Azure portal to reset local administrator account and password
  - Use Serial console access and CMD to disjoin VM from domain

    Windows Command Prompt

    ```
    cmd /c "netdom remove <<MachineName>> /domain:<<DomainName>> /userD:
    <<DomainAdminhere>> /passwordD:<<PasswordHere>> /reboot:10 /Force"
    ```

- Once the VM is disjoined and restarted, you'll be able to successfully RDP to the VM with local admin credentials and rejoin VM back to domain successfully.

Backing up restored VMs

- If you restored a VM to the same resource group with the same name as the originally backed-up VM, backup continues on the VM after restore.
- If you restored the VM to a different resource group or you specified a different name for the restored VM, you need to set up backup for the restored VM.

Next steps

- If you experience difficulties during the restore process, review common issues and errors.
- After the VM is restored, learn about managing virtual machines

---

Last updated on 01/23/2026

Support matrix for Azure VM backups

Summarize this article for me

You can use the Azure Backup service to back up on-premises machines and workloads, along with Azure virtual machines (VMs). This article summarizes support settings and limitations when you back up Azure VMs by using Azure Backup.

Other support matrices include:

- General support matrix for Azure Backup
- Support matrix for Azure Backup servers and System Center Data Protection Manager (DPM) backup
- Support matrix for backup with the Microsoft Azure Recovery Services (MARS) agent

Supported scenarios

Here's how you can back up and restore Azure VMs by using the Azure Backup service.

Expand table

Scenario	Backup	Agent	Restore
Direct backup of Azure VMs	Back up the entire VM.	For application or file-system consistent backups, no additional agent is needed on the Azure VM. Azure Backup installs and uses an extension to the Azure VM agent that's running on the VM. You can also use agentless crash-consistent backups. Learn more.	Restore as follows: - **Create a basic VM**. This is useful if the VM has no special configuration, such as multiple IP addresses. - **Restore the VM disk**. Restore the disk. Then attach it to an existing VM, or create a new VM from the disk by using PowerShell. - **Replace the VM disk**. If a VM exists and it uses managed disks (unencrypted), you can restore a disk and use it to replace an existing disk on the VM. - **Restore specific files or folders**. You can restore files

Scenario	Backup	Agent	Restore
			or folders from a VM instead of restoring the entire VM.
Direct backup of Azure VMs (Windows only)	Back up specific files, folders, or volumes.	Install the Azure Recovery Services agent. You can run the MARS agent alongside the backup extension for the Azure VM agent to back up the VM at the file or folder level.	Restore specific files or folders.
Backup of Azure VMs to the backup server	Back up files, folders, or volumes; system state or bare metal files; and app data to System Center DPM or to Microsoft Azure Backup Server (MABS). DPM or MABS then backs up to the backup vault.	Install the DPM or MABS protection agent on the VM. The MARS agent is installed on DPM or MABS.	Restore files, folders, or volumes; system state or bare metal files; and app data.

Learn more about using a backup server and about support requirements.

Supported backup actions

Expand table

Action	Support
Back up a VM that's shut down or offline	Supported. Snapshot is crash-consistent only, not app consistent.
Back up disks after migrating to managed disks	Supported. Backup will continue to work. No action is required.
Back up managed disks after enabling a resource group lock	Not supported. Azure Backup can't delete the older restore points. Backups will start to fail when the limit of restore points is reached.

Action	Support
Modify backup policy for a VM	Supported. The VM will be backed up according to the schedule and retention settings in the new policy. If retention settings are extended, existing recovery points are marked and kept. If they're reduced, existing recovery points will be pruned in the next cleanup job and eventually deleted.
Cancel a backup job	Supported during the snapshot process. Not supported when the snapshot is being transferred to the vault.
Back up the VM to a different region or subscription	Not supported. For successful backup, virtual machines must be in the same subscription as the vault for backup.
Back up daily via the Azure VM extension	Four backups per day: one scheduled backup as defined in the backup policy, and three on-demand backups. To allow user retries in case of failed attempts, the hard limit for on-demand backups is set to nine attempts in a 24 hour UTC period.
Back up daily via the MARS agent	Three scheduled backups per day.
Back up daily via DPM or MABS	Two scheduled backups per day.
Back up monthly or yearly	Not supported when you're backing up with the Azure VM extension. Only daily and weekly are supported. You can set up the policy to retain daily or weekly backups for a monthly or yearly retention period.
Automatically adjust the clock	Not supported. Azure Backup doesn't automatically adjust for daylight saving time when you're backing up a VM. Modify the policy manually as needed.
Disable security features for hybrid backup	Not supported.
Back up Azure VMs with Azure Extended zones	Supported through Enhanced policy. Snapshot is crash-consistent only, you can also retain snapshots locally for instant restore.

Action	Support
Back up a VM whose machine time is changed	Not supported. If you change the machine time to a future date/time after enabling backup for that VM, even if the time change is reverted, successful backup isn't guaranteed.
Do multiple backups per day	Supported through **Enhanced policy**. For hourly backup, the minimum recovery point objective (RPO) is 4 hours and the maximum is 24 hours. You can set the backup schedule to 4, 6, 8, 12, and 24 hours, respectively. Note that the maximum limit of instant recovery point retention range depends on the number of snapshots you take per day. If the snapshot count is more (for example, every *4 hours* frequency in *24 hours* duration - *6* scheduled snapshots), then the maximum allowed days for retention reduces. However, if you choose lower RPO of *12* hours, the snapshot retention is increased to *30 days*. Learn about how to back up an Azure VM using Enhanced policy.
Back up a VM with a deprecated plan when the publisher has removed it from Azure Marketplace	Not supported. Backup is supported. Restore to existing/original VM is supported. Restore by creating a new VM will fail. If you've already configured backup for a VM with a deprecated virtual machine offer and encounter a restore error, see Troubleshoot backup errors with Azure VMs.
Back up VMs with docker (containers)	Not supported
Soft Delete for backups	Not Supported with Azure role-based access control (Azure RBAC) permissions at the resource group level. Requires Azure RBAC permissions at the subscription level, including the `Microsoft.DataProtection/locations/checkFeatureSupport/action` permission.

Operating system support (Windows)

The following table summarizes the supported operating systems when you're backing up Azure VMs running Windows.

Expand table

Scenario	OS support
Back up with the Azure VM agent extension for application-consistent backup	- Windows 11 client (64 bit only) - Windows 10 client (64 bit only) - Windows Server 2025 (Datacenter, Datacenter Core, and Standard) - Windows Server 2022 (Datacenter, Datacenter Core, and Standard) - Windows Server 2019 (Datacenter, Datacenter Core, and Standard) - Windows Server 2016 (Datacenter, Datacenter Core, and Standard) Note that *32 bit Operating Systems* aren't supported.
Back up Azure VM directly using agentless crash-consistent backup	Agentless crash-consistent backups are operating system agnostic.
Back up with the MARS agent	Supported operating systems
Back up with DPM or MABS	Supported operating systems for backup with MABS and DPM

Support for Linux backup

Here's what's supported if you want to back up Linux machines.

Expand table

Action	Support
Back up Linux Azure VMs with the Linux Azure VM agent	Supported for file-consistent backup. Also supported for app-consistent backup that uses custom scripts. During restore, you can create a new VM, restore a disk and use it to create a VM, or restore a disk and use it to replace a disk on an existing VM. You can also restore individual files and folders. *Note: Azure Backup requires the /var and /var/lib directories on Linux machines not to be mounted with the noexec flag. Certain backup*

Action	Support
	operations rely on executing helper scripts from /var/lib, and enforcing noexec prevents these components from running.
Back up Azure VM directly by using agentless crash-consistent backup	Agentless crash-consistent backups are operating system agnostic.
Back up Linux Azure VMs with the MARS agent	Not supported. The MARS agent can be installed only on Windows machines.
Back up Linux Azure VMs with DPM or MABS	Not supported.
Back up Linux Azure VMs with Docker mount points	Currently, Azure Backup doesn't support exclusion of Docker mount points because these are mounted at different paths every time.
Backup Linux Azure VMs with ZFS Pool Configuration	Not supported

Operating system support (Linux)

For Linux VM backups using the Linux Azure VM agent, Azure Backup supports the list of Linux distributions endorsed by Azure. Note the following:

- Agent-based VM backup doesn't support CoreOS Linux.
- Agent-based VM backup doesn't support 32-bit operating systems.
- Other bring-your-own Linux distributions might work as long as the Azure VM agent for Linux is available on the VM, and as long as Python is supported.
- Agent-based VM backup doesn't support a proxy-configured Linux VM if it doesn't have Python version 2.7 or later installed.
- Agentless crash-consistent backups are operating system agnostic and can be used to back up VMs whose operating system isn't supported for agent-based backups.
- Azure Backup doesn't support backing up Network File System (NFS) files that are mounted from storage, or from any other NFS server, to Linux or Windows machines. It backs up only disks that are locally attached to the VM.

Support matrix for managed pre and post scripts for Linux databases

Azure Backup provides the following support for customers to author their own pre and post scripts.

Expand table

Supported database	OS version	Database version
Oracle in Azure VMs	Oracle Linux	Oracle 12.x or later

Support for Confidential VM backup (preview)

The following table lists the supported scenarios for Confidential VM backup:

Expand table

Scenario	Supportability
Virtual Machine size	v6-series is supported. v5-series isn't supported.
Region availability	Supported in UAE North, Korea Central.
Key rotation for backups	When key rotation occurs on a confidential virtual machine, the keys for the VM disks, related restore points, and snapshots update automatically. **Known issue:** The key rotation in this *preview release* might have performance issues or fail in the following scenarios: - More than 40 disks are attached to one DES when (only) restore points are associated with these disks. - If you also directly create disk snapshots outside of Azure backup for these disks connected to the same DES, this lowers the safe threshold of 40 disks to DES mapping. **Recommendation**: Keep the number of disks connected to each DES to a minimum until the issue is resolved.
Backup capabilities	- You can backup Confidential VMs with OS disk encryption only. - Backup and restore fail if the CVM v2 opt-out feature flag is enabled for your subscription. - Multi-disk crash consistent backup is unsupported. - Cross Region Restore is currently unsupported as CVM v6 VM size isn't generally available in Azure paired regions.

Support for agentless multi-disk crash-consistent VM backup

The following table lists the supported scenarios for agentless multi-disk crash-consistent Azure Virtual Machine (VM) backup:

Expand table

Scenario	Supportability
Region availability	Supported in all Azure public regions.
Backup policy type	Agentless crash-consistent backup is supported only with Enhanced Policy.
VM type and size	- Supported for VM sizes that are premium storage capable (VM size that include "s" in their name) such as `DSv2`. - Trusted Launch VMs are supported. - VMs with Ultra-disks, Premium v2 SSD, Ephemeral OS disks, Shared disks, and Write Accelerated disks aren't supported. - VM sizes using Azure Boost aren't supported - Intel V6 and above (Dsv6-series, Edsv6-series, Esv6-series, and more) and AMD V7 and above (Dasv7-series, Dadsv7-series, Easv7-series, Faldsv7-series, and more)
Pre/post script	Not applicable Pre/Post scripts are used for app-consistent Linux VM backup.

Learn more about Agentless multi-disk crash-consistent VM backup.

Backup frequency and retention

Expand table

Setting	Limits
Maximum recovery points per protected instance (machine or workload)	9999.
Maximum expiry time for a recovery point	No limit (99 years).
Maximum backup frequency to a vault (Azure VM extension)	Once a day.
Maximum backup frequency to a vault (MARS agent)	Three backups per day.
Maximum backup frequency to DPM or MABS	Every 15 minutes for SQL Server. Once an hour for other workloads.
Recovery point retention	Daily, weekly, monthly, and yearly.

Setting	Limits
Maximum retention period	Depends on backup frequency.
Recovery points on DPM or MABS disk	64 for file servers, and 448 for app servers. Tape recovery points are unlimited for on-premises DPM.

Supported restore methods

Expand table

Restore option	Details
Create a new VM	This option quickly creates and gets a basic VM up and running from a restore point. You can specify a name for the VM, select the resource group and virtual network in which it will be placed, and specify a storage account for the restored VM. The new VM must be created in the same region as the source VM.
Restore disk	This option restores a VM disk, which can you can then use to create a new VM. Azure Backup provides a template to help you customize and create a VM. The restore job generates a template that you can download and use to specify custom VM settings and create a VM. The disks are copied to the resource group that you specify. Alternatively, you can attach the disk to an existing VM, or create a new VM by using PowerShell. This option is useful if you want to customize the VM, add configuration settings that weren't there at the time of backup, or add settings that must be configured via the template or PowerShell.
Replace existing	You can restore a disk and use it to replace a disk on the existing VM. The current VM must exist. If it has been deleted, you can't use this option. Azure Backup takes a snapshot of the existing VM before replacing the disk, and it stores the snapshot in the staging location that you specify. Existing disks connected to the VM are replaced with the selected restore point. The snapshot is copied to the vault and retained in accordance with the retention policy.

Restore option	Details
	After the replace-disk operation, the original disk is retained in the resource group. You can choose to manually delete the original disks if they aren't needed. This option is supported for unencrypted managed VMs and for VMs created from custom images. It's not supported for unmanaged disks and VMs, classic VMs, and generalized VMs. If the restore point has more or fewer disks than the current VM, the number of disks in the restore point will only reflect the VM configuration. This option is also supported for VMs with linked resources, like user-assigned managed identity and Azure Key Vault.
Cross Region (secondary region)	You can use cross-region restore to restore Azure VMs in the secondary region, which is an Azure paired region. You can restore all the Azure VMs for the selected recovery point if the backup is done in the secondary region. This feature is available for the following options: - Create a VM - Restore disks We don't currently support the Replace existing disks option. Backup admins and app admins have permissions to perform the restore operation on a secondary region.
Cross Subscription	Allowed only if the Cross Subscription Restore property is enabled for your Recovery Services vault. You can restore Azure Virtual Machines or disks to a different subscription within the same tenant as the source subscription (as per the Azure RBAC capabilities) from restore points. This feature is available for the following options: - Create a VM - Restore disks Cross Subscription Restore is unsupported for: - Snapshots tier recovery points. It's also unsupported for unmanaged VMs and VMs with disks having Azure Encryptions (ADE). - VMs with CMK encrypted disks.
Cross Zonal Restore	You can use cross-zonal restore to restore Azure zone-pinned VMs in available zones. You can restore Azure VMs or disks to different zones (one of the Azure RBAC capabilities) from restore points. Note that when you select a zone to restore, it selects the logical zone (and not the physical zone) as per the Azure subscription you

Restore option	Details
	will use to restore to. This feature is available for the following options: - Create a VM - Restore disks Cross-zonal restore is unsupported for snapshots of restore points. Cross-zonal restore doesn't support restoring encrypted Azure VMs directly. If you perform a restore, the restored VM uses **SSE encryption** by default. You must manually reapply **CMK encryption** to the restored VM after the restore process is complete.

Support for file-level restore

Expand table

Restore	Supported
Restore files across operating systems	You can restore files on any machine that has the same OS as the backed-up VM, or a compatible OS. See the compatible OS table.
Restore files from encrypted VMs	Not supported.
Restore files from network-restricted storage accounts	Not supported.
Restore files on VMs by using Windows Storage Spaces	Not supported.
Restore files on a Linux VM by using LVM or RAID arrays	Not supported on the same VM. Restore on a compatible VM.
Restore files with special network settings	Not supported on the same VM. Restore on a compatible VM.
Restore files from a shared disk, temporary drive, deduplicated disk, Ultra disk, Premium SSD v2 disk, or disk with a write accelerator enabled	Not supported. See Azure VM storage support.

Support for VM management

The following table summarizes support for backup during VM management tasks, such as adding or replacing VM disks.

Expand table

Restore	Supported
Restore across a subscription	Cross-subscription restore is now supported in Azure VMs.
Restore across a region	Supported.
Restore across a zone	Cross-zonal restore is now supported in Azure VMs.
Restore to an existing VM	Use the replace-disk option.
Restore a disk with a storage account enabled for Azure Storage service-side encryption (SSE)	Not supported. Restore to an account that doesn't have SSE enabled.
Restore to mixed storage accounts	Not supported. Based on the storage account type, all restored disks will be either premium or standard, and not mixed.
Restore a VM directly to an availability set	For managed disks, you can restore the disk and use the availability set option in the template. Not supported for unmanaged disks. For unmanaged disks, restore the disk, and then create a VM in the availability set.
Restore backup of unmanaged VMs after upgrading to a managed VM	Supported. You can restore disks and then create a managed VM.
Restore a VM to a restore point before the VM was migrated to managed disks	Supported. You restore to unmanaged disks (default), convert the restored disks to managed disks, and create a VM with the managed disks.
Restore a VM that has been deleted	Supported. You can restore the VM from a recovery point.
Restore a domain controller VM	Supported. For details, see Restore domain controller VMs.
Restore a VM in a different virtual network	Supported.

Restore	Supported
	The virtual network must be in the same subscription and region.

VM compute support

Expand table

Compute	Support
Back up VMs of a certain size	You can back up any Azure VM that has at least two CPU cores and 1 GB of RAM. Learn more.
Back up VMs in availability sets	Supported. You can't restore a VM in an availability set by using the option to quickly create a VM. Instead, when you restore the VM, restore the disk and use it to deploy a VM, or restore a disk and use it to replace an existing disk.
Back up VMs that are deployed with Azure Hybrid Benefit	Supported.
Back up VMs that are deployed from Azure Marketplace (published by Microsoft or a third party)	Supported. The VMs must be running a supported operating system. When you're recovering files on the VM, you can restore only to a compatible OS (not an earlier or later OS). We don't restore Azure Marketplace VMs backed as VMs, because these need purchase information. They're restored only as disks.
Back up VMs that are deployed from a custom image (third-party)	Supported. The VMs must be running a supported operating system. When you're recovering files on VMs, you can restore only to a compatible OS (not an earlier or later OS).
Back up VMs that are migrated to Azure	Supported. To back up a VM, make sure that the VM agent is installed on the migrated machine.

Compute	Support
Back up multiple VMs and provide consistency	Azure Backup doesn't provide data and application consistency across multiple VMs.
Back up a VM with diagnostic settings	Not supported. If the restore of the Azure VM with diagnostic settings is triggered via the Create new option, the restore fails.
Restore zone-pinned VMs	Supported (where availability zones are available). Azure Backup now supports restoring Azure VMs to a any availability zones other than the zone that's pinned in VMs. This support enables you to restore VMs when the primary zone is unavailable.
Back up Gen2 VMs	Supported. Azure Backup supports backup and restore of Gen2 VMs. When these VMs are restored from a recovery point, they're restored as Gen2 VMs.
Back up Azure VMs with locks	Supported for managed VMs. Not supported for unmanaged VMs.
Restore spot VMs	Not supported. Azure Backup restores spot VMs as regular Azure VMs.
Restore VMs in an Azure dedicated host	Supported. When you're restoring an Azure VM through the Create new option, the VM can't be restored in the dedicated host, even when the restore is successful. To achieve this, we recommend that you restore as disks. While you're restoring as disks by using the template, create a VM in a dedicated host, and then attach the disks. This is not applicable in a secondary region while you're performing cross-region restore.
Configure standalone Azure VMs in Windows Storage Spaces	Not supported.
Configure standalone Azure VMs in Windows Storage Spaces direct	Not supported.
Restore Virtual Machine Scale Sets	Supported for the flexible orchestration model to back up and restore a single Azure VM.

Compute	Support
Restore with managed identities	Supported for managed Azure VMs. Not supported for classic and unmanaged Azure VMs. Cross-region restore isn't supported with managed identities. Currently, this is available in all Azure public and national cloud regions. Learn more.
Back up trusted launch VMs	Supported via Enhanced policy and Standard policy. **Feature details**: - Backup of a VM with Data disks having Security Type 'TrustedLaunch' or 'ConfidentialVM' is not supported. This can happen if you attach the OS disk from a Trusted Launch as a Data disk to another VM. - Item-level restore is supported for the applicable scenarios. **Backup support via Enhanced policy**: - Azure portal, PowerShell, and REST API clients support trusted launch VM backup with Enhanced policy. - The Azure portal allows you to enable backup through a Recovery Services vault, the VM management pane, and the VM creation pane. - Resiliency supports backup, alerts, and monitoring for trusted launch VMs. **Backup support via Standard policy**: - CLI (version 2.73.0 and later), PowerShell (version Az 14.0.0 and later), and REST API (version 2025-01-01 and later) only support trusted launch VM backup with Standard policy. - Trusted Launch virtual machines with standard policy use managed disk snapshots for Instant Restore. In this scenario, you incur Snapshot storage cost same as that of Enhanced policy. Learn more. Note that migration of an existing Gen2 VM (protected by Azure Backup Standard policy) to Trusted Launch VM requires first switching to Enhanced policy.
Back up confidential VMs	Supported (in preview). Learn more about the supported scenarios.

VM storage support

Expand table

Component	Support
Azure VM data disks	Support for backup of Azure VMs is up to 32 disks. Support for backup of Azure VMs with unmanaged disks or classic VMs is deprecated. *Note that backup support for disks in Azure Classic Storage is no longer available. To ensure continued functionality, migrate these disks to Azure Resource Manager (ARM) Storage, as Classic Storage accounts don't support ARM-based Virtual Machines (VMs). Also, backup for all Classic VMs is set to be deprecated on September 30, 2025. To prevent disruptions, migrate Classic VMs and associated disks to ARM Storage before this timeline.*
Data disk size	Individual disk size can be up to 64 TB and a maximum of 512 TB combined for all disks in a VM.
Storage type	Standard HDD, Standard SSD, Premium SSD, Premium SSD v2 and Ultra disk. Backup and restore of zone-redundant storage disks is supported.
Managed disks	Supported.
Encrypted disks	Supported. Azure VMs enabled with Azure Disk Encryption can be backed up (with or without the Microsoft Entra app). Encrypted VMs can't be recovered at the file or folder level. You must recover the entire VM. You can enable encryption on VMs that Azure Backup is already protecting. You can back up and restore disks encrypted via platform-managed keys or customer-managed keys. You can also assign a disk-encryption set while restoring in the same region. That is, providing a disk-encryption set while performing cross-region restore is currently not supported. However, you can assign the disk-encryption set to the restored disk after the restore is complete.
Disks with a write accelerator enabled	Azure VMs with disk backup for a write accelerator became available in all Azure public regions on May 18, 2022. If disk backup for a write accelerator is not required as part of VM backup, you can choose to remove it by using the selective disk feature. **Important**

Component	Support
	Virtual machines with write accelerator disks need internet connectivity for a successful backup, even though those disks are excluded from the backup.
Disks enabled for access with a private endpoint	Supported.
Disks with both public and private access disabled	Supported.
Backup and restore of deduplicated VMs or disks	Azure Backup doesn't support deduplication. For more information, see this article. Azure Backup doesn't deduplicate across VMs in the Recovery Services vault. If there are VMs in a deduplication state during restore, the files can't be restored because the vault doesn't understand the format. However, you can successfully perform the full VM restore.
Adding a disk to a protected VM	Supported.
Resizing a disk on a protected VM	Supported.
Shared storage	Backing up VMs by using Cluster Shared Volumes (CSV) or Scale-Out File Server isn't supported. CSV writers are likely to fail during backup. On restore, disks that contain CSV volumes might not come up.
Shared disks	Not supported. - You can exclude shared disk with Enhanced policy and backup the other supported disks in the VM. - You can use S2D to create a shared disk or standalone volumes by combining capacities from disks in different VMs. Azure Backup doesn't support backup of a shared volume (between VMs for database cluster or cluster Configuration) created using S2D.
Ultra disks	Supported with Enhanced policy.Learn about the disk considerations for Azure VM. - Configuration of Ultra disk protection is supported via Recovery Services vault and via virtual machine blade. - File-level restore is currently not supported for machines using Ultra disks.

Component	Support
	- GRS vaults and Cross-Region Restore are currently supported in the following regions for machines using Ultra Disks: South Central US, Brazil South, Canada East, Canada Central, East US2, South East Asia, West US, Central US, Korea South, Korea Central, South Central US, West Europe, North Central US, East Asia, USGov Texas, USGov Arizona, USGov Texas, West US2, North Europe, East US, West Central US, East US.
Premium SSD v2	Supported with Enhanced policy. Learn about the disk considerations for Azure VM. - Configuration of Premium SSD v2 disk protection is supported via Recovery Services vault and via virtual machine blade. - File-level restore is currently not supported for machines using Premium SSD v2 disks. - GRS vaults and Cross-Region Restore are currently supported in the following regions for machines using Premium SSDv2 Disks: Brazil South, Central US, East Asia, East US, East US2, North Central US, North Europe, South Central US, South East Asia, UK South, UK West, West Europe, West US, West US3, Norway West, Norway East, Japan West, Japan East.
Temporary disks	Azure Backup doesn't back up temporary disks.
NVMe/ephemeral disks	Supported.
Resilient File System (ReFS) restore	Supported. Volume Shadow Copy Service (VSS) supports app-consistent backups on ReFS.
Dynamic disk with spanned or striped volumes	Supported, unless you enable the selective disk feature on an Azure VM.
VMs with encryption at host	Supported
Disks with enabled Data Access with Microsoft Entra authentication for disk upload/download	Not Supported
Storage Replicas	Not supported
Performance-plus disks	Recovering from snapshot tier (instant restore) will recover disks as performance-plus disks. Recovering from vault does not respect the performance-plus flag and will restore disks as normal disks.

VM network support

Expand table

Component	Support
Number of network interfaces (NICs)	Supported up to the maximum number for a specific Azure VM size. NICs are created when the VM is created during the restore process. The number of NICs on the restored VM mirrors the number of NICs on the VM at the time of the backup. The number of NICs on the restore VM depends on the NICs in the restore point, which in turn depends on the NICs in the source VM at the time of backup.
External or internal load balancer	Supported. Learn more about restoring VMs with special network settings.
Multiple reserved IP addresses	Supported. Learn more about restoring VMs with special network settings.
VMs with multiple network adapters	Supported. Learn more about restoring VMs with special network settings.
VMs with public IP addresses	Supported. Associate an existing public IP address with the NIC, or create an address and associate it with the NIC after the restore is done.
Network security group (NSG) on a NIC or subnet	Supported.
Static IP address	Not supported. A new VM that's created from a restore point is assigned a dynamic IP address. For classic VMs, you can't back up a VM with a reserved IP address and no defined endpoint.
Dynamic IP address	Supported. If the NIC on the source VM uses dynamic IP addressing, the NIC on the restored VM will also use it by default.
Azure Traffic Manager	Supported. If the backed-up VM is in Traffic Manager, manually add the restored VM to the same Traffic Manager instance.

Component	Support
Azure DNS	Supported.
Custom DNS	Supported.
Outbound connectivity via HTTP proxy	Supported. An authenticated proxy isn't supported.
Virtual network service endpoints	Supported. Storage account settings for a firewall and a virtual network should allow access from all networks.

Support for VM security and encryption

Azure Backup supports encryption for in-transit and at-rest data.

For network traffic to Azure:

- The Backup traffic from servers to the Recovery Services vault is encrypted via Advanced Encryption Standard 256.
- Backup data is sent over a secure HTTPS link.
- Backup data is stored in the Recovery Services vault in encrypted form.
- Only you have the encryption key to unlock this data. Microsoft can't decrypt the backup data at any point.

  > ⚠ **Warning**
  >
  > After you set up the vault, only you have access to the encryption key. Microsoft never maintains a copy and doesn't have access to the key. If the key is misplaced, Microsoft can't recover the backup data.

For data security:

- When you're backing up Azure VMs, you need to set up encryption *within* the virtual machine.
- Azure Backup supports Azure Disk Encryption, which uses BitLocker on virtual machines running Windows and uses *dm-crypt* on Linux virtual machines.
- On the back end, Azure Backup uses Azure Storage service-side encryption to help protect data at rest.

Expand table

Machine	In transit	At rest
On-premises Windows machines without DPM or MABS	✔	✔
Azure VMs	✔	✔
On-premises or Azure VMs with DPM	✔	✔
On-premises or Azure VMs with MABS	✔	✔

VM compression support

Azure Backup supports the compression of backup traffic. Note the following:

- For Azure VMs, the VM extension reads the data directly from the Azure storage account over the storage network. It isn't necessary to compress this traffic.
- If you're using DPM or MABS, you can save bandwidth by compressing the data before it's backed up.

Expand table

Machine	Compress to DPM/MABS (TCP)	Compress to vault (HTTPS)
On-premises Windows machines without DPM or MABS	Not applicable	✔
Azure VMs	Not applicable	Not applicable
On-premises or Azure VMs with DPM	✔	✔
On-premises or Azure VMs with MABS	✔	✔

Next steps

- Back up Azure VMs.
- Back up Windows machines directly, without a backup server.
- Set up MABS for backup to Azure, and then back up workloads to MABS.
- Set up DPM for backup to Azure, and then back up workloads to DPM.

Last updated on 01/28/2026

Estimate and understand Azure Backup pricing

09/19/2025

This article describes how to estimate and understand the costs associated with using Azure Backup. You learn about the available pricing tools, key parameters that affect backup costs, and how to use the Azure Backup pricing estimator to budget for different workloads and scenarios.

To learn about Azure Backup pricing, visit the Azure Backup pricing page.

Download detailed estimates for Azure Backup pricing

If you're looking to estimate your costs for budgeting or cost comparison purposes, download the detailed Azure Backup pricing estimator.

What does the estimator contain?

The Azure Backup cost estimator sheet has an option for you to estimate all possible workloads you're looking to back up using Azure Backup. These workloads include:

- Azure VMs
- On-premises servers
- SQL in Azure VMs
- SAP HANA in Azure VMs
- Azure files shares

Estimate costs for backing up Azure VMs or on-premises servers

To estimate the costs of backing up Azure VMs or on-premises servers using Azure Backup, you'll need the following parameters:

- Size of the VMs or on-premises servers that you're trying to back up
  - Enter the **used size** of disks or servers required to be backed up
- Number of servers with that size

- What is the expected amount of data churn on these servers?
  Churn refers to the amount of change in data. For example, if you had a VM with 200 GB of data to be backed up and 10 GB of it changes every day, the daily churn is 5%.
  - Higher churn will mean that you back up more data
  - Pick **Low** or **Moderate** for file servers and **High** if you're running databases
  - If you know your **churn%**, you can use the **Enter your own%** option
- Choose the backup policy
  - How long do you expect to retain **Daily** backups? (in days)
  - How long do you expect to retain **Weekly** backups? (in weeks)
  - How long do you expect to retain **Monthly** backups? (in months)
  - How long do you expect to retain **Yearly** backups? (in years)
  - How long do you expect to retain **Instant restore snapshots**? (1-5 days)
    - This option lets you restore from as far back as seven days in a quick manner using snapshots stored on disks.
- **Optional** – Selective Disk backup
  - If you're using the **Selective Disk Backup** option while backing up Azure VMs, choose the **Exclude Disk** option and enter the percentage of disks excluded from backup in terms of size. For example, if you have a VM connected to three disks with 200 GB used in each disk and if you want to exclude two of them from backing up, enter 66.7%.
- **Optional** – Backup Storage Redundancy
  - This indicates the redundancy of the Storage Account your backup data goes into. We recommend using **GRS** for the highest availability. Since it ensures that a copy of your backup data is kept in a different region, it helps you meet multiple compliance standards. Change the redundancy to **LRS** if you're backing up development or test environments that don't need an enterprise-level backup. Select the **RAGRS** option in the sheet if you want to understand costs when Cross-Region Restore is enabled for your backups.
- **Optional** – Modify regional pricing or apply discounted rates
  - If you want to check your estimates for a different region or discounted rates, select **Yes** for the **Try estimates for a different region?** option and enter the rates with which you want to run the estimates.

Estimate costs for backing up SQL servers in Azure VMs

To estimate the costs of backing up SQL servers running in Azure VMs using Azure Backup, you'll need the following parameters:

- Size of the SQL servers that you're trying to back up
- Number of SQL servers with the above size
- What is the expected compression for your SQL servers' backup data?
    - Most Azure Backup customers see that the backup data has 80% compression compared to the SQL server size when the SQL compression is **enabled**.
    - If you expect to see a different compression, enter the number in this field
- What is the expected size of log backups?
    - The % indicates daily log size as a % of the SQL server size
- What is the expected amount of daily data churn on these servers?
    - Typically, databases have **High** churn
    - If you know your **churn%**, you can use the **Enter your own%** option
- Choose the backup policy
    - Backup Type
        - The most effective policy you can choose is **Daily differentials** with weekly/monthly/yearly full backups. Azure Backup can restore from differentials through single-click as well.
        - You can also choose to have a policy with daily/weekly/monthly/yearly full backups. This option will consume slightly more storage than the first option.
    - How long do you expect to retain **log** backups? (in days) [7-35]
    - How long do you expect to retain **Daily** backups? (in days)
    - How long do you expect to retain **Weekly** backups? (in weeks)
    - How long do you expect to retain **Monthly** backups? (in months)
    - How long do you expect to retain **Yearly** backups? (in years)

- **Optional** – Backup Storage Redundancy
  - This indicates the redundancy of the Storage Account your backup data goes into. We recommend using **GRS** for the highest availability. Since it ensures that a copy of your backup data is kept in a different region, it helps you meet multiple compliance standards. Change the redundancy to **LRS** if you're backing up development or test environments that don't need an enterprise-level backup.

- **Optional** – Modify regional pricing or apply discounted rates
  - If you want to check your estimates for a different region or discounted rates, select **Yes** for the **Try estimates for a different region?** option and enter the rates with which you want to run the estimates.

Estimate costs for backing up SAP HANA servers in Azure VMs

To estimate the costs of backing up SAP HANA servers running in Azure VMs using Azure Backup, you'll need the following parameters:

- Total size of the SAP HANA databases that you're trying to back up. This should be the sum of full backup size of each of the databases, as reported by SAP HANA.

- Number of SAP HANA servers with the above size

- What is the expected size of log backups?
  - The % indicates average daily log size as a % of the total size of SAP HANA databases that you're backing up on the SAP HANA server

- What is the expected amount of daily data churn on these servers?
  - The % indicates average daily churn size as a % of the total size of SAP HANA databases that you're backing up on the SAP HANA server
  - Typically, databases have **High** churn
  - If you know your **churn%**, you can use the **Enter your own%** option

- Choose the backup policy
  - Backup Type
    - The most effective policy you can choose is **Daily differentials** with **weekly/monthly/yearly** full backups. Azure Backup can restore from differentials through single-click as well.
    - You can also choose to have a policy with **daily/weekly/monthly/yearly** full backups. This option will consume slightly more storage than the first option.
  - How long do you expect to retain **log** backups? (in days) [7-35]
  - How long do you expect to retain **Daily** backups? (in days)

    - How long do you expect to retain **Weekly** backups? (in weeks)
    - How long do you expect to retain **Monthly** backups? (in months)
    - How long do you expect to retain **Yearly** backups? (in years)

- **Optional** – Backup Storage Redundancy
    - This indicates the redundancy of the Storage Account your backup data goes into. We recommend using **GRS** for the highest availability. Since it ensures that a copy of your backup data is kept in a different region, it helps you meet multiple compliance standards. Change the redundancy to **LRS** if you're backing up development or test environments that don't need an enterprise-level backup.

- **Optional** – Modify regional pricing or apply discounted rates
    - If you want to check your estimates for a different region or discounted rates, select **Yes** for the **Try estimates for a different region?** option and enter the rates with which you want to run the estimates.

Estimate costs for backing up Azure file shares

To estimate the costs of backing up Azure file shares using the snapshot-based backup solution offered by Azure Backup, you'll need the following parameters:

- Size (**in GB**) of the file shares that you want to back up.

- If you want to back up file shares spread across multiple storage accounts, specify the number of storage accounts hosting the file shares with the above size.

- Expected amount of data churn on the file shares that you want to back up.
  Churn refers to the amount of change in data and it directly impacts the snapshot storage size. For example, if you have a file share with 200 GB of data to be backed up, and 10 GB of it changes every day, the daily churn is 5%.
    - Higher churn means the amount of data change in the file share contents every day is high, and so incremental snapshot (capturing only the data changes) size would also be more.
    - Select Low (1%), Moderate (3%), or High (5%) based on your file share characteristics and usage.
    - If you know the exact **churn%** for your file share, you can select the **Enter your own%** option from the drop-down. Specify the values (in %) for daily, weekly, monthly, and yearly churn.

- Type of storage account (standard or premium) and the storage redundancy setting of the storage account hosting the backed-up file share.
  In the current backup solution for Azure file shares, snapshots are stored in the same

storage account as the backed-up file share. So the storage cost associated with snapshots is billed as part of your Azure files bill, based on the snapshot pricing for the account type and redundancy setting of the storage account hosting the backed-up file share and snapshots.

- Retention for different backups
  - How long do you expect to retain “Daily” backups? (in days)
  - How long do you expect to retain “Weekly” backups? (in weeks)
  - How long do you expect to retain “Monthly” backups? (in months)
  - How long do you expect to retain “Yearly” backups? (in years)

  Refer to the Azure File share support matrix for the maximum supported retention values in each category.

- **Optional** – Modify regional pricing or apply discounted rates.
  - The default values set for snapshot storage cost per GB and protected instance cost in the estimator are for the East US region. If you want to check your estimates for a different region or discounted rates, select **Yes** for the **Try estimates for a different region?** option, and enter the rates with which you want to run the estimates.

Related content

- What is the Azure Backup service?.
- FAQ: Understand the Azure Backup pricing calculator.

Frequently asked questions-Back up Azure VMs

This article answers common questions about backing up Azure VMs with the Azure Backup service.

Backup

Which VM images can be enabled for backup when I create them?

When you create a VM, you can enable backup for VMs running supported operating systems.

Why is the Initial backup taking longer time to complete?

Initial backup is always a full backup and its duration depends on the size of the data and when the backup is processed.

To improve backup performance, see, backup best practices; Backup considerations and Backup Performance

Although the total backup time for incremental backups is less than 24 hours that might not be the case for the first backup.

Is the backup cost included in the VM cost?

No. Backup costs are separate from a VM's costs. Learn more about Azure Backup pricing.

Which permissions are required to enable backup for a VM?

If you're a VM contributor, you can enable backup on the VM. If you're using a custom role, you need the following permissions to enable backup on the VM:

- Microsoft.RecoveryServices/Vaults/write

- Microsoft.RecoveryServices/Vaults/read
- Microsoft.RecoveryServices/locations/*
- Microsoft.RecoveryServices/Vaults/backupFabrics/protectionContainers/protectedItems/*/read
- Microsoft.RecoveryServices/Vaults/backupFabrics/protectionContainers/protectedItems/read
- Microsoft.RecoveryServices/Vaults/backupFabrics/protectionContainers/protectedItems/write
- Microsoft.RecoveryServices/Vaults/backupFabrics/backupProtectionIntent/write
- Microsoft.RecoveryServices/Vaults/backupPolicies/read
- Microsoft.RecoveryServices/Vaults/backupPolicies/write

If your Recovery Services vault and VM have different resource groups, make sure you have write permissions in the resource group for the Recovery Services vault.

Does an on-demand backup job use the same retention schedule as scheduled backups?

No. Specify the retention range for an on-demand backup job. By default, it's retained for 30 days when triggered from the portal.

I recently enabled Azure Disk Encryption on some VMs. Will my backups continue to work?

Provide permissions for Azure Backup to access the Key Vault. Specify the permissions in PowerShell as described in the **Enable backup** section in the Azure Backup PowerShell documentation.

I migrated VM disks to managed disks. Will my backups continue to work?

Yes, backups work seamlessly. There's no need to reconfigure anything.

Why can't I see my VM in the Configure Backup wizard?

The wizard only lists VMs in the same region as the vault, and that aren't already being backed up.

My VM is shut down. Will an on-demand or a scheduled backup work?

Yes. Backups run when a machine is shut down. The recovery point is marked as crash consistent.

Can I cancel an in-progress backup job?

Yes. You can cancel the backup job in a **Taking snapshot** state. You can't cancel a job if data transfer from the snapshot is in progress.

I enabled a lock on the resource group created by Azure Backup Service (for example, `AzureBackupRG_<geo>_<number>`). Will my backups continue to work?

If you lock the resource group created by the Azure Backup Service, backups start to fail. Remove the lock, and clear the restore point collection from that resource group to make the future backups successful. Follow these steps to remove the restore point collection.

I have a lock at the resource group level that contains all the resources related to my virtual machine. Will my backup work?

Azure Backup creates a separate resource group in the format `AzureBackupRG_<geo>_<number>` to store ResourcePointCollections objects. Since this resource group is service owned, locking it will cause backups to fail. Locks can be only applied to customer-created resource groups.

Does Azure Backup support Standard SSD-managed disks?

Yes, Azure Backup supports Standard SSD managed disks.

Can we back up a VM with a Write Accelerator (WA)-enabled disk?

Snapshots can be taken on only data disks that are WA enabled and not OS disks. So, only data disks that are WA enabled can be protected.

I have a VM with Write Accelerator (WA) disks and SAP HANA installed. How do I back up?

Azure Backup can back up the WA-enabled data disk. However, the backup won't provide database consistency.

Azure Backup provides a streaming backup solution for SAP HANA databases with an RPO of 15 minutes. It's Backint certified by SAP to provide a native backup support leveraging SAP HANA's native APIs. Learn more about backing up SAP HANA databases in Azure VMs.

What is the maximum delay I can expect in backup start time from the scheduled backup time I have set in my VM backup policy?

The scheduled backup is triggered within 2 hours of the scheduled backup time. For example, If 100 VMs have their backup start time scheduled at 2:00 AM, then by 4:00 AM at the latest all the 100 VMs have their backup job in progress. If scheduled backups have been paused because of an outage and resumed or retried, then the backup can start outside of this scheduled two-hour window.

What is the minimum allowed retention range for a daily backup point?

Azure Virtual Machine backup policy supports a minimum retention range from seven days up to 9999 days. By default, backup of VMs are kept for *7 days* in snapshot and *180 days* in vault. Any modification to an existing VM backup policy with less than seven days will require an update to meet the minimum retention range of seven days.

What happens if I change the case of the name of my VM or my VM resource group?

If you change the case (to upper or lower) of your VM or VM resource group, the case of the backup item name won't change; this is expected Azure Backup behavior. The case change won't appear in the backup item, but is updated at the backend.

Can I back up or restore selective disks attached to a VM?

Azure Backup now supports selective disk backup and restore using the Azure Virtual Machine backup solution. For more information, see Selective disk backup and restore for Azure VMs.

Are managed identities preserved if a tenant change occurs during backup?

If tenant changes occur, you're required to disable and re-enable managed identities to make backups work again.

Does Azure Backup support backing up NFS files mounted from storage?

Azure Backup doesn't support backing up NFS files that are mounted from storage, or from any other NFS server, to Linux or Windows machines. It only backs up disks that are locally attached to the VM.

What are the VM configurations stored in the virtual machine backup?

All the VM configurations required to perform the restore operations are stored in the VM backup. That includes the encryption copies of the VM encrypted keys, which are accessible to you on restore. The encryption copies can only be decrypted using the key vault. Temporary disks and memory state aren't captured in the snapshot.

Can we back up tags from Azure VM? If so, how many tags?

Azure Backup can back up and restore tags, except NICs and IPs. Azure Backup honors the subscription limitations of Azure Resource Group and restores up to 50 tags.

For detailed information, see Subscription limits.

Can I take on-demand (ad-hoc) backup without scheduling backup for an Azure VM?

No, you can't trigger on-demand backups by disabling scheduled backup.

How does Azure Backup process work for cluster nodes?

Every Azure VM in a cluster is considered as an individual Azure VM. So, all backup operations are applicable as per individual Azure VMs.

Does Azure Backup interfere with application performance?

Creating a VM Snapshot takes few minutes, and there will be a minimal interference on application performance at this stage. But, data transfer to a vault takes a couple of hours; so we recommend scheduling backups during off business hours. Learn more about best practices for backup and restore.

Will a new disk added to VM be backed up automatically?

Yes, a new disk added to a VM will be backed up automatically during the next backup, unless selective disk backup is configured on the VM. A newly added disk to the VM has a Full snapshot (Initial Replica) taken on the first backup, the existing disks continue the Incremental logic chain.

Can I restore the files and folders from an encrypted VM backup?

Restoring files and folders from encrypted VM backup is currently not supported, you must recover the entire VM to restore files and folders. See, steps to restore an encrypted Azure Virtual machine. However you can recover files from backups before they were encrypted.

Can I stop a VM and scale it up during the Transfer data to vault phase?

Yes, you can do this when *Transfer data to vault* phase is in progress.

Does Azure Backup take backup of keys for ADE encrypted VMs and restore it along with the restored disk?

Azure Backup backs up encryption keys and secrets of the backup data. Generally, the keys aren't restored in the Key vault, but Azure Backup allows restoring the keys during the loss of keys.

Can I remove all restore points and retain the latest one?

No, you can't retain one single restore point. When you delete the previous restore points, the chain gets deleted. Therefore, you can't selectively retain or delete a recovery point.

In Azure VM Backup, what is the Snapshot Tier and Storage Redundancy for snapshot taken as part of backup?

In Azure VM Backup, snapshots are taken using the Standard HDD tier type, regardless of the source VM disk tier. By default, the storage redundancy for these snapshots is Locally Redundant Storage (LRS). However, if the VM backups are managed by an Enhanced Policy and the underlying source disk storage locates in the region that supports Zone Redundancy, snapshots are of Zone Redundant Storage.

How long can I retain backup points of Azure VMs using the backup policy?

The maximum retentions of the backup points are:

- **Retention of daily backup points**: 9999 days
- **Retention of weekly backup points**: 5163 weeks
- **Retention of monthly backup points**: 1188 months

- **Retention of yearly backup points**: 99 years

Can I back up a VM disks that are created in the Azure Classic Storage?

No, the support for disks created in Azure Classic Storage is no longer available. To ensure continued functionality, you must migrate these disks to Azure Resource Manager Storage. Also, Classic Storage accounts can't support ARM-based Virtual Machines (VMs), and no workarounds exist to enable compatibility.

> **Important**
>
> Support for backup of all Classic VMs is planned for deprecation on September 30, 2025. We recommend that you migrate the Classic VMs and associated disks to Azure Resource Manager Storage before this timeline to avoid disruptions.

How Soft-Delete behavior occurs when the retention of the backup policy is changed or modified?

When the retention period of a backup policy is changed, the soft-delete behavior for restore points (RPs) is determined based on the current policy at the time of evaluation—not the policy in place when the RP was created.

I have an Azure VM stored in a zone with a Premium SSD ZRS disk. What happens if I restore this VM to a Paired region using CRR where ZRS isn't supported?

When restoring a VM to a peered region that doesn't support Zone-Redundant Storage (ZRS), Azure Backup will automatically create the disk in the same performance tier (e.g., Standard or Premium) as the original disk, but with Locally Redundant Storage (LRS) instead of ZRS. This ensures that the restore operation completes successfully, even though the redundancy model changes.

Is 64 TB disk support for Azure Backup available in all regions?

Azure Backup supports individual disk sizes up to 64 TB and a combined total of 512 TB per VM. This support is available in all Azure regions where Azure Backup services are offered.

How does the MDCC backup works via Enhanced Policy with and without selecting the option "only Crash Consistency Snapshot"?

The behavior of MDCC backup under Enhanced Policy depends on whether the **"Only Crash Consistency Snapshot"** checkbox is selected:

When the checkbox is NOT selected:

Enhanced Policy will attempt to take application-consistent or filesystem-consistent backups. However, if these attempts fail, the policy will fallback to crash-consistent backups at specific intervals:

- A crash-consistent snapshot will be taken on the 4th, 6th, and 8th attempt.
- This fallback ensures backup continuity even when consistency cannot be guaranteed. Reasons for fallback to MDCC include offline VMs, VMs with unsupported OS, Guest Agent issues, VMSnapshotLinux extension provisioning issues etc.

When the checkbox is selected:

The policy won't attempt application or filesystem consistency. Instead, it will directly initiate a crash-consistent MDCC backup without any retries for higher consistency levels.

This option is useful when speed and simplicity are prioritized over consistency guarantees.

When we migrate a protected item from the Standard to Enhanced policy, which retention period applies on the restore points?

Retention Period applies as per the new enhanced policy for all the restore points.

Why don’t I see the updated expiry time for my Recovery Points immediately after changing the backup policy?

When you modify a backup policy (for example, changing the retention duration), the updated expiry time for Recovery Points (RPs) isn't reflected instantly on the Azure portal. This is expected behavior.

The expiry time is updated only after the system runs its daily cleanup process (also known as the expired Recovery Points pruning job). This job typically runs once every 24 hours. Therefore, it may take up to 24 hours for the new expiry time to appear on the portal for existing or newly created Recovery Points.

If you don’t see the updated expiry time immediately, please wait for up to 24 hours before raising a support request.

Restore

How do I decide whether to restore disks only or a full VM?

Think of a VM restore as a quick create option for an Azure VM. This option changes disk names, containers used by the disks, public IP addresses, and network interface names. The change maintains unique resources when a VM is created. The VM isn't added to an availability set.

You can use the restore disk option if you want to:

- Customize the VM that gets created. For example, change the size.
- Add configuration settings that weren't there at the time of backup.
- Control the naming convention for resources that are created.
- Add the VM to an availability set.
- Add any other setting that must be configured using PowerShell or a template.

Can I restore backups of unmanaged VM disks after I upgrade to managed disks?

Yes, you can use backups taken before disks were migrated from unmanaged to managed. Azure Backup restores those legacy recovery points as managed disks.

How do I restore a VM to a restore point before the VM was migrated to managed disks?

The restore process remains the same. If the recovery point is from a point in time when the VM had unmanaged disks, follow Restore legacy recovery points from VMs that used unmanaged disks. If the VM had managed disks, restore the disks as managed disks by following managed disk restore. Then you can create a VM from those disks.

Learn more about doing this in PowerShell.

If the restore fails to create the VM, what happens to the disks included in the restore?

In the event of a managed VM restore, even if the VM creation fails, the disks will still be restored.

Can I restore a VM that's been deleted?

Yes. Even if you delete the VM, you can go to the corresponding backup item in the vault and restore from a recovery point.

How do I restore a VM to the same availability sets?

For managed disk Azure VMs, restoring to the availability sets is enabled by providing an option in the template while restoring as managed disks. This template has the input parameter called **Availability sets**.

How do we get faster restore performance?

Instant Restore capability helps with faster backups and instant restores from the snapshots.

What happens when we change the key vault settings for the encrypted VM?

After you change the key vault settings for the encrypted VM, backups will continue to work with the new set of details. However, after the restore from a recovery point before the change, you'll have to restore the secrets in a key vault before you can create the VM from it. For more information, see this article.

Operations like secret/key roll-over don't require this step and the same key vault can be used after restore.

Can I access the VM once restored due to a VM having a broken relationship with the domain controller?

Yes, you can access the VM once restored due to a VM having a broken relationship with the domain controller. For more information, see this article.

Can I cancel an in-progress restore job?

Yes, you can cancel the restore job till the data transfer phase. Once it enters VM creation phase, you can't cancel the restore job.

Why is my restore operation taking long time to complete?

The total restore time depends on the input/output operations per second (IOPS) speed and the throughput of the storage account. The total restore time can be affected if the target storage account is loaded with other application read and write operations. To improve the speed of restore operation, select a storage account that isn't loaded with other application data. Learn about the best practices for Azure VM backup and restore.

How do we handle "Create New Virtual Machine"-restore type conflicts with governance policies?

Azure Backup uses "attach" disks from recovery points and doesn't look at your image references or galleries. So in the policy you can check "storageProfile.osDisk.createOption as Attach", and the script condition will be:

```
if (storageProfile.osDisk.createOption == "Attach") then { exclude <Policy> }
```

How do I restore a VM into powered down state?

To restore a VM in powered down state, you can create a VM or restore disks, but you can't replace an existing VM. Learn more about the available restore options.

Can I restore Azure VM that is in the powered-off state?

Azure VM must be in the powered-off state while you restore. Otherwise, the restore operation would fail in the pre-check stage, with the error code *UserErrorVmNotShutDown*.

Also, before you swap VM disks, you must power off the VM.

Is there a limit to trigger restore in a day?

The restore limit applies per data source within a 24-hour window. You can restore the same VM up to 20 times in that period. Restoring multiple VMs simultaneously is allowed, but may trigger throttling from other Azure services, such as Azure Resource Manager due to request limits.

How does restored disk SKU depend on target SA provided?

This guidance applied to older restore flows that restored disks as unmanaged VHDs. Recovery points from VMs that used unmanaged disks are now restored only as managed disks, so the target storage account doesn't determine the final disk type. The storage account is used only to hold restore metadata and temporary VHD files.

> **Note**
>
> - Azure Backup currently doesn't support taking different SAs as inputs for different disks.

Can I delete JSON template and VHD files after the restore process is complete?

Yes, you can delete these files once the restoration process is complete. By default, Azure Backup retains these files for future use.

How do I run restore operation for Cross Region Restore (CRR) of ADE encrypted VMs?

The encrypted keys aren't expected to be present in the target region as part of Cross Regions Restore (CRR). Therefore, you need to restore the encrypted keys and secrets using the

restored file. When the restore is complete, you can create Azure encrypted VM using restored disks.

Are there any recommended naming conventions of an Azure VM in Azure Backup?

Microsoft Windows allows a VM name that has maximum of 15 characters. Also, you can't specify a DNS host name that differs from the *NETBIOS* host name. However, you can create host headers for a website hosted on an Azure VM with the name as per recommendation.

Learn more about the VM naming convention limitations for Azure VMs.

Can I restore an Azure Virtual Machine in a different subscription?

Yes, Cross Subscription Restore now allows you to restore Azure VMs from a recovery point in one subscription to another under tenant as per Azure role-based access control (Azure RBAC) rules. Cross Subscription Restore is unsupported from snapshots.

Does Cross Subscription Restore support all Azure VM?

No, it's unsupported for Encrypted Azure VMs.

Can I use Azure VM snapshots to restore in another subscription?

No, Cross Subscription Restore is unsupported from snapshot restore.

Can I perform Cross Subscription Restore for Azure VMs running in secondary regions?

Yes.

Can I use Enhanced policy for Cross Subscription Restore?

Yes, it's supported for Cross subscription Restore.

Can I restore an Azure zone pinned Virtual Machine in a different zone?

Yes, Cross Zonal Restore now allows you to restore Azure zone pinned VMs to a different available zone using a recovery point in a vault with Zone-redundant storage (ZRS) enabled as per Azure role-based access control (Azure RBAC) rules. It's also supported from vaults with Cross Region Restore (CRR).

Can I restore an Azure nonzone pinned Virtual Machine in a different zone?

Yes, Cross Zonal Restore now allows you to restore Azure non-zone pinned VMs to any available zones using a recovery point in a vault with Zone-redundant storage (ZRS) enabled as per Azure role-based access control (Azure RBAC) rules.

Does Cross Zonal Restore support all Azure VM?

No, it's unsupported for Encrypted Azure VMs.

Can I use Azure VM snapshots to restore in another zone?

No, Cross Zonal Restore is unsupported from snapshot restore.

Can I restore Azure zone pinned VMs to secondary regions?

Yes, Azure Backup support restore of Azure zone pinned VMs to secondary regions.

Can I use Enhanced policy for Cross Zonal Restore?

Yes, it's supported for Cross Zonal Restore.

How do I clear iSCSI session and its processes after running the Python script for Linux ILR?

After you unmount disks from the Azure portal, run the Python script with `clean` parameter (`python scriptName.py clean`) to clear the session and remove the mount paths of the recovery point from the machine.

Why do restored VMs lose the Offer/SKU/Publisher properties of the source VM?

During a virtual machine (VM) restore using Azure Backup, you might notice that the restored VM doesn't retain the Offer, SKU, and Publisher properties that were present on the source VM. This is an expected behavior, and happens due to the following reasons:

- **MarketPlace Image VMs**: Only VMs created from a MarketPlace image has the Offer, SKU, and Publisher details. These properties are derived from the VM image itself. When a VM is created from a MarketPlace image, the createOption is set to **from image**. This ensures that the Offer, SKU, and Publisher properties are included during the VM creation process.
- **VMs Created from Disks**: When a VM is created from disks (such as restored VMs, uploaded VMs, or VMs created from snapshots), the createOption is set to **Attach** instead of "from image". In this case, the Offer, SKU, and Publisher properties are not included because these properties are read-only and can only be added during the initial VM creation process when the createOption is **from image**.

What is the expected throughput when we perform the restore from the Vault storage.

For Tier-2 restores (restores from vault data), the time required depends on the amount of allocated data that needs to be transferred. In one observed case, disks with approximately 2 TB of allocated data each contributed significantly to the overall transfer time. While a transfer rate of 30 MBps is considered optimal, we achieved around 50 MBps in this instance.

For example, if a customer has a 16-TB disk with 2 TB of allocated data per segment, the total restore time could extend up to 20 hours at the optimal transfer rate.

Manage VM backups

What happens if I modify a backup policy?

The VM is backed up using the schedule and retention settings in the modified policy.

- If retention is extended, existing recovery points are marked and kept in accordance with the new policy.
- If retention is reduced, recovery points are marked for pruning in the next cleanup job, and subsequently deleted.

However, recovery points are specific to retention range frequency. For example, adding or modifying a yearly retention policy doesn't affect the retention of preexisting monthly recovery points.

How do I move a VM backed up by Azure Backup to a different resource group?

1. Temporarily stop the backup and retain backup data.

2. To move virtual machines configured with Azure Backup, do the following steps:
   a. Find the location of your virtual machine.
   b. Find a resource group with the following naming pattern: `AzureBackupRG_<location of your VM>_1`. For example, *AzureBackupRG_westus2_1*
   c. In the Azure portal, check **Show hidden types**.
   d. Find the resource with type **Microsoft.Compute/restorePointCollections** that has the naming pattern `AzureBackup_<name of your VM that you're trying to move>_###########`.
   e. Delete this resource. This operation deletes only the instant recovery points, not the backed-up data in the vault.
   f. After the delete operation is complete, you can move your virtual machine.

3. Move the VM to the target resource group.

4. Resume the backup.

You can restore the VM from available restore points that were created before the move operation.

What happens after I move a VM to a different resource group?

Once a VM is moved to a different resource group, it's a new VM as far as Azure Backup is concerned.

After moving the VM to a new resource group, you can reprotect the VM either in the same vault or a different vault. Since this is a new VM for Azure Backup, you are billed for it separately.

The old VM's restore points will be available for restore if needed. If you don't need this backup data, you can stop protecting your old VM with delete data.

Is there a limit on number of VMs that can be associated with the same backup policy?

Yes, there's a limit of 100 VMs that can be associated to the same backup policy. We recommend that for more than 100 VMs, create multiple backup policies with same schedule or different schedule.

There's a daily limit of 1000 for overall configure/modify protections in a vault.

How can I view the retention settings for my backups?

Currently, you can view retention settings at a backup item (VM) level based on the backup policy that's assigned to the VM.

One way to view the retention settings for your backups, is to navigate to the backup item dashboard for your VM, in the Azure portal. Selecting the link to its backup policy helps you view the retention duration of all the daily, weekly, monthly, and yearly retention points associated with the VM.

You can also use Backup Explorer to view the retention settings for all your VMs within a single pane of glass. Go to the Backup Explorer from any Recovery Services vault, go to the **Backup Items** tab and select the Advanced View to see detailed retention information for each VM.

When the snapshot is moved from a storage account to a vault, how is encryption in the transit managed?

Azure VM Backup uses HTTPS communication for encryption in transit. The data transfer uses Azure fabric (and not public endpoints), which do not need Internet access for VM backup.

How can I disable the File Recovery option?

This API provision a script for invoking an **iSCSI connection** for file recovery from Azure Backup.

- You can disable this option using **custom role-definitions** by excluding API action.
- You can also use the private endpoints to restrict access to the iSCSI server from within the private network.
- You can also disable this option across an organization using the deny assignment feature.

I have changed the retention policy, what is the time needed for the policy to be effective?

The policy takes effect immediately after the modifications of the parameters, such as retention, schedule, and so on. This is applicable for all new backups taken from the modified policy. However, the pruning of the recovery points (if applicable) according to the new policy takes 24 hours.

How do I extend or reduce the retention of a specific recovery point?

This feature is currently not supported. You can post any feature ask in the Azure Backup community portal.

How to modify retention period for Stopped backups?

Retention of stopped backups cannot be modified since they do not have any policy attached to it. However, you can resume protection and assign a policy.

How long are the stopped backups retained?

Stopped backups are retained until manually deleted.

I'm unable to select a virtual network, subnet, or storage account in the secondary region when performing a Cross Region Restore.

You need to check the subscription permissions in the secondary region. Write to us at AskAzureBackupTeam@microsoft.com for subscription enrollment.

How can I check the traffic between Azure Backup service and Azure VM?

Because Azure resources handle this traffic, it can't be determined by an external user.

What is the minimum RPO and RTO for VM backups in Azure Backup?

RPO: The minimum RPO is 1 day or 24 hours when you're using Standard policy. If you use Enhanced policy, the minimum RPO is 4 hours.

Why does the error VMMarketplaceInvalidInputmessage appear?

This error appears when you try to start a VM after creating an Azure VM from a non-Marketplace image or swap the OS disk of a VM with a non-Marketplace image, and then the VM deployment fails. To resolve this issue, remove the plan information from the VM.

How do I manage key rotations? How to ensure which key is used during backup and if it’s present to be used with the restored VM?

Azure Backup backs up the secrets and KEK data of the key version during backup, and restores the same. However, booting ADE VMs with older version keys are also possible.

How often are Microsoft managed keys rotated for Azure Backup?

Microsoft-managed keys are rotated as per compliance requirements. If you have specific key rotation requirements, we recommend that you move to customer-managed keys so that you can manage and audit the rotation manually. Learn more [about Azure Storage encryption] (..\storage\common\storage-service-encryption.md).

When Vault is configured with CRR, what happens to the secondary data if the primary region fails?

Backup data fully replicated to the secondary region before the failure of the primary region remains intact. This remains true even after the primary region recovers from the failure. In other words, the virtual machine can be recovered in the secondary region with the data it had before the failure, based on the replication schedule. For Azure VM backups, the secondary region RPO can be up to 36 hours in the worst case. With the Standard policy, this is based on up to 24 hours in the primary region plus up to 12 hours for replication to the secondary region. With the Enhanced policy, more frequent local recovery point creation can improve the best-case achievable secondary region RPO, but the worst-case can still be up to 36 hours.

When I update the backup policy, why is the expiry time not getting updated immediately?

Expiry Time of recovery points are updated when Garbage Collector (GC) runs, which is *every 24 hours*. Once you update the backup policy, it can take up to *24 hours* to show the updates in the **Expiry Time**, if there are no delays in GC jobs.

Why the expiry time for latest recovery points doesn't appear in the Azure portal for VMs?

The expiry time for the latest recovery point doesn't appear because it has infinite retention until a newer one gets created. The expiry time of other recovery points will be available in the Azure portal after 24 hours of creation, depending on when the garbage collection runs (that will be within 24 hours).

Will an Azure VM restore operation to original or alternate location affect a scheduled backup?

No, the restore operation (to the original or an alternate location) doesn't affect scheduled backups.

- **Restore to Original Location**: Azure Backup replaces the existing VM's disks with the ones from the restore point. This process includes taking a snapshot of the current VM before the restore ensuring that the backup schedule remains.

- **Restore to Alternate Location**: Azure Backup creates a new VM using the restore point. This new VM won't have the same backup schedule as the original VM; you need to configure the Backup policy manually. Learn about Azure VM restore.

In these restore scenarios, the original backup policy and schedule for the original VM remain unaffected. However, if you want the restored VM (in the alternate location) to have a backup schedule, you need to set up the Backup policy.

What happens to the backup configuration if I do Original/Alternate Location Restore of Azure VMs?

In case of an Original Location Restore (OLR) or an Alternate Location Restore (ALR) for Azure VMs, Azure Backup performs the backup configuration as:

Expand table

Restore type	Backup configuration	Backup data
Original Location Restore (OLR)	The backup configuration remains the same. The restored VM uses the same backup policy and schedule as the original VM.	The existing backup data is preserved, and new backups are triggered as per the original backup schedule.
Alternate Location Restore (ALR)	The new VM created in the alternate location doesn't inherit the backup configuration of the original VM. You need to configure a new backup policy for this VM manually.	The backup data for the original VM remains unaffected. However, the new VM starts with a fresh backup history once you configure its backup.

> ⓘ Note
>
> - **OLR**: Provides a seamless continuation of backups with the same policy.
> - **ALR**: Requires manual setup of backup policies for the new VM.

Why VMs with the Storage Spaces direct and Storage Spaces aren't supported with the Azure VM Backup?

Storage Spaces Direct isn't supported since it has data spread across two different virtual machines.

Storage spaces in a single VM not supported for application consistent VM backup. For application consistent VM backup, VSS freezes the I/O and ensures that disk snapshots are consistent. Storage Spaces can still perform writes in the background, breaking the application consistent guarantee. This can result in a small probability that the backup taken isn't application consistent. Today, backup, and recovery of VM with storage spaces isn't blocked in the product.

If I remove one of the disks from VM and increase the size of existing disk, what happens to the backups and if next backup creates Full Backup for all disk?

If you remove a disk, the snapshot or data transfer operation won't run for the removed disk. Number of disks included in the restore point differs depending on the VM state at the time of the backup.

Resizing a disk continues with incremental backups unless the underlying storage account of the disk changes, which results in a full backup.

What is directory "/var/log/azure/Microsoft.Azure.RecoveryServices.VMSnapshotLinux"?

This is the directory used by Backup Extension VMSnapshotLinux and it gets created when backups are enabled on VM using Recovery Services Vault. Logs are written in this location during backups, restores, or extension operations such as schedule updates, generally they should use up only a few GB and the size of the logs is not proportional with the size of the backup.

If backup fails with a timeout error for large disk VMs after running sometime, when can I retrigger the backup job again and will the backup continue from where it failed?

For Initial Replica, a full backup will be reinitiated in case of failure. However, for Incremental the snapshot from the last successful backup will be compared to the latest snapshot to copy the changed data into the vault.

Why Premium SSD v2 with GRS Vaults/Cross Region Restore is not supported?

Premium SSD v2/Ultra Disks are currently not available in all regions, so Geo-Redundant Storage (GRS) can't be enabled.

> ⓘ **Note**
>
> GRS functions only when both paired regions support Premium SSD v2/Ultra Disks.

Why don’t I see the updated expiry time for my Recovery Points immediately after changing the backup policy?

When you modify a backup policy (for example, changing the retention duration), the updated expiry time for Recovery Points (RPs) is not reflected instantly on the Azure portal. This is expected behavior.

The expiry time is updated only after the system runs its daily cleanup process (also known as the expired Recovery Points pruning job). This job typically runs once every 24 hours. Therefore, it may take up to 24 hours for the new expiry time to appear on the portal for existing or newly created Recovery Points.

If you don’t see the updated expiry time immediately, please wait for up to 24 hours before raising a support request.

About Site Recovery

Article • 04/01/2025

Welcome to the Azure Site Recovery service! This article provides a quick service overview.

As an organization, you need to adopt a business continuity and disaster recovery (BCDR) strategy that keeps your data safe, and your apps and workloads online, when planned and unplanned outages occur.

Azure Recovery Services contributes to your BCDR strategy:

- **Site Recovery service**: Site Recovery helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. When an outage occurs at your primary site, you fail over to a secondary location, and access apps from there. After the primary location is running again, you can fail back to it.
- **Backup service**: The Azure Backup service keeps your data safe and recoverable.

Azure Site Recovery has an option of *High Churn*, enabling you to configure disaster recovery for Azure VMs having data churn up to 100 MB/s. This helps you to enable disaster recovery for more IO intensive workloads. Learn more.

Site Recovery can manage replication for:

- Azure VMs replicating between Azure regions
- Replication from Azure Extended Zones to the region
- On-premises VMs, Azure Stack VMs, and physical servers

> **Note**
>
> The Azure Site Recovery functionality for Extended Zones is in preview state.

What does Site Recovery provide?

Expand table

Feature	Details
Simple BCDR solution	Using Site Recovery, you can set up and manage replication, failover, and failback from a single location in the Azure portal.
VMware VM replication	You can replicate VMware VMs to Azure using the improved Azure Site Recovery replication appliance that offers better security and resilience than the configuration

Feature	Details
	server. For more information, see Disaster recovery of VMware VMs.
On-premises VM replication	You can replicate on-premises VMs and physical servers to Azure. Replication to Azure eliminates the cost and complexity of maintaining a secondary datacenter.
Workload replication	Replicate any workload running on supported Azure VMs, on-premises Hyper-V and VMware VMs, and Windows/Linux physical servers.
Data resilience	Site Recovery orchestrates replication without intercepting application data. When you replicate to Azure, data is stored in Azure storage, with the resilience that provides. When failover occurs, Azure VMs are created based on the replicated data. This also applies to Extended Zones to Azure region Azure Site Recovery scenario.
RTO and RPO targets	Keep recovery time objectives (RTO) and recovery point objectives (RPO) within organizational limits. Site Recovery provides continuous replication for Azure VMs and VMware VMs, and replication frequency as low as 30 seconds for Hyper-V. You can reduce RTO further by integrating with Azure Traffic Manager.
Keep apps consistent over failover	You can replicate using recovery points with application-consistent snapshots. These snapshots capture disk data, all data in memory, and all transactions in process.
Testing without disruption	You can easily run disaster recovery drills, without affecting ongoing replication.
Flexible failovers	You can run planned failovers for expected outages with zero-data loss. Or, unplanned failovers with minimal data loss, depending on replication frequency, for unexpected disasters. You can easily fail back to your primary site when it's available again.
Customized recovery plans	Using recovery plans, you can customize and sequence the failover and recovery of multi-tier applications running on multiple VMs. You group machines together in a recovery plan, and optionally add scripts and manual actions. Recovery plans can be integrated with Azure Automation runbooks. **Note**: This functionality is currently supported for Region-to-Region replication and will be available on Azure Extended Zones soon.
BCDR integration	Site Recovery integrates with other BCDR technologies. For example, you can use Site Recovery to protect the SQL Server backend of corporate workloads, with native support for SQL Server Always On, to manage the failover of availability groups.
Azure automation integration	A rich Azure Automation library provides production-ready, application-specific scripts that can be downloaded and integrated with Site Recovery.
Network integration	Site Recovery integrates with Azure for application network management. For example, to reserve IP addresses, configure load-balancers, and use Azure Traffic Manager for efficient network switchovers.

Feature	Details
Shared disk	You can protect, monitor, failover, and re-protect your workloads running on Windows Server Failover Clusters (WSFC) on Azure VMs using shared disk. You can use shared disks for your critical applications such as SQL FCI, SAP ASCS, Scale-out File Servers, etc., while ensuring business continuity and disaster recovery with Azure Site Recovery.

What can I replicate?

Expand table

Supported	Details
Replication scenarios	Replicate Azure VMs from 1. One Azure region to another. 2. One Azure Extended zone to the Azure region it's connected to. Replicate on-premises VMware VMs, Hyper-V VMs, physical servers (Windows and Linux), Azure Stack VMs to Azure. Replicate AWS Windows instances to Azure. Replicate on-premises VMware VMs, Hyper-V VMs managed by System Center VMM, and physical servers to a secondary site.
Regions	Review supported regions for Site Recovery.
Replicated machines	Review the replication requirements for Azure VM replication, on-premises VMware VMs and physical servers, and on-premises Hyper-V VMs.
Workloads	You can replicate any workload running on a machine that's supported for replication. Learn more about the app-specific workload summary.

Next steps

- Read more about workload support.
- Get started with Azure VM replication between regions.
- Get started with VMware VM replication.
- Get started with Disaster recovery for VMs on Azure Extended Zones.

Tutorial: Set up disaster recovery for Linux virtual machines

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Flexible scale sets

This tutorial shows you how to set up disaster recovery for Azure VMs running Linux. In this article, learn how to:

- ✔ Enable disaster recovery for a Linux VM
- ✔ Run a disaster recovery drill to check it works as expected
- ✔ Stop replicating the VM after the drill

When you enable replication for a VM, the Site Recovery Mobility service extension installs on the VM, and registers it with Azure Site Recovery. During replication, VM disk writes are sent to a cache storage account in the source VM region. Data is sent from there to the target region, and recovery points are generated from the data. When you fail a VM over to another region during disaster recovery, a recovery point is used to create a VM in the target region.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

1. Check that your Azure subscription allows you to create a VM in the target region. If you just created your free Azure account, you're the administrator of the subscription, and you have the permissions you need.
2. If you're not the subscription administrator, work with the administrator to assign you:
   - Either the Virtual Machine Contributor built-in role, or specific permissions to:
     - Create a VM in the selected virtual network.
     - Write to an Azure storage account.
     - Write to an Azure managed disk.
   - The Site Recovery Contributor built-in role, to manage Site Recovery operations in the vault.
3. Check that the Linux VM is running a supported operating system.
4. If VM outbound connections use a URL-based proxy, make sure it can access these URLs. Using an authenticated proxy isn't supported.

Expand table

Name	Public cloud	Government cloud	Details
Storage	`*.blob.core.windows.net`	`*.blob.core.usgovcloudapi.net`	Write data from the VM to the cache storage account in the source region.

Name	Public cloud	Government cloud	Details
Microsoft Entra ID	`login.microsoftonline.com`	`login.microsoftonline.us`	Authorize and authenticate to Site Recovery service URLs.
Replication	`*.hypervrecoverymanager.windowsazure.com`	`*.hypervrecoverymanager.windowsazure.com`	VM communication with the Site Recovery service.
Service Bus	`*.servicebus.windows.net`	`*.servicebus.usgovcloudapi.net`	VM writes to Site Recovery for monitoring and diagnostic data.

5. If you're using network security groups (NSGs) to limit network traffic for VMs, create NSG rules that allow outbound connectivity (HTTPS 443) for the VM using these service tags (groups of IP addresses). Try out the rules on a test NSG first.

Expand table

Tag	Allow
Storage tag	Allows data to be written from the VM to the cache storage account.
Microsoft Entra ID tag	Allows access to all IP addresses that correspond to Microsoft Entra ID.
Events Hub tag	Allows access to Site Recovery monitoring.
Azure Site Recovery tag	Allows access to the Site Recovery service in any region.
GuestAndHybridManagement	Use if you want to automatically upgrade the Site Recovery Mobility agent that's running on VMs enabled for replication.

6. Make sure VMs have the latest root certificates. On Linux VMs, follow the guidance provided by your Linux distributor, to get the latest trusted root certificates and certificate revocation list on the VM.

Create a VM and enable disaster recovery

You can optionally enable disaster recovery when you create a VM.

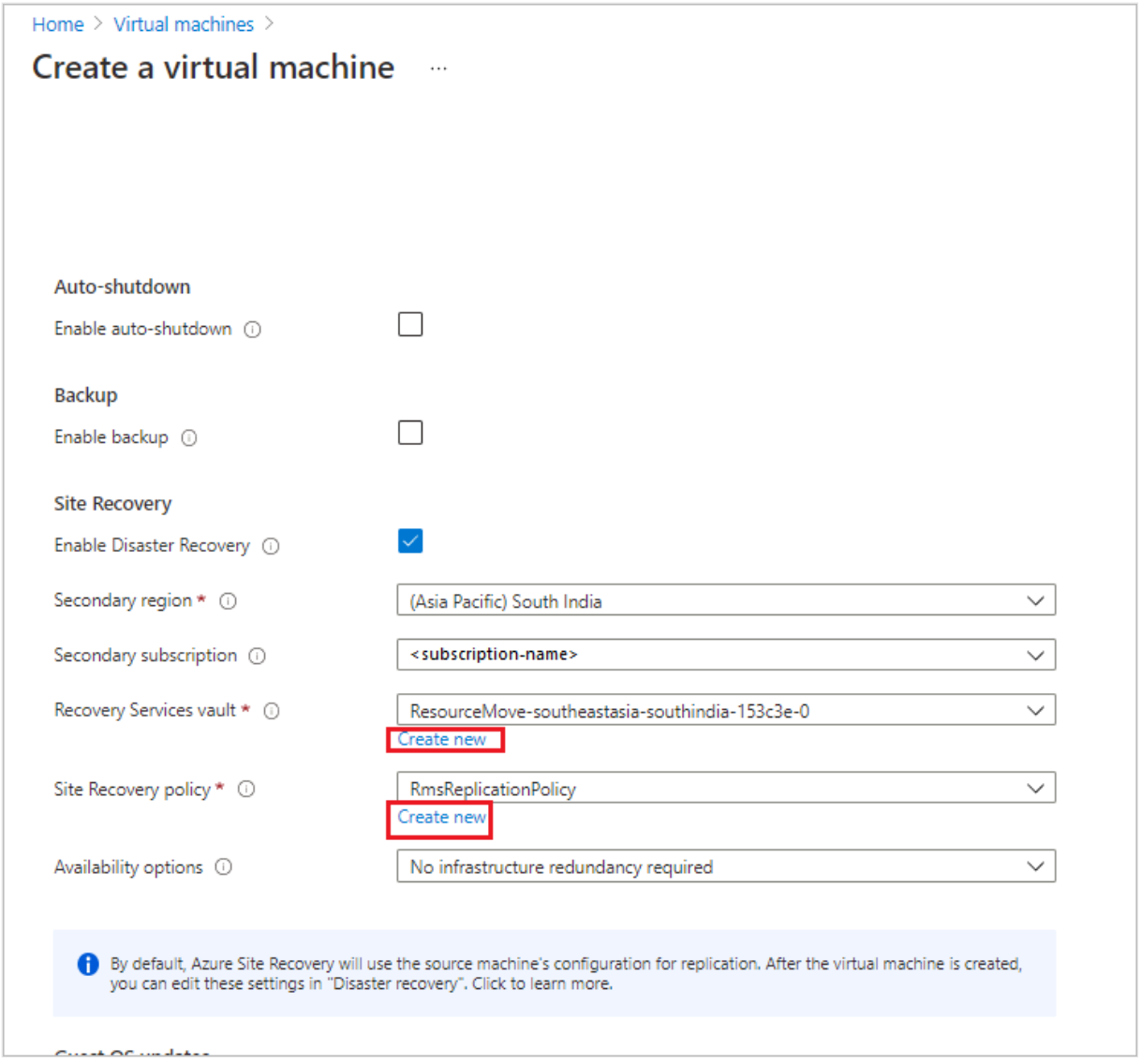
1. Create a Linux VM.
2. On the **Management** tab, under **Site Recovery** select **Enable disaster recovery**.
3. In **Secondary region**, select the target region to which you want to replicate the VM for disaster recovery.
4. In **Secondary subscription**, select the target subscription in which the target VM will be created. The target VM is created when you fail over the source VM from the source region to the target region.

5. In **Recovery Services vault**, select the vault you want to use for the replication. If you don't have a vault, select **Create new**. Select a resource group in which to place the vault, and a vault name.

6. In **Site Recovery policy**, leave the default policy, or select **Create new** to set custom values.

    - Recovery points are created from snapshots of VM disks taken at a specific point in time. When you fail over a VM, you use a recovery point to restore the VM in the target region.
    - A crash-consistent recovery point is created every five minutes. This setting can't be modified. A crash-consistent snapshot captures data that was on the disk when the snapshot was taken. It doesn't include anything in memory.
    - By default Site Recovery keeps crash-consistent recovery points for 24 hours. You can set a custom value between 0 and 72 hours.
    - An app-consistent snapshot is taken every 4 hours.
    - By default Site Recovery stores recovery points for 24 hours.

7. In **Availability options**, specify whether the VM will deploy as standalone, in an availability zone, or in an availability set.

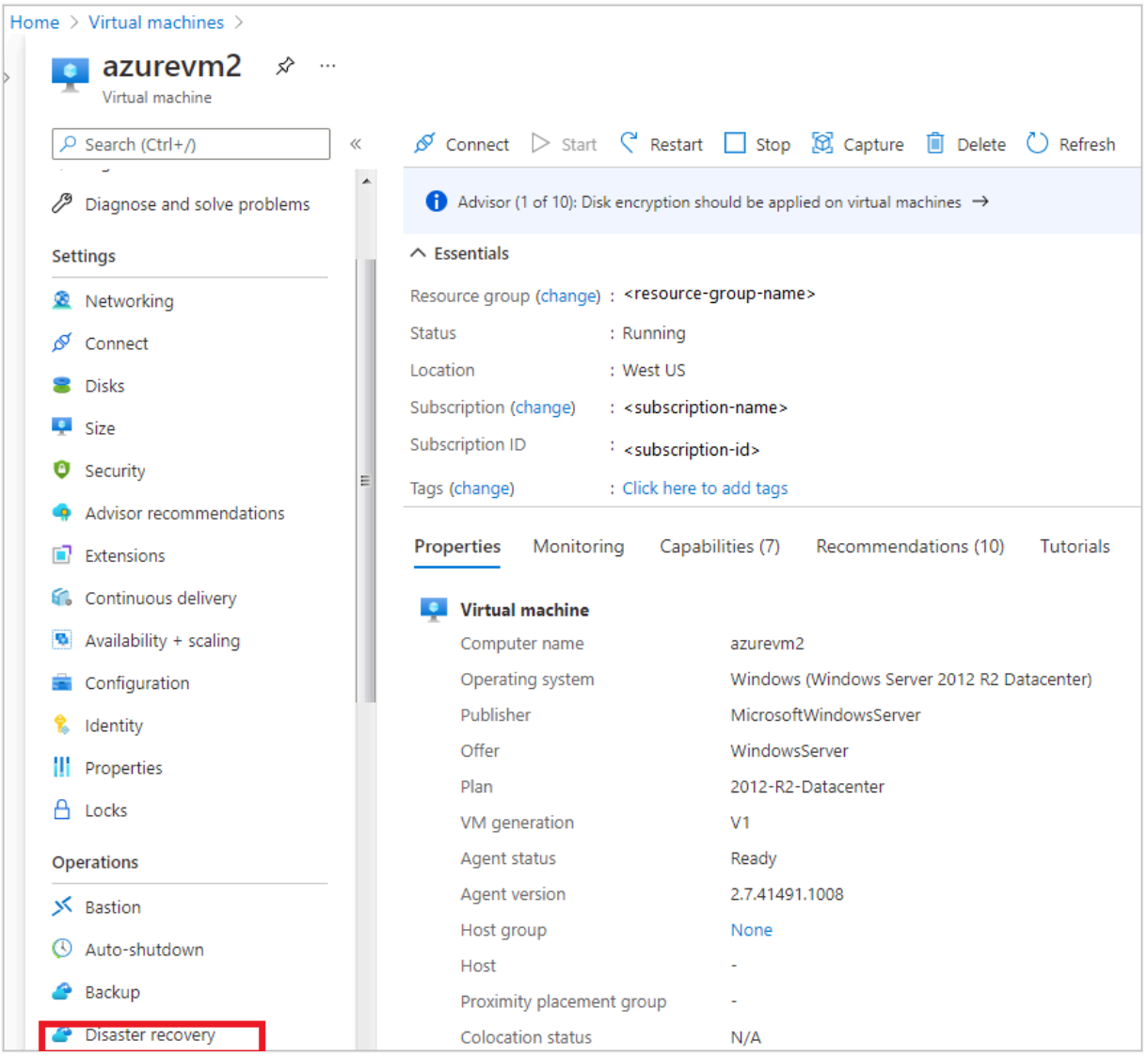


8. Finish creating the VM.

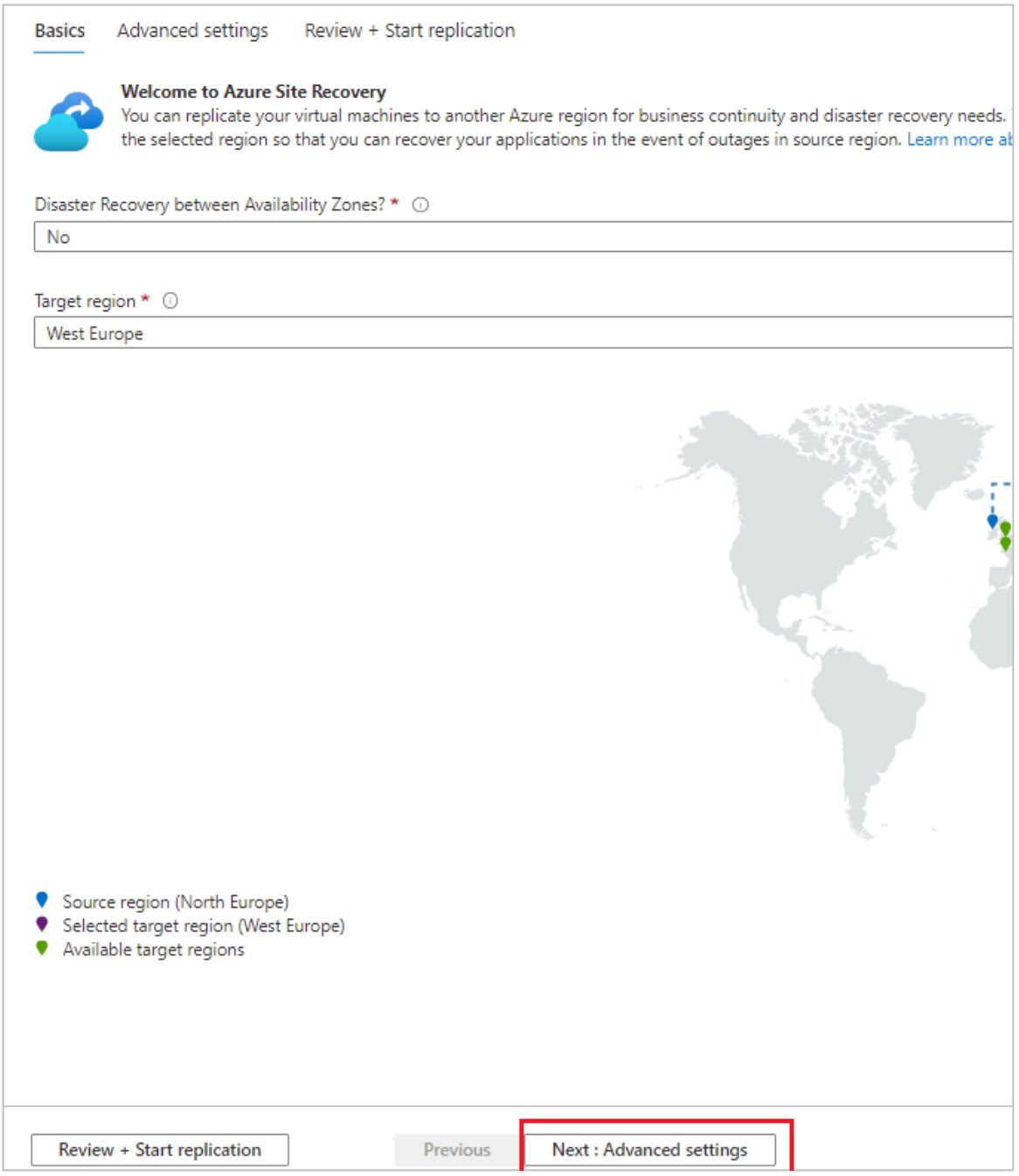
Enable disaster recovery for an existing VM

If you want to enable disaster recovery on an existing VM, use this procedure.

1. In the Azure portal, open the VM properties page.
2. In **Operations**, select **Disaster recovery**.



3. In **Basics**, if the VM is deployed in an availability zone, you can select disaster recovery between availability zones.
4. In **Target region**, select the region to which you want to replicate the VM. The source and target regions must be in the same Microsoft Entra tenant.

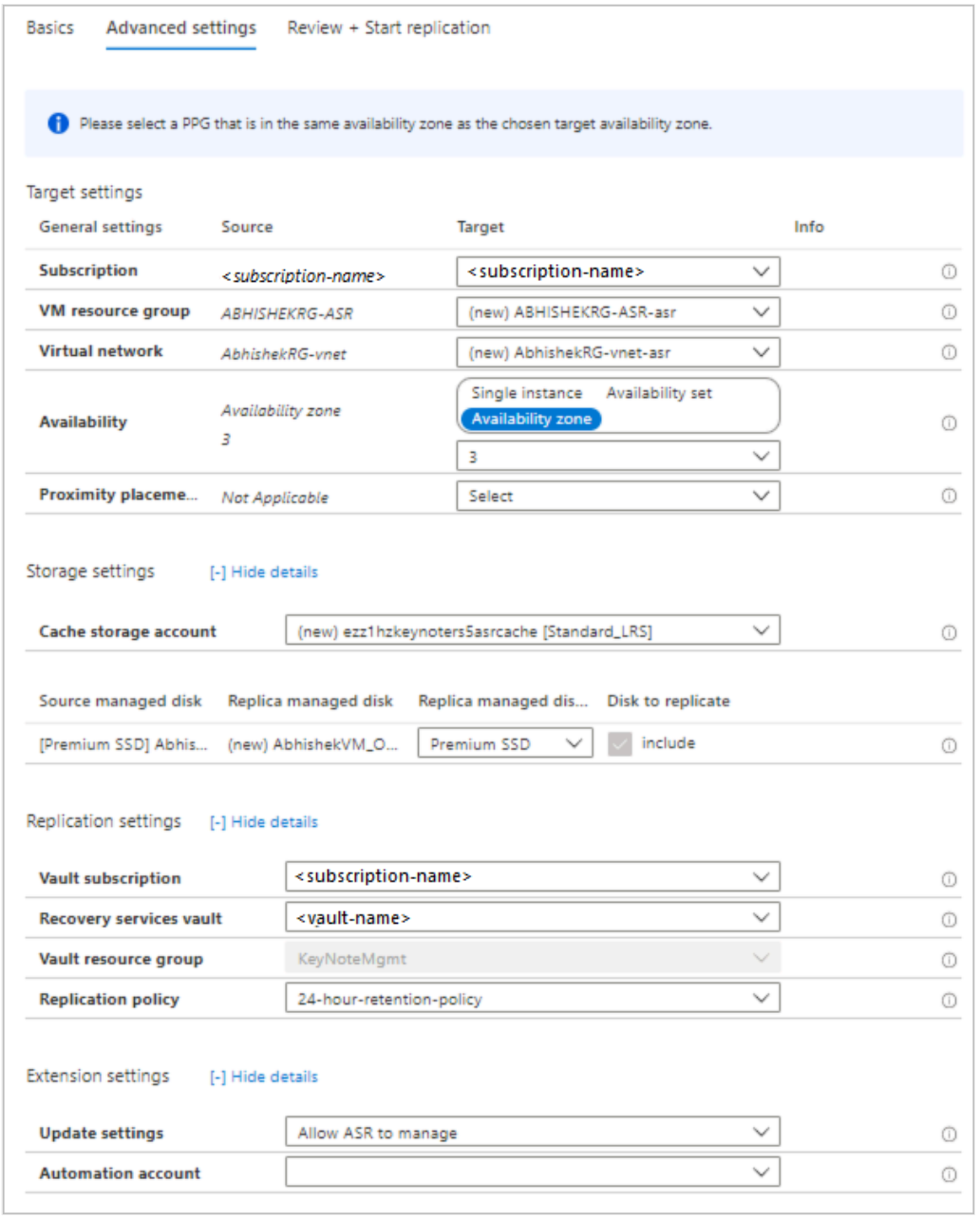


5. Select **Next: Advanced settings**.

6. In **Advanced settings**, you can review settings, and modify values to custom settings. By default, Site Recovery mirrors the source settings to create target resources.

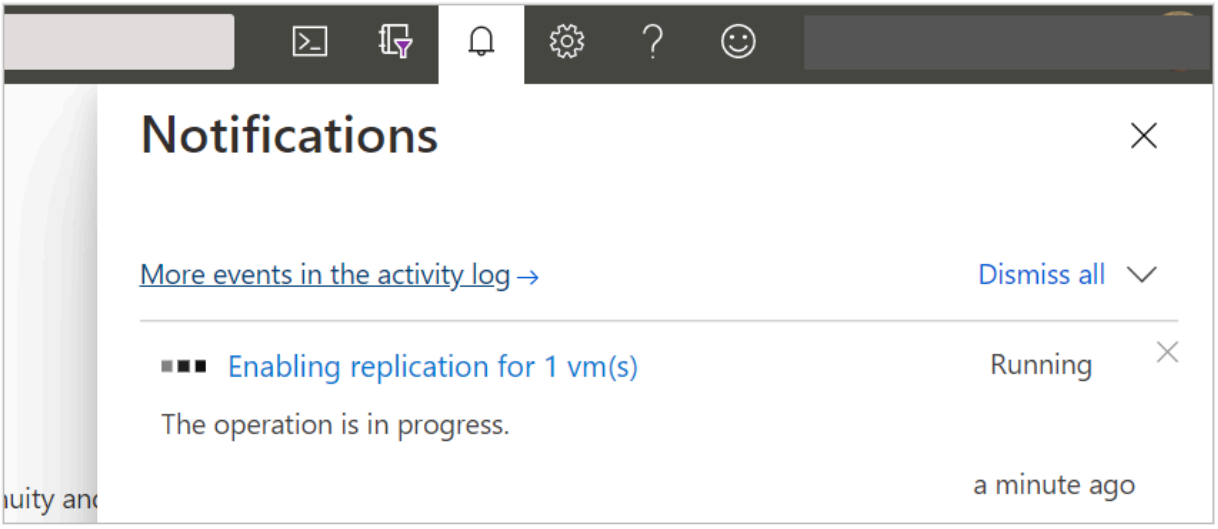
   - **Target subscription**. The subscription in which the target VM is created after failover.
   - **Target VM resource group**. The resource group in which the target VM is created after failover.
   - **Target virtual network**. The Azure virtual network in which the target VM is located when it's created after failover.
   - **Target availability**. When the target VM is created as a single instance, in an availability set, or availability zone.
   - **Proximity placement**. If applicable, select the proximity placement group in which the target VM is located after failover.
   - **Storage settings-Cache storage account**. Recovery uses a storage account in the source region as a temporary data store. Source VM changes are cached in this account, before being replicated to the target location.
     - By default one cache storage account is created per vault and reused.

    - You can select a different storage account if you want to customize the cache account for the VM.
- **Storage settings-Replica managed disk**. By default, Site Recovery creates replica managed disks in the target region.
    - By default the target managed disk mirror the source VM managed disks, using the same storage type (standard HDD/SSD, or premium SSD).
    - You can customize the storage type as needed.
- **Replication settings**. Shows the vault in which the VM is located, and the replication policy used for the VM. By default, recovery points created by Site Recovery for the VM are kept for 24 hours.
- **Extension settings**. Indicates that Site Recovery manages updates to the Site Recovery Mobility Service extension that's installed on VMs you replicate.
    - The indicated Azure Automation account manages the update process.
    - You can customize the automation account.



7. Select **Review + Start replication**.

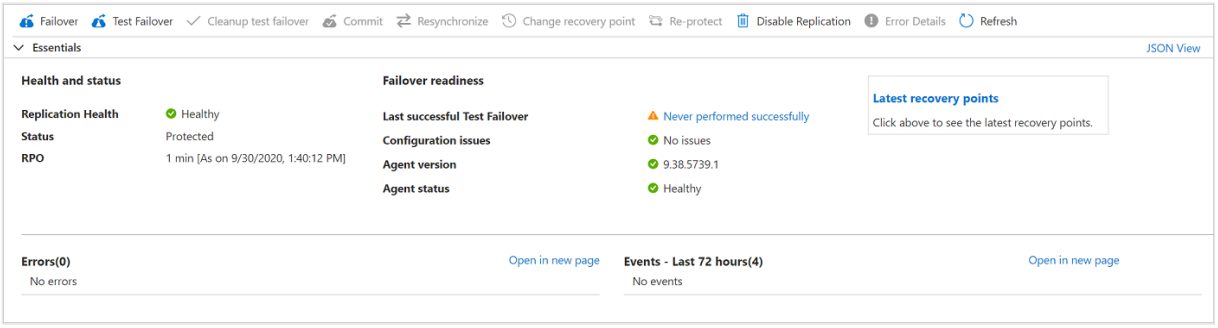
8. Select **Start replication**. Deployment starts, and Site Recovery starts creating target resources. You can monitor replication progress in the notifications.



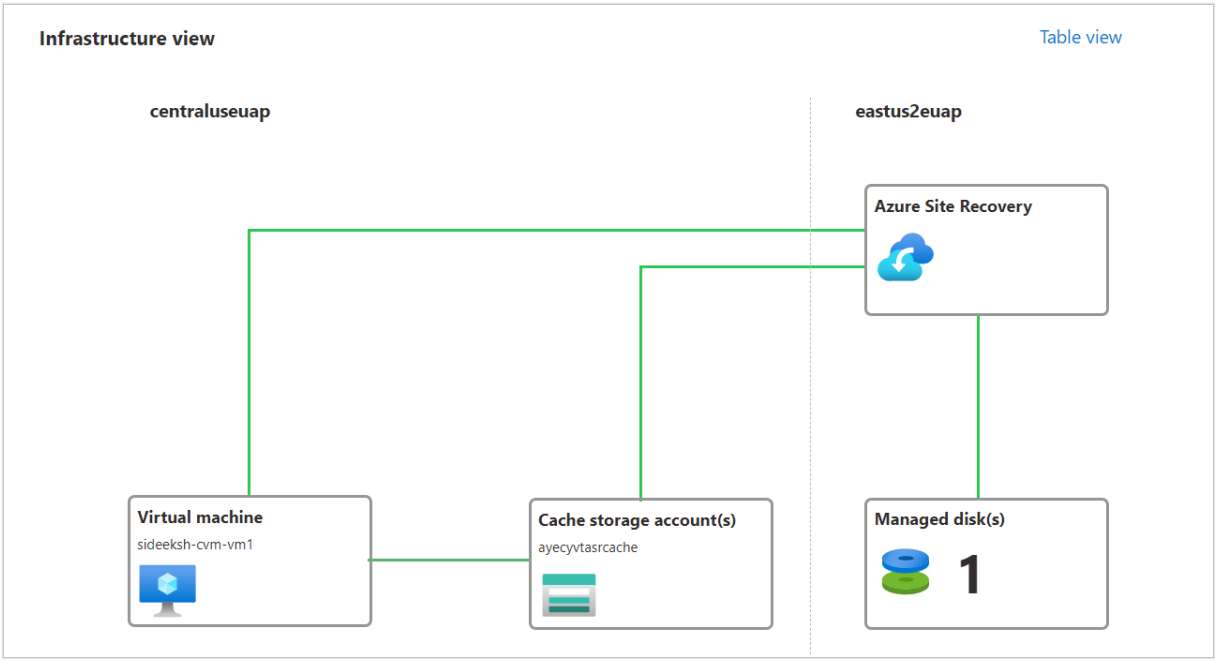
Check VM status

After the replication job finishes, you can check the VM replication status.

1. Open the VM properties page.
2. In **Operations**, select **Disaster recovery**.
3. Expand the **Essentials** section to review defaults about the vault, replication policy, and target settings.
4. In **Health and status**, get information about replication state for the VM, the agent version, failover readiness, and the latest recovery points.



5. In **Infrastructure view**, get a visual overview of source and target VMs, managed disks, and the cache storage account.

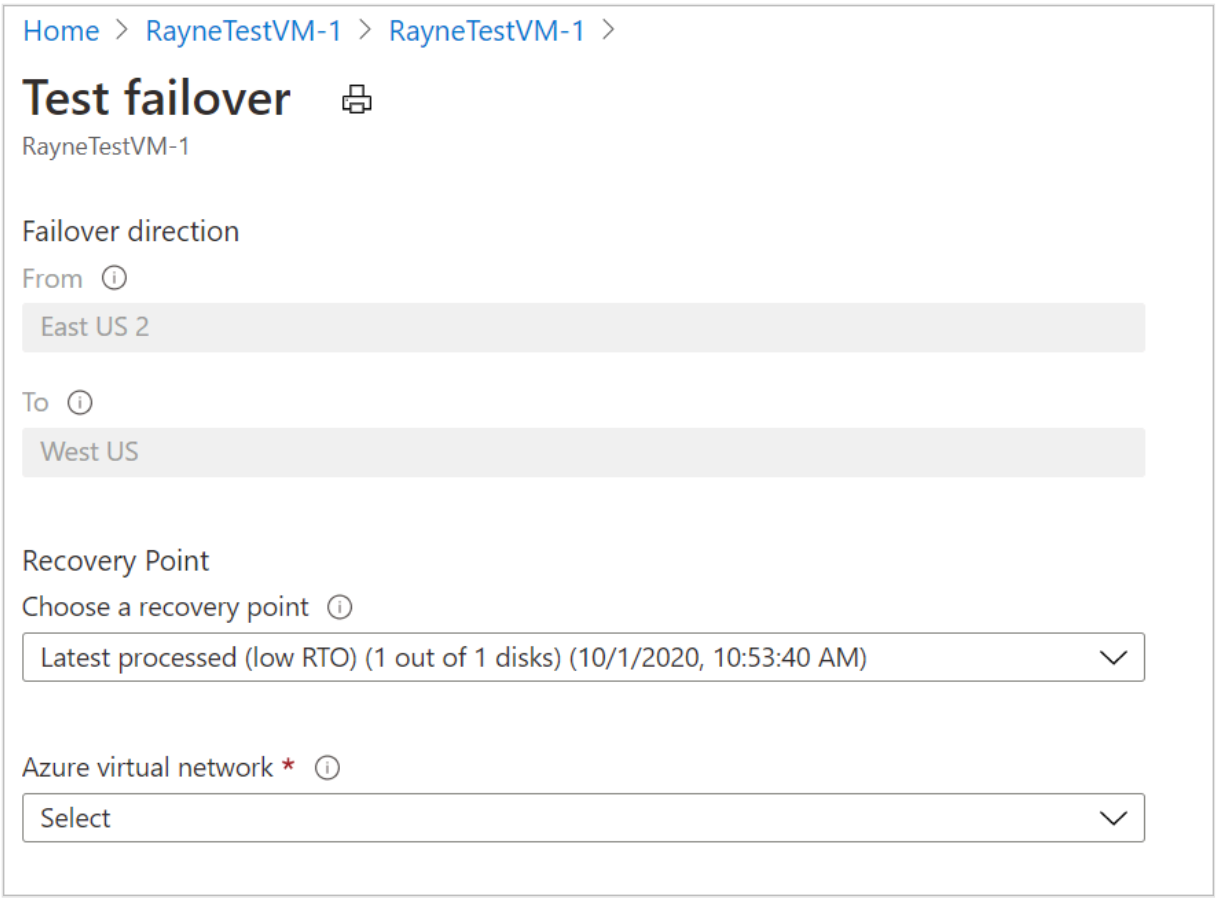


Run a drill

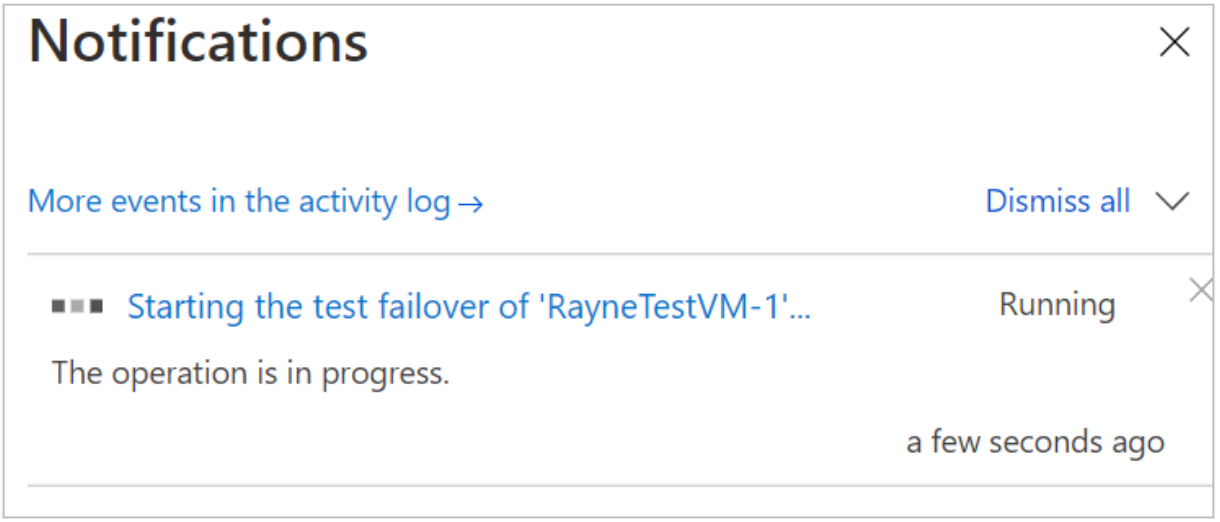
Run a drill to make sure disaster recovery works as expected. When you run a test failover, it creates a copy of the VM, with no impact on ongoing replication, or on your production environment.

1. In the VM disaster recovery page, select **Test failover**.
2. In **Test failover**, leave the default **Latest processed (low RPO)** setting for the recovery point.

   This option provides the lowest recovery point objective (RPO), and generally spins up the VM most quickly in the target region. It first processes all the data that has been sent to Site Recovery service, to create a recovery point for each VM, before failing over to it. This recovery point has all the data replicated to Site Recovery when the failover was triggered.
3. Select the virtual network in which the VM will be located after failover.



4. The test failover process begins. You can monitor the progress in notifications.

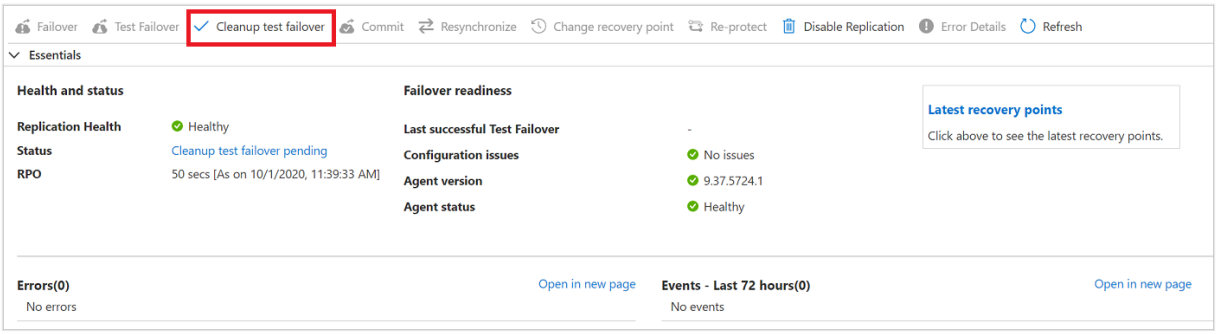


After the test failover completes, the VM is in the *Cleanup test failover pending* state on the **Essentials** page.

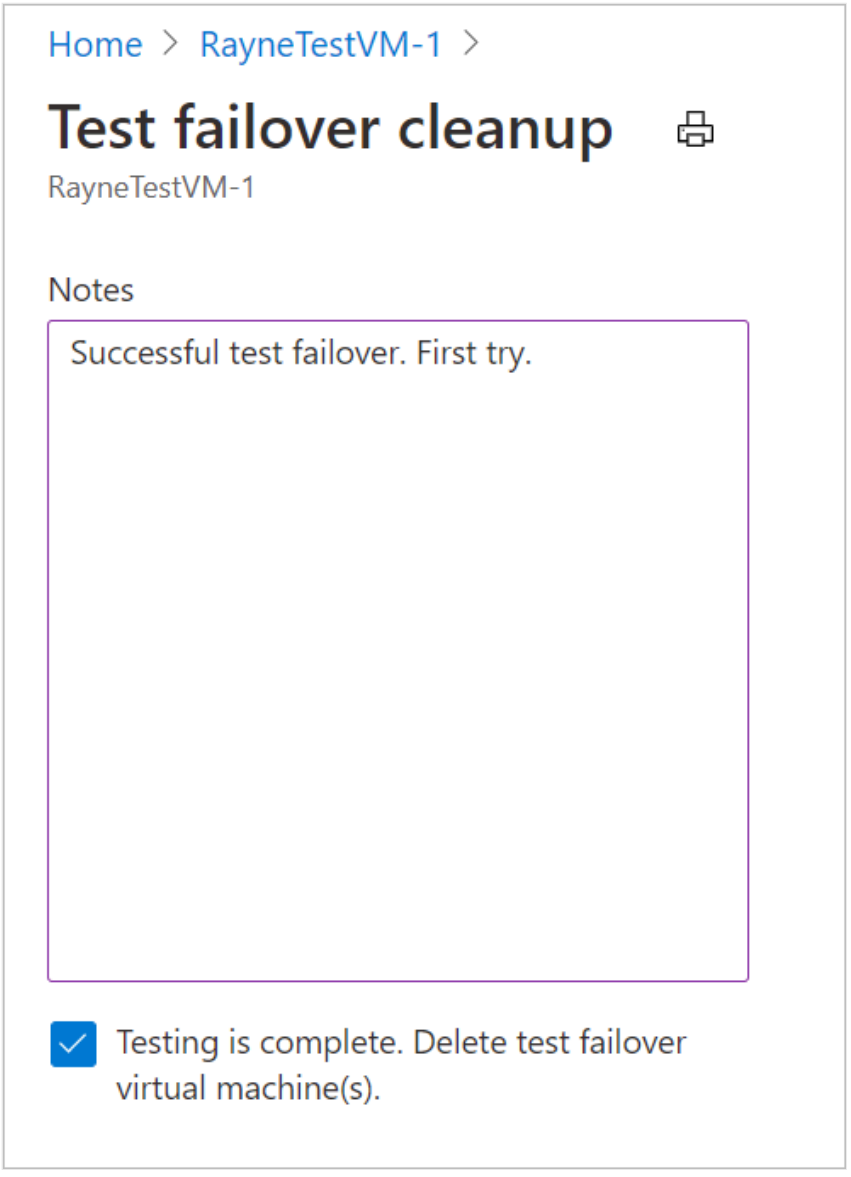
Clean up resources

The VM is automatically cleaned up by Site Recovery after the drill.

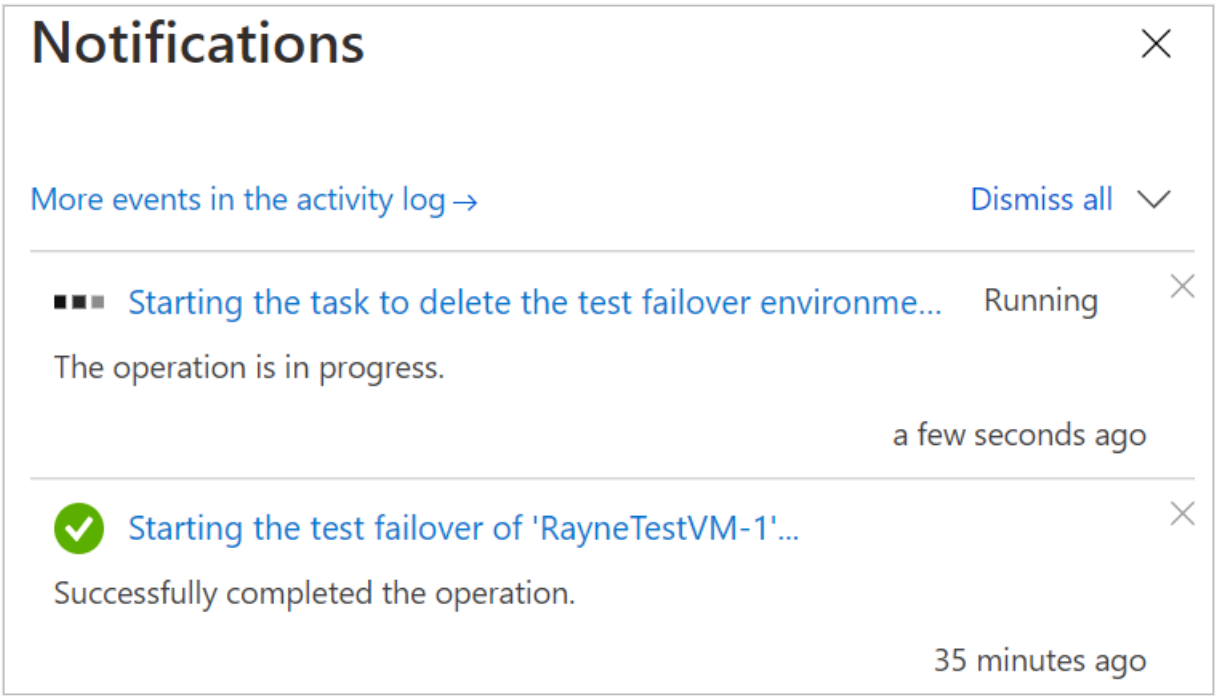
1. To begin automatic cleanup, select **Cleanup test failover**.



2. In **Test failover cleanup**, type in any notes you want to record for the failover, and then select **Testing is complete. Delete test failover virtual machine**. Then select **OK**.



3. The delete process begins. You can monitor progress in notifications.



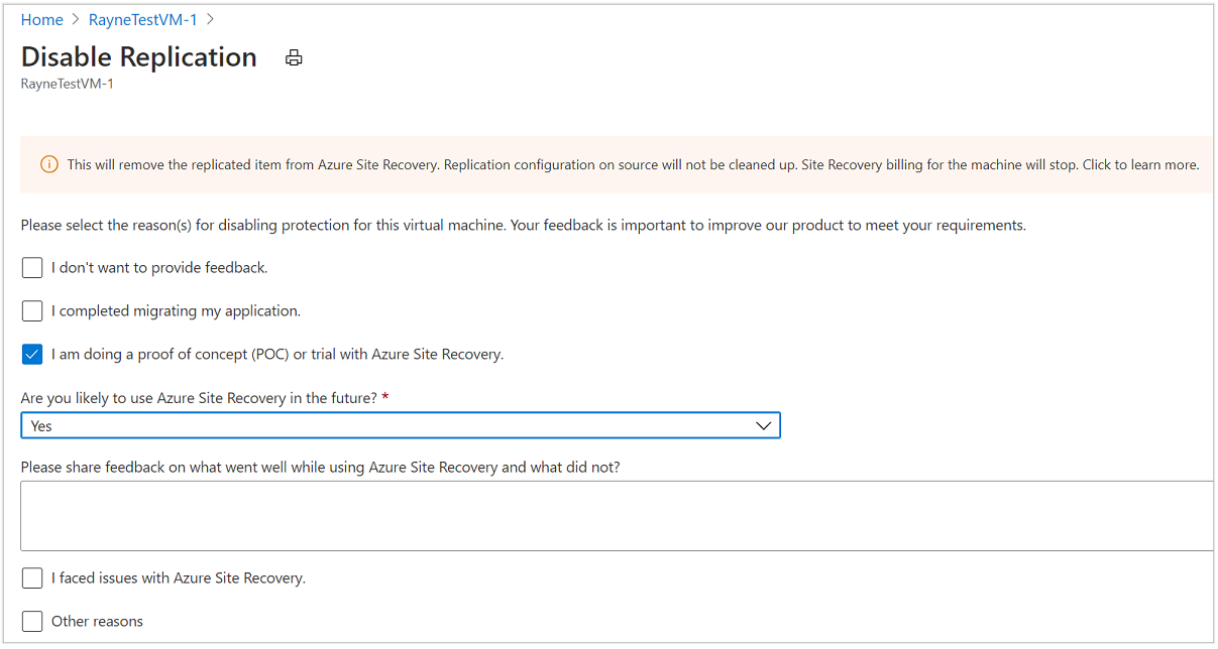
Stop replicating the VM

After completing a disaster recovery drill, we suggest you continue to try out a full failover. If you don't want to do a full failover, you can disable replication. Disabling replication will:

- Remove the VM from the Site Recovery list of replicated machines.
- Stop Site Recovery billing for the VM.
- Automatically clean up source replication settings.

Stop replication as follows:

1. In the VM disaster recovery page, select **Disable Replication**.
2. In **Disable Replication**, select the reasons that you want to disable replication. Then select **OK**.



The Site Recovery extension installed on the VM during replication isn't removed automatically. If you disable replication for the VM, and you don't want to replicate it again at a later time, you can remove the

Site Recovery extension manually, as follows:

1. Go to the VM > **Settings** > **Extensions**.
2. In the **Extensions** page, select each *Microsoft.Azure.RecoveryServices* entry for Linux.
3. In the properties page for the extension, select **Uninstall**.

Next steps

In this tutorial, you configured disaster recovery for an Azure VM, and ran a disaster recovery drill. Now, you can perform a full failover for the VM.

Fail over a VM to another region

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Enable disaster recovery for Windows VMs

Article • 08/22/2024

Applies to: ✔ Windows VMs ✔ Flexible scale sets

This tutorial shows you how to set up disaster recovery for Azure VMs running Windows. In this article, learn how to:

- ✔ Enable disaster recovery for a Windows VM
- ✔ Run a disaster recovery drill to check it works as expected
- ✔ Stop replicating the VM after the drill

When you enable replication for a VM, the Site Recovery Mobility service extension installs on the VM, and registers it with Azure Site Recovery. During replication, VM disk writes are sent to a cache storage account in the source region. Data is sent from there to the target region, and recovery points are generated from the data. When you fail over a VM during disaster recovery, a recovery point is used to create a VM in the target region.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

1. Check that your Azure subscription allows you to create a VM in the target region. If you just created your free Azure account, you're the administrator of the subscription, and you have the permissions you need.
2. If you're not the subscription administrator, work with the administrator to assign you:
   - Either the Virtual Machine Contributor built-in role, or specific permissions to:
     - Create a VM in the selected virtual network.
     - Write to an Azure storage account.
     - Write to an Azure-managed disk.
   - The Site Recovery Contributor built-in role, to manage Site Recovery operations in the vault.
3. We recommend you use a Windows VM running Windows Server 2012 or later. The VM disk shouldn't be encrypted for the purpose of this tutorial.
4. If VM outbound connections use a URL-based proxy, make sure it can access these URLs. Using an authenticated proxy isn't supported.

Expand table

Name	Public cloud	Government cloud	Details
Storage	`*.blob.core.windows.net`	`*.blob.core.usgovcloudapi.net`	Write data from the VM to the cache storage

Name	Public cloud	Government cloud	Details
			account in the source region.
Microsoft Entra ID	`login.microsoftonline.com`	`login.microsoftonline.us`	Authorize and authenticate to Site Recovery service URLs.
Replication	`*.hypervrecoverymanager.windowsazure.com`	`*.hypervrecoverymanager.windowsazure.com`	VM communication with the Site Recovery service.
Service Bus	`*.servicebus.windows.net`	`*.servicebus.usgovcloudapi.net`	VM writes to Site Recovery monitoring and diagnostic data.

5. If you're using network security groups (NSGs) to limit network traffic for VMs, create NSG rules that allow outbound connectivity (HTTPS 443) for the VM using these service tags (groups of IP addresses). Try out the rules on a test NSG first.

Expand table

Tag	Allow
Storage tag	Allows data to be written from the VM to the cache storage account.
Microsoft Entra ID tag	Allows access to all IP addresses that correspond to Microsoft Entra ID.
EventsHub tag	Allows access to Site Recovery monitoring.
AzureSiteRecovery tag	Allows access to the Site Recovery service in any region.
GuestAndHybridManagement	Use if you want to automatically upgrade the Site Recovery Mobility agent that's running on VMs enabled for replication.

6. On Windows VMs, install the latest Windows updates, to make sure that VMs have the latest root certificates.

Create a VM and enable disaster recovery

You can optionally enable disaster recovery when you create a VM.

1. Create a VM.
2. On the **Management** tab, select **Enable disaster recovery**.
3. In **Secondary region**, select the target region to which you want to replicate a VM for disaster recovery.

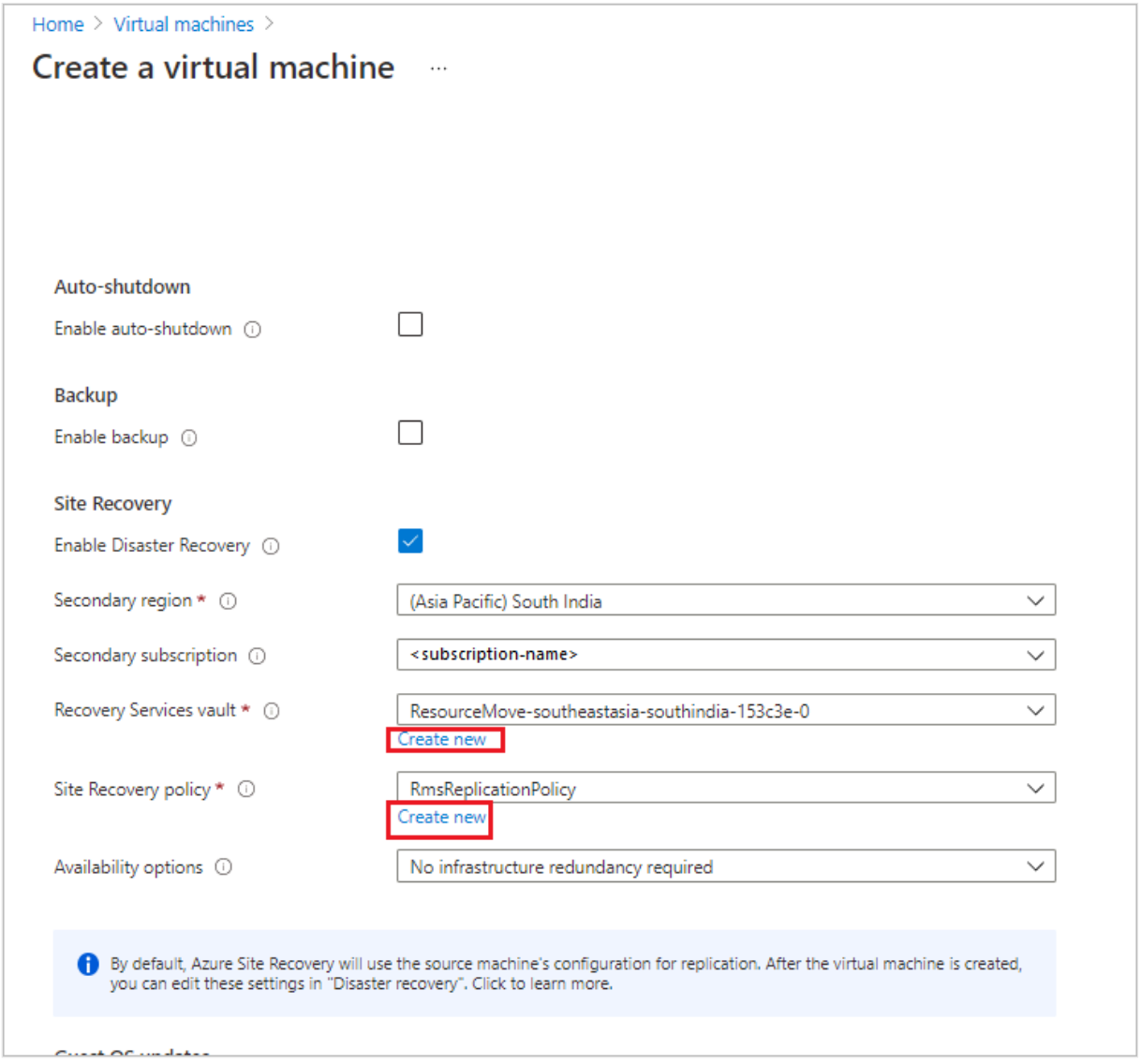
4. In **Secondary subscription**, select the target subscription in which the target VM will be created. The target VM is created when you fail over the source VM from the source region to the target region.

5. In **Recovery Services vault**, select the vault you want to use for the replication. If you don't have a vault, select **Create new**. Select a resource group in which to place the vault, and a vault name.

6. In **Site Recovery policy**, leave the default policy, or select **Create new** to set custom values.

    - Recovery points are created from snapshots of VM disks taken at a specific point in time. When you fail over a VM, you use a recovery point to restore the VM in the target region.
    - A crash-consistent recovery point is created every five minutes. This setting can't be modified. A crash-consistent snapshot captures data that was on the disk when the snapshot was taken. It doesn't include anything in memory.
    - By default Site Recovery keeps crash-consistent recovery points for 24 hours. You can set a custom value between 0 and 72 hours.
    - An app-consistent snapshot is taken every 4 hours. An app-consistent snapshot
    - By default Site Recovery stores recovery points for 24 hours.

7. In **Availability options**, specify whether the VM is deploy as standalone, in an availability zone, or in an availability set.



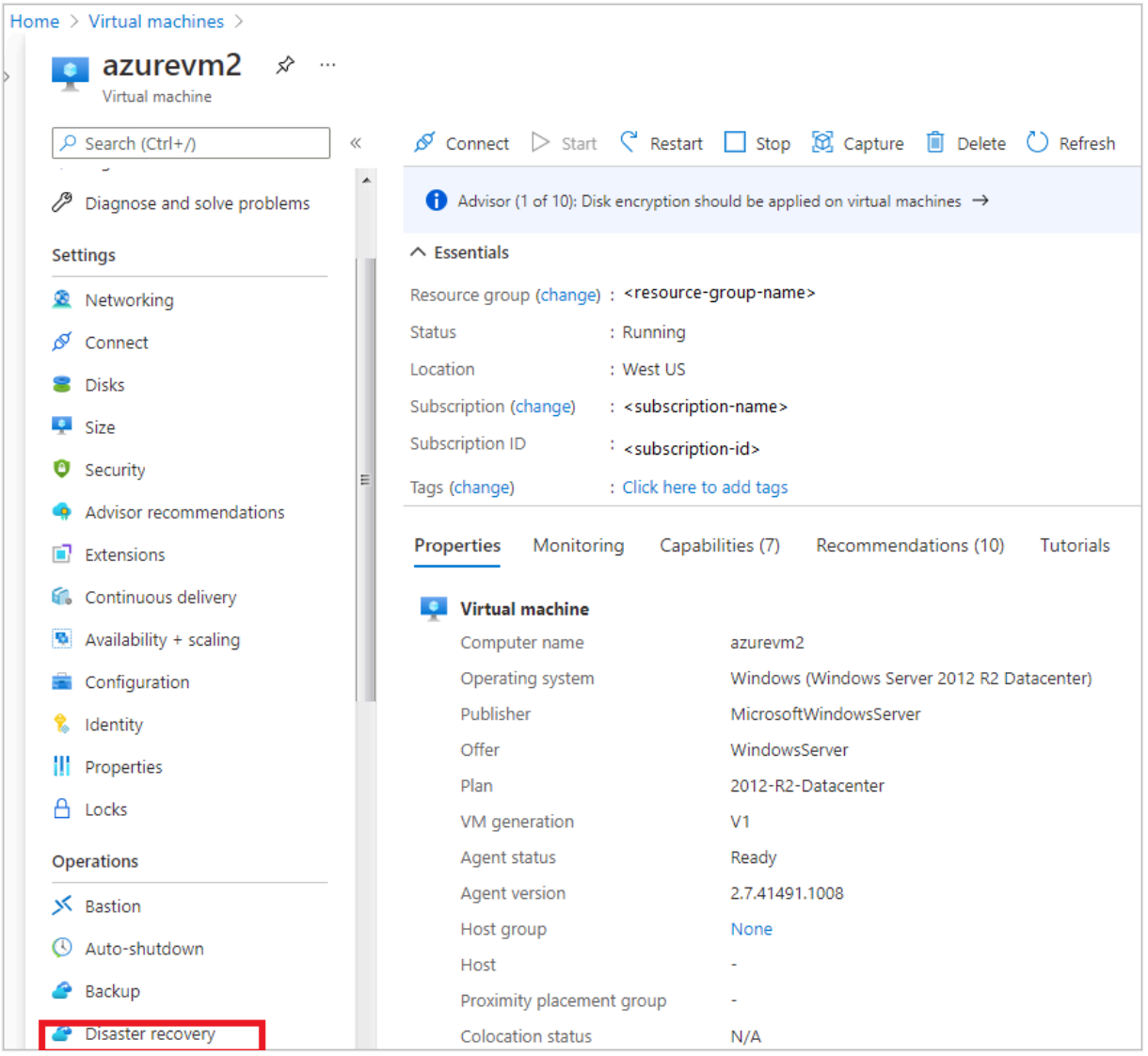
8. Finish creating the VM.

> **Note**
>
> When you enable replication while creating a Windows VM, only the OS disk gets replicated. Data disks need to be initialized by you, after which Azure Site Recovery automatically replicates them.

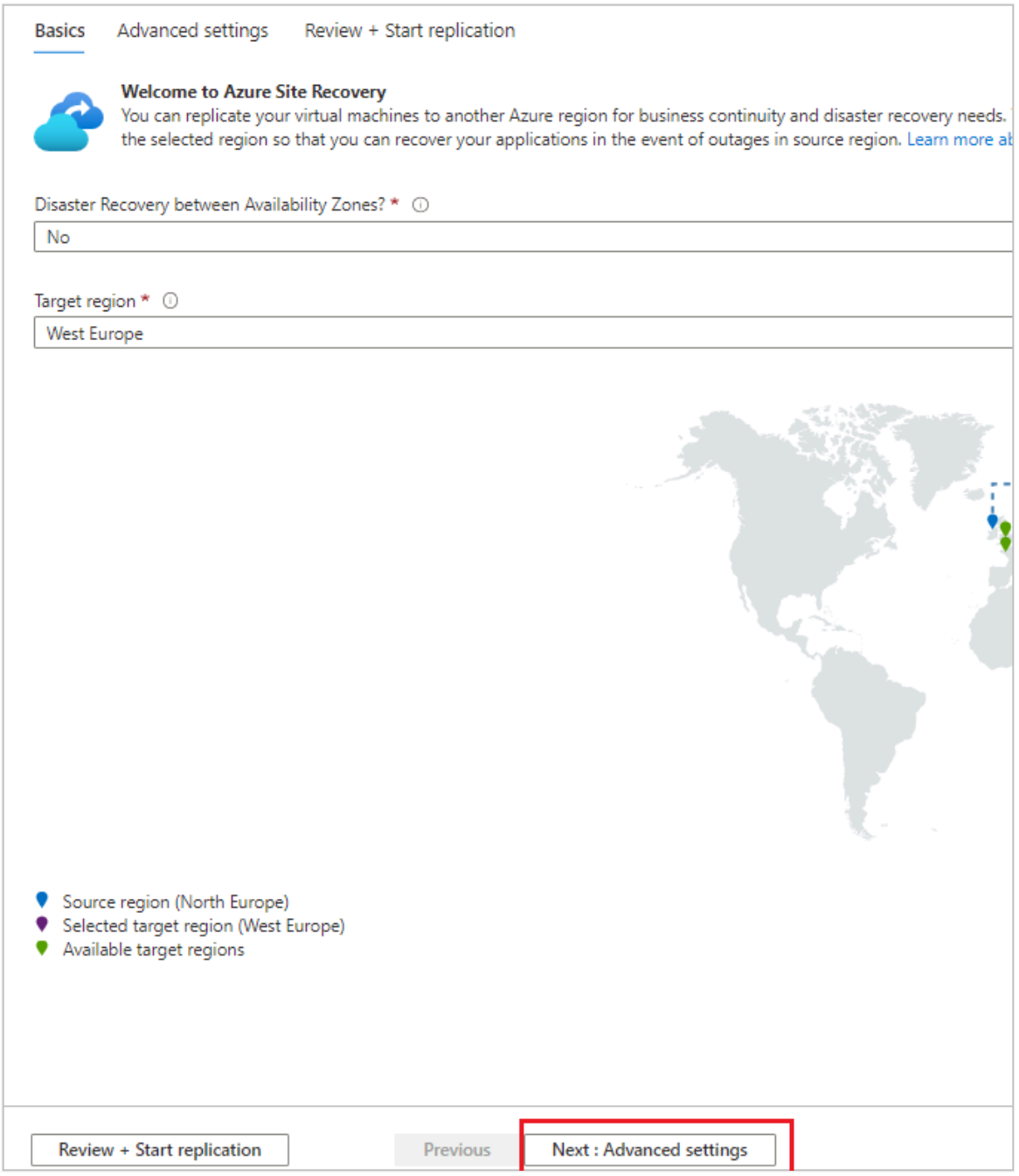
Enable disaster recovery for an existing VM

If you want to enable disaster recovery on an existing VM instead of for a new VM, use this procedure.

1. In the Azure portal, open the VM properties page.
2. In **Operations**, select **Disaster recovery**.



3. In **Basics**, if the VM is deployed in an availability zone, you can select disaster recovery between availability zones.
4. In **Target region**, select the region to which you want to replicate the VM. The source and target regions must be in the same Microsoft Entra tenant.

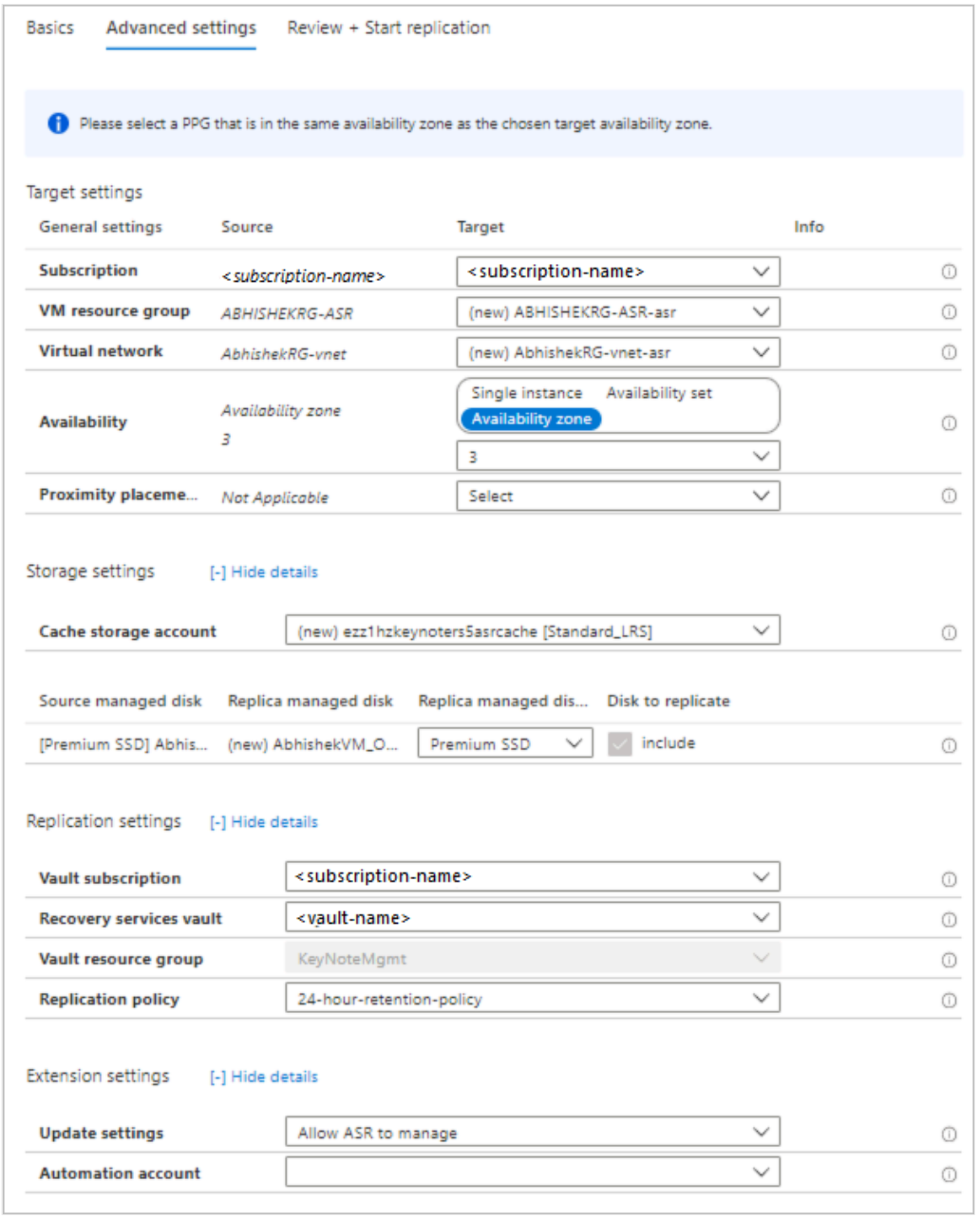


5. Select **Next: Advanced settings**.

6. In **Advanced settings**, you can review settings, and modify values to custom settings. By default, Site Recovery mirrors the source settings to create target resources.

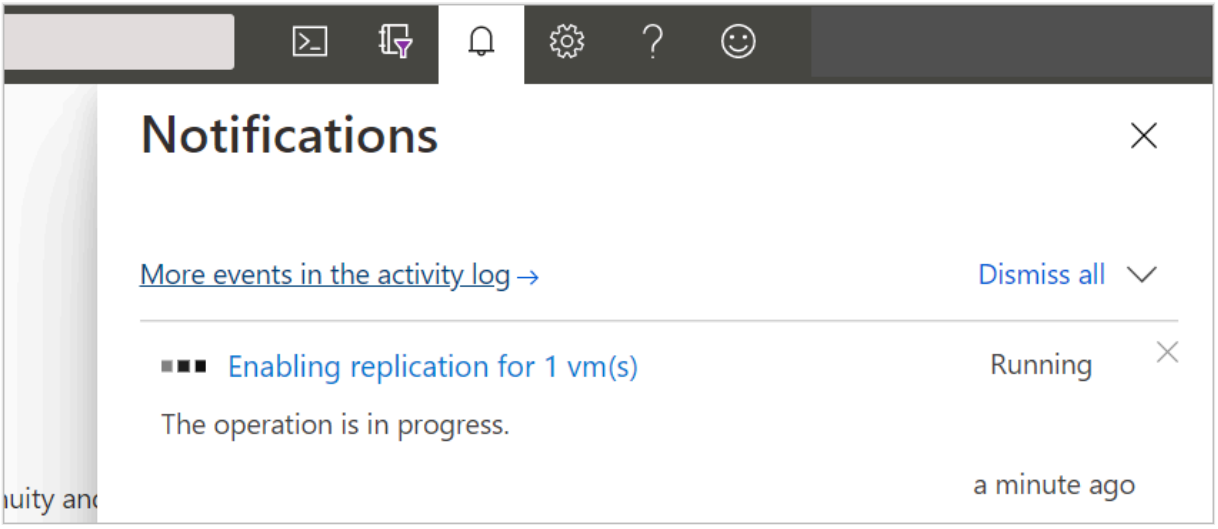
   - **Target subscription**. The subscription in which the target VM is created after failover.
   - **Target VM resource group**. The resource group in which the target VM is created after failover.
   - **Target virtual network**. The Azure virtual network in which the target VM is located when it's created after failover.
   - **Target availability**. When the target VM is created as a single instance, in an availability set, or availability zone.
   - **Proximity placement**. If applicable, select the proximity placement group in which the target VM is located after failover.
   - **Storage settings-Cache storage account**. Recovery uses a storage account in the source region as a temporary data store. Source VM changes are cached in this account, before being replicated to the target location.
     - By default one cache storage account is created per vault and reused.

    - You can select a different storage account if you want to customize the cache account for the VM.
- **Storage settings-Replica managed disk**. By default, Site Recovery creates replica managed disks in the target region.
    - By default the target managed disk mirror the source VM managed disks, using the same storage type (standard HDD/SSD, or premium SSD).
    - You can customize the storage type as needed.
- **Replication settings**. Shows the vault in which the VM is located, and the replication policy used for the VM. By default, recovery points created by Site Recovery for the VM are kept for 24 hours.
- **Extension settings**. Indicates that Site Recovery manages updates to the Site Recovery Mobility Service extension that's installed on VMs you replicate.
    - The indicated Azure automation account manages the update process.
    - You can customize the automation account.



7. Select **Review + Start replication**.

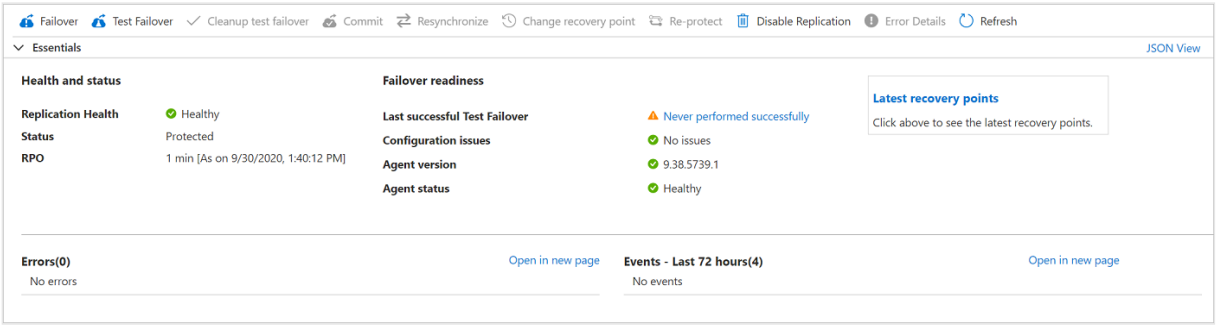
8. Select **Start replication**. Deployment starts, and Site Recovery starts creating target resources. You can monitor replication progress in the notifications.



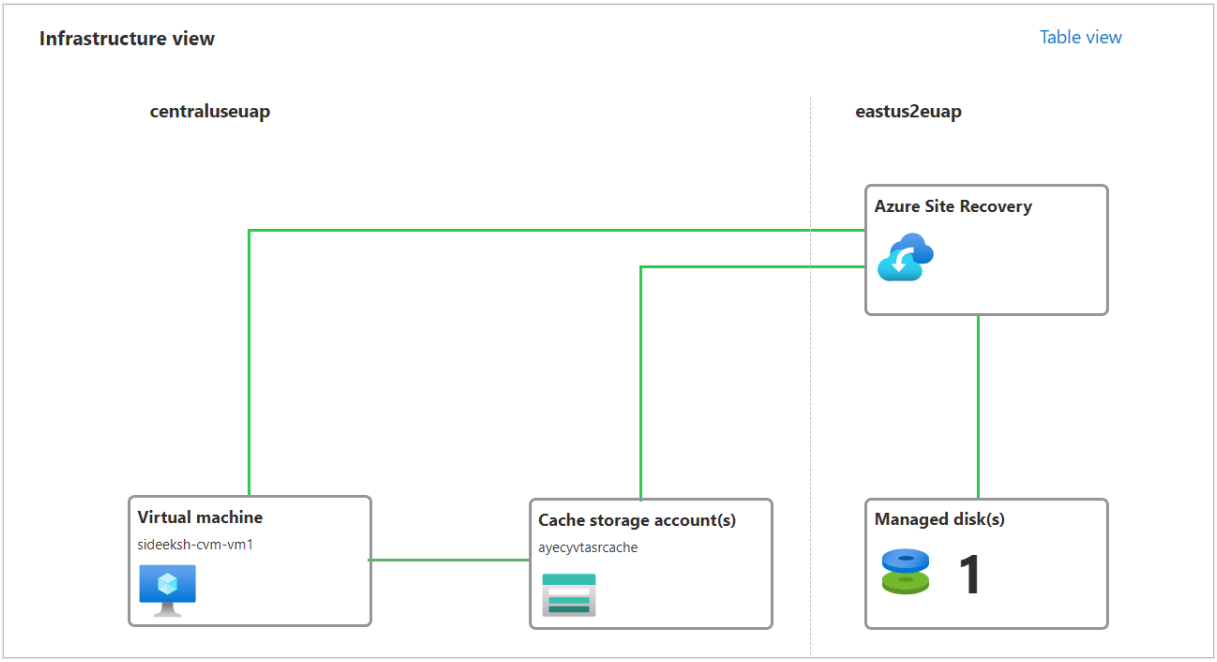
Check VM status

After the replication job finishes, you can check the VM replication status.

1. Open the VM properties page.
2. In **Operations**, select **Disaster recovery**.
3. Expand the **Essentials** section to review defaults about the vault, replication policy, and target settings.
4. In **Health and status**, get information about replication state for the VM, the agent version, failover readiness, and the latest recovery points.



5. In **Infrastructure view**, get a visual overview of source and target VMs, managed disks, and the cache storage account.

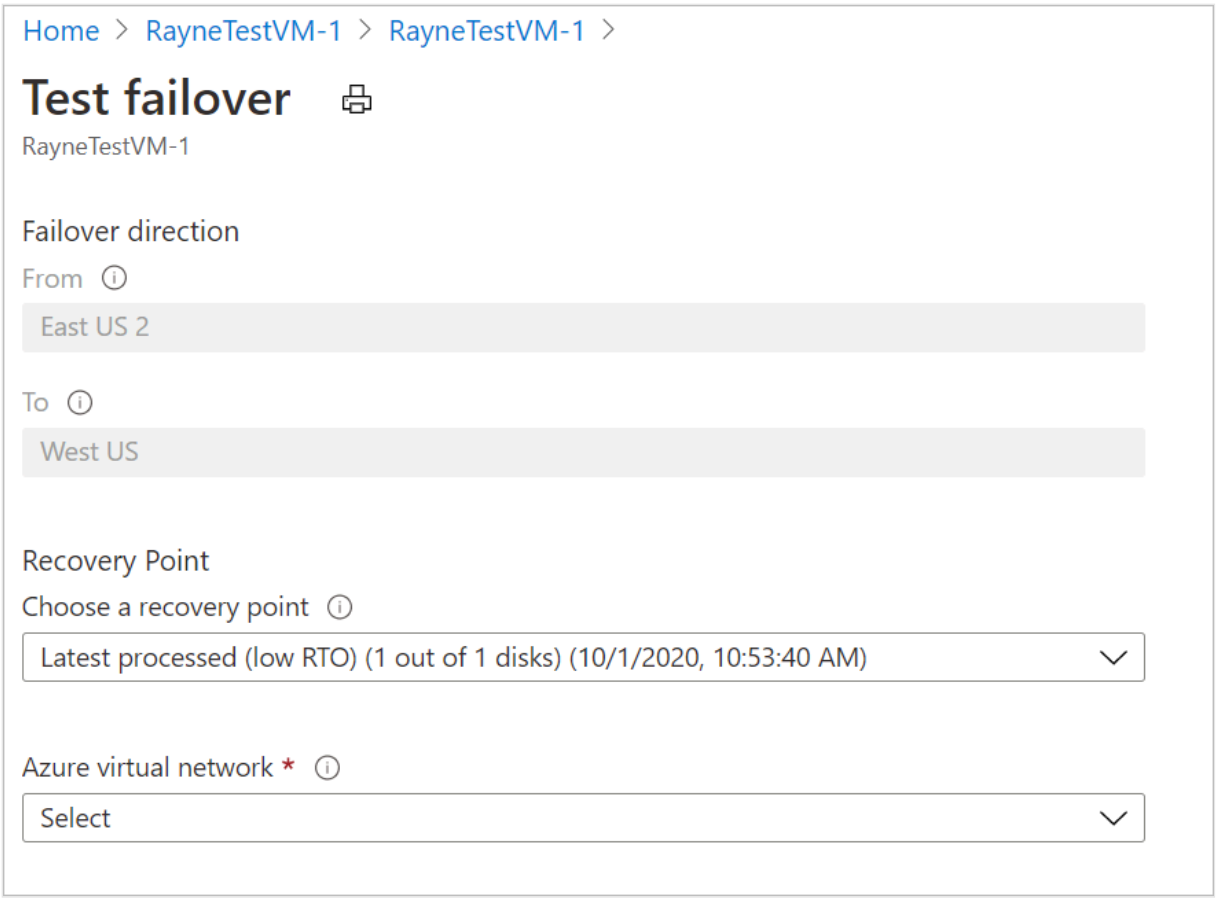


Run a drill

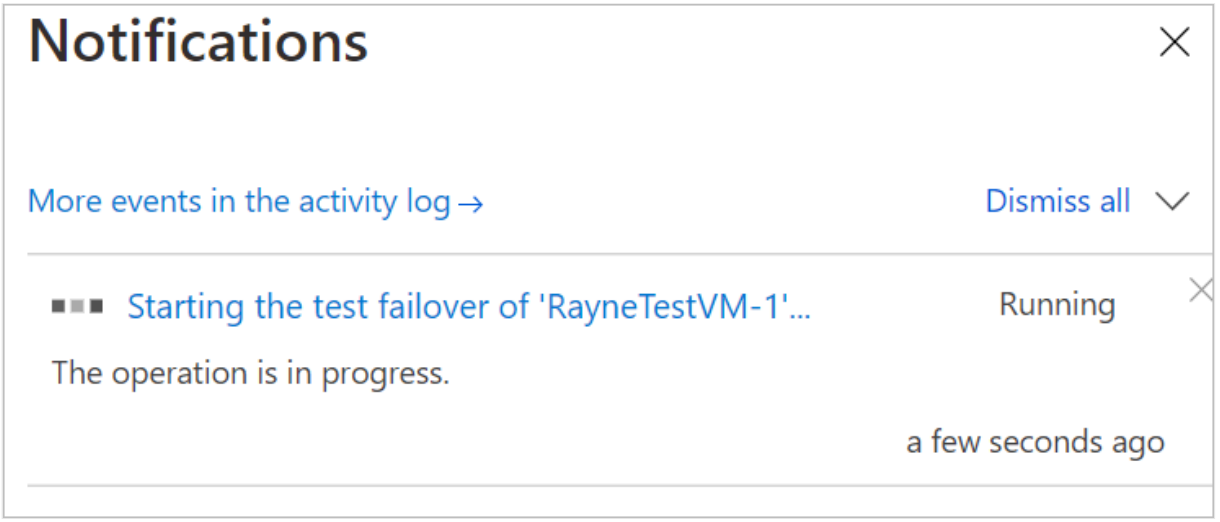
Run a drill to make sure disaster recovery works as expected. When you run a test failover, it creates a copy of the VM, with no impact on ongoing replication, or on your production environment.

1. In the VM disaster recovery page, select **Test failover**.
2. In **Test failover**, leave the default **Latest processed (low RPO)** setting for the recovery point.

   This option provides the lowest recovery point objective (RPO), and generally the quickest spin up of the target VM. It first processes all the data that has been sent to Site Recovery service, to create a recovery point for each VM, before failing over to it. This recovery point has all the data replicated to Site Recovery when the failover was triggered.
3. Select the virtual network in which the VM will be located after failover.



4. The test failover process begins. You can monitor the progress in notifications.

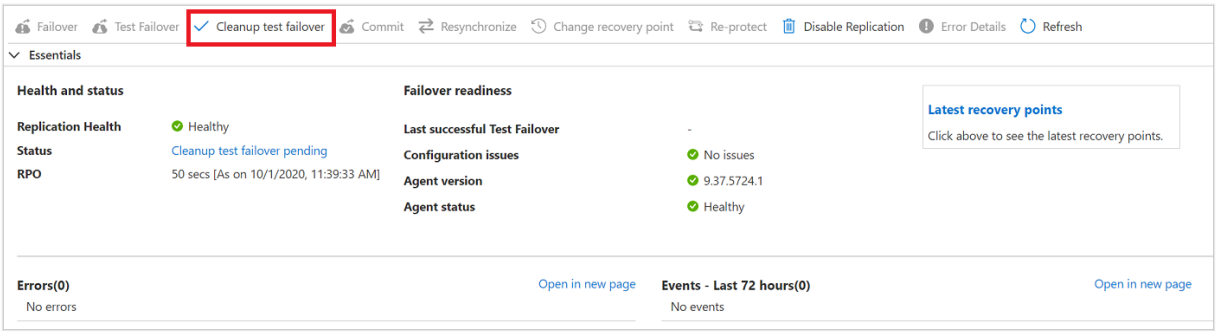


After the test failover completes, the VM is in the *Cleanup test failover pending* state on the **Essentials** page.

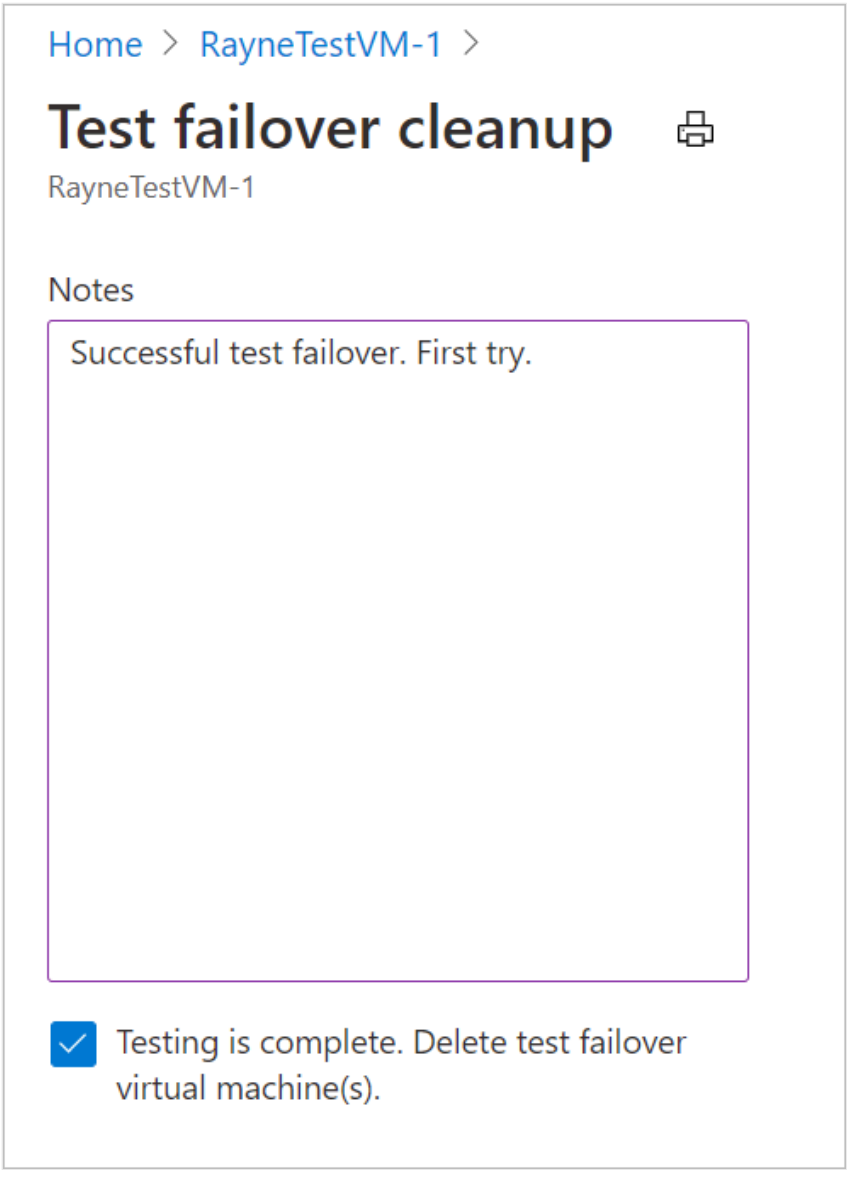
Clean up resources

The VM is automatically cleaned up by Site Recovery after the drill.

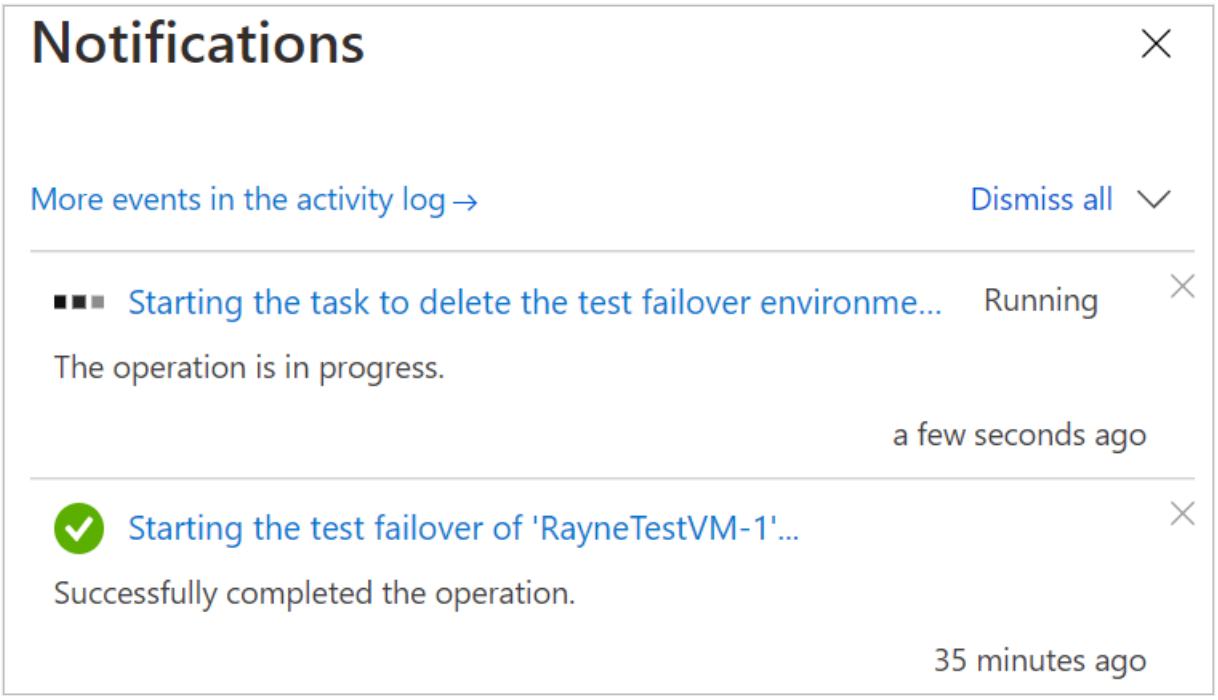
1. To begin automatic cleanup, select **Cleanup test failover**.



2. In **Test failover cleanup**, type in any notes you want to record for the failover, and then select **Testing is complete. Delete test failover virtual machine**. Then select **OK**.



3. The delete process begins. You can monitor progress in notifications.



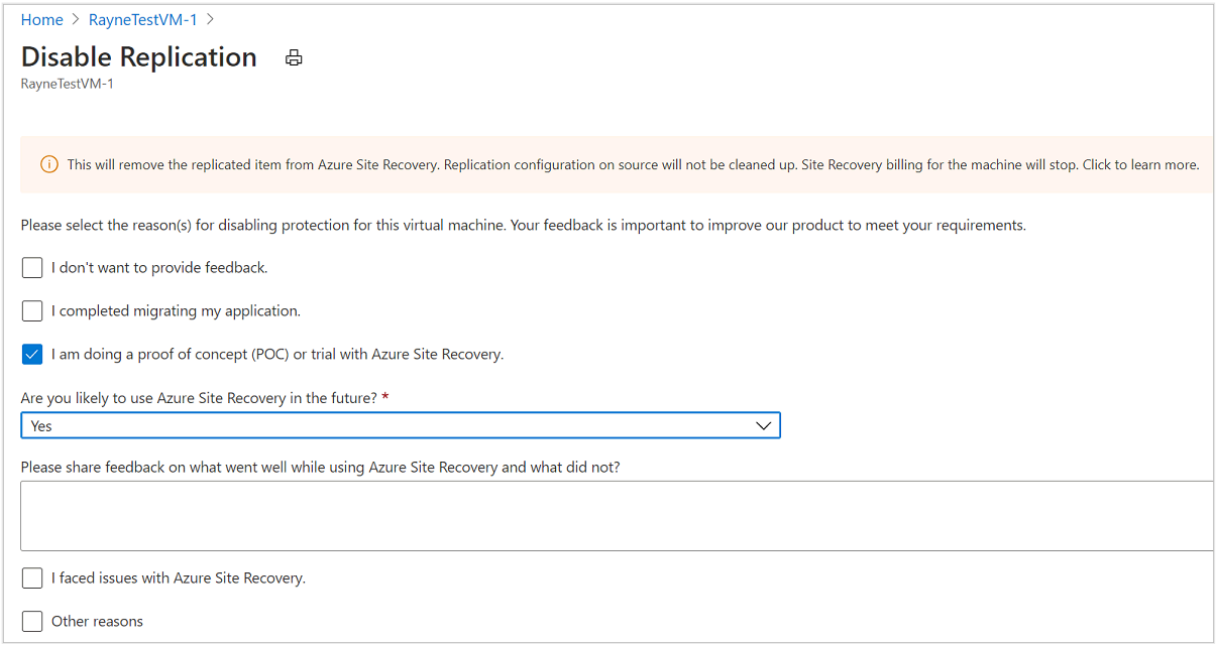
Stop replicating the VM

After completing a disaster recovery drill, we suggest you continue to try out a full failover. If you don't want to do a full failover, you can disable replication. This does the following:

- Removes the VM from the Site Recovery list of replicated machines.
- Stops Site Recovery billing for the VM.
- Automatically cleans up source replication settings.

Stop replication as follows:

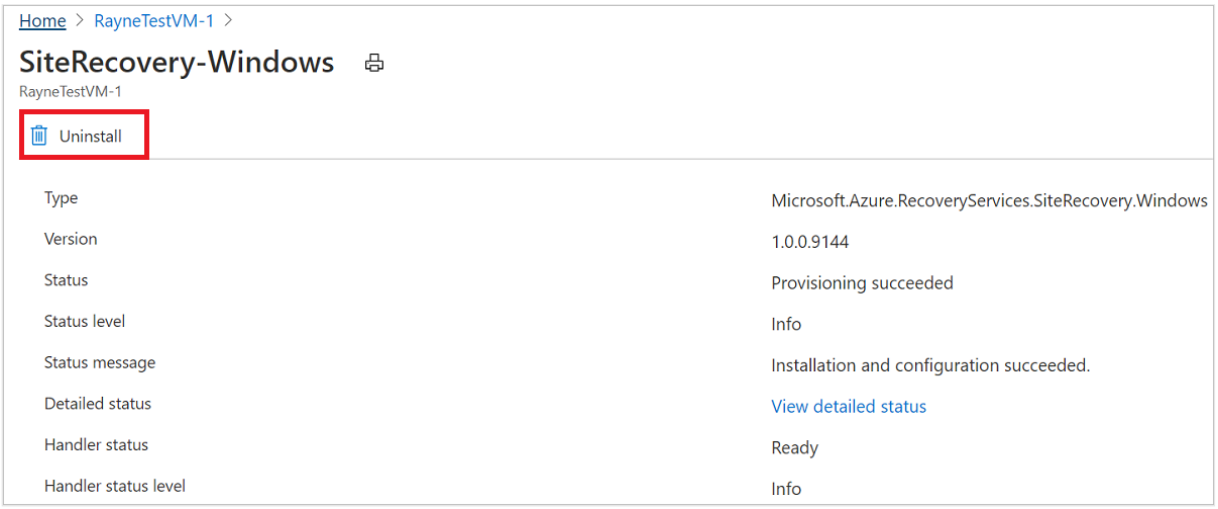
1. In the VM disaster recovery page, select **Disable Replication**.
2. In **Disable Replication**, select the reasons that you want to disable replication. Then select **OK**.



The Site Recovery extension installed on the VM during replication isn't removed automatically. If you disable replication for the VM, and you don't want to replicate it again at a later time, you can remove the

Site Recovery extension manually, as follows:

1. Go to the VM > **Settings** > **Extensions**.
2. In the **Extensions** page, select each *Microsoft.Azure.RecoveryServices* entry for Linux.
3. In the properties page for the extension, select **Uninstall**.



Next steps

In this tutorial, you configured disaster recovery for an Azure VM, and ran a disaster recovery drill. Now, you can perform a full failover for the VM.

Fail over a VM to another region

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Tutorial: Fail over Azure VMs to a secondary region

09/08/2025

Learn how to fail over Azure VMs that are enabled for disaster recovery with Azure Site Recovery, to a secondary Azure region. After failover, you reprotect VMs in the target region so that they replicate back to the primary region. In this article, you learn how to:

- ✓ Check prerequisites
- ✓ Verify VM settings
- ✓ Run a failover to the secondary region
- ✓ Start replicating the VM back to the primary region.

> **Note**
>
> This tutorial shows you how to fail over VMs with minimal steps. If you want to run a failover with full settings, learn about Azure VM networking, automation, and troubleshooting.

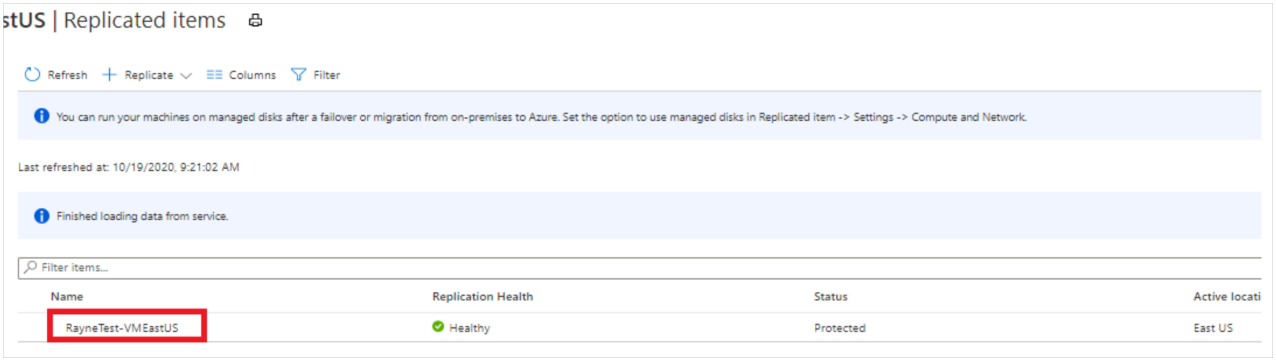
Prerequisites

Before you start this tutorial, you should have:

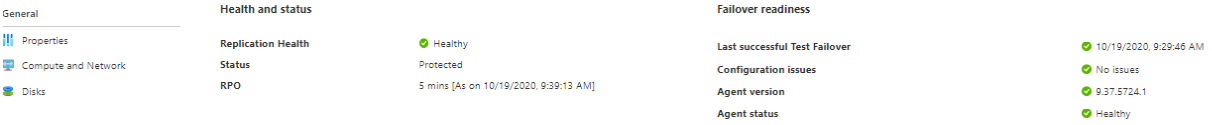
1. Set up replication for one or more Azure VMs. If you haven't, complete the first tutorial in this series to do that.
2. We recommend you run a disaster recovery drill for replicated VMs. Running a drill before you run a full failover helps ensure everything works as expected, without impacting your production environment.

Verify the VM settings

1. In the vault > **Replicated items**, select the VM.



2. On the VM **Overview** page, check that the VM is protected and healthy, before you run a failover.

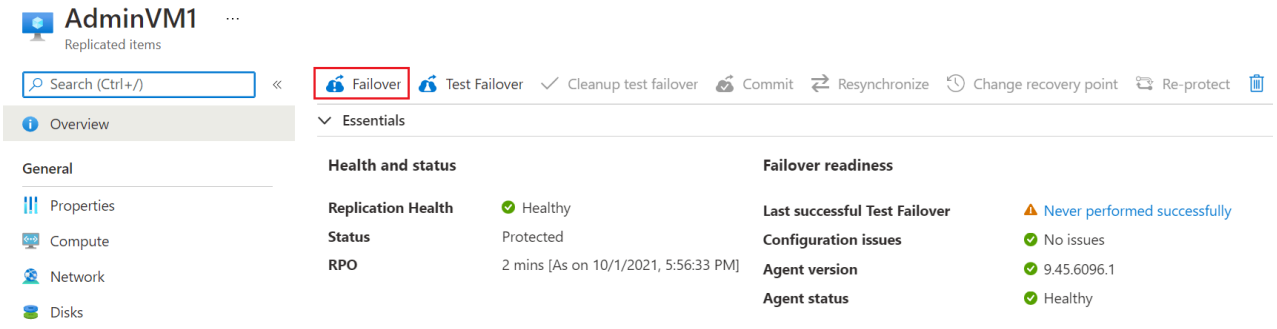


3. Before you fail over, check that:

   - The VM is running a supported Windows or Linux operating system.
   - The VM complies with compute, storage, and networking requirements.

Run a failover

1. On the VM **Overview** page, select **Failover**.



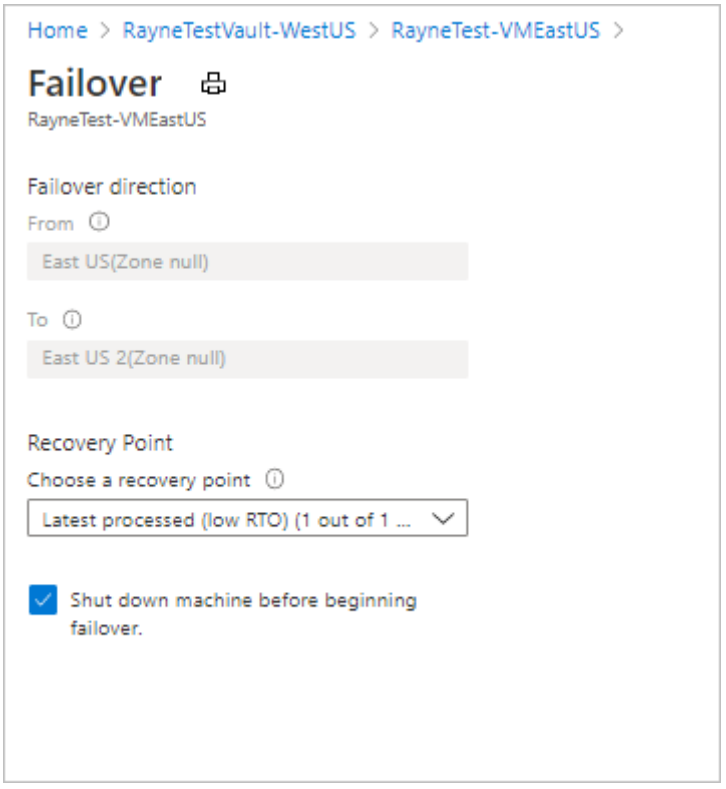
2. In **Failover**, choose a recovery point. The Azure VM in the target region is created using data from this recovery point.

   - **Latest processed**: Uses the latest recovery point processed by Site Recovery. The time stamp is shown. No time is spent processing data, so it provides a low recovery time objective (RTO).
   - **Latest**: Processes all the data sent to Site Recovery, to create a recovery point for each VM before failing over to it. Provides the lowest recovery point objective (RPO), because all data is replicated to Site Recovery when the failover is triggered.
   - **Latest app-consistent**: This option fails over VMs to the latest app-consistent recovery point. The time stamp is shown.

- **Custom**: Fail over to particular recovery point. Custom is only available when you fail over a single VM, and don't use a recovery plan.

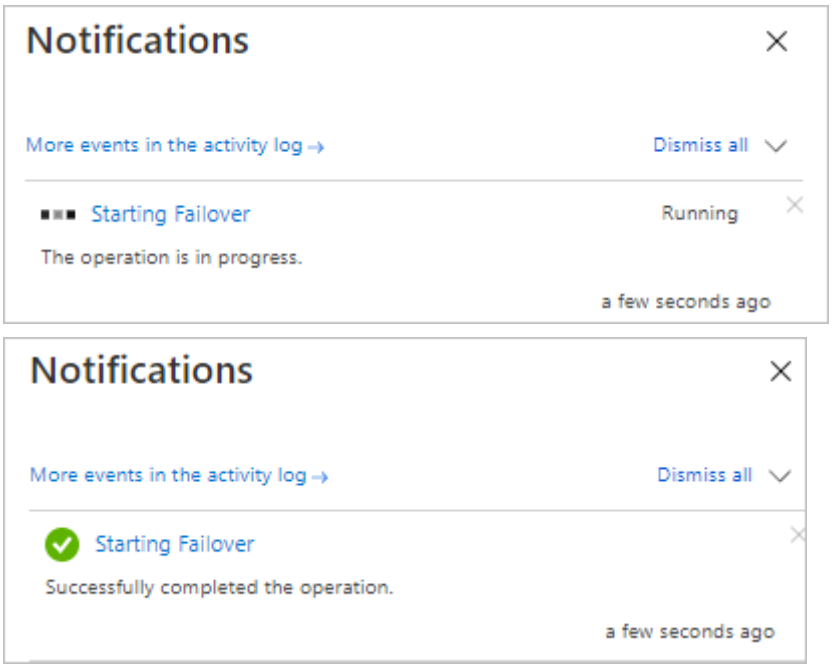
> ⓘ Note
>
> If you added a disk to a VM after you enabled replication, replication points shows disks available for recovery. For example, a replication point created before you added a second disk will show as "1 of 2 disks".

3. Select **Shut down machine before beginning failover** if you want Site Recovery to try to shut down the source VMs before starting failover. Shutdown helps to ensure no data loss. Failover continues even if shutdown fails.



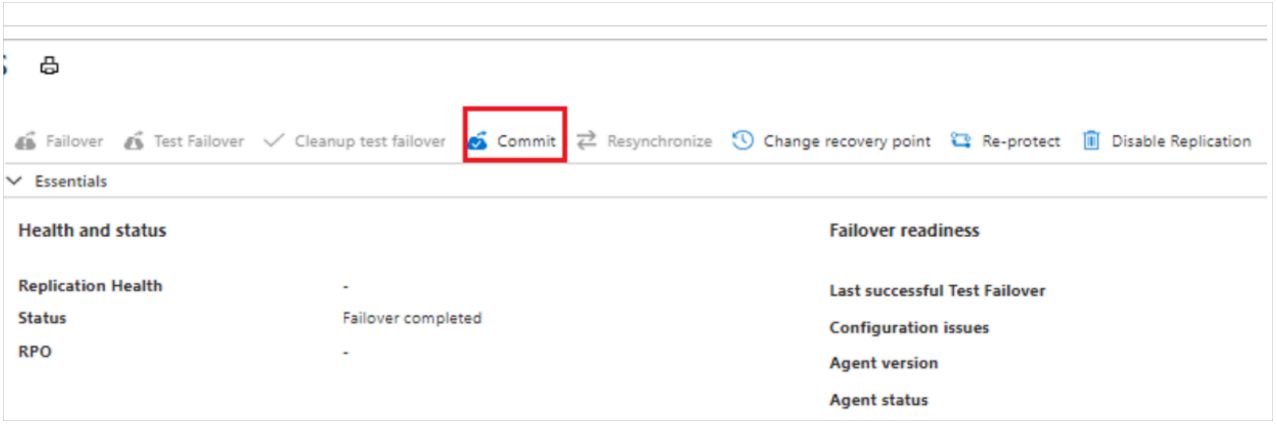
4. To start the failover, select **OK**.

5. Monitor the failover in notifications.



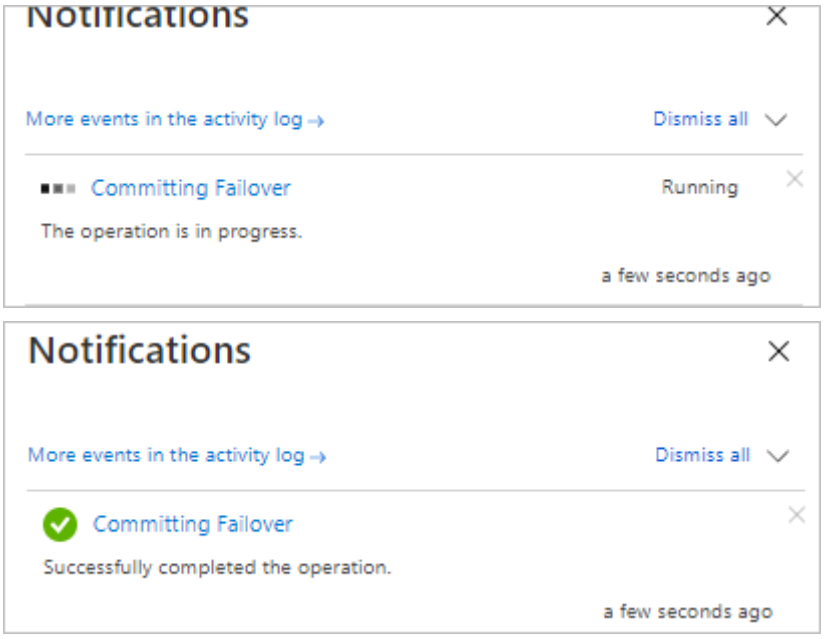
6. After the failover, the Azure VM created in the target region appears in **Virtual Machines**. Make sure that the VM is running, and sized appropriately. If you want to use a different recovery point for the VM, select **Change recovery point**, on the **Essentials** page.

7. When you're satisfied with the failed over VM, select **Commit** on the overview page, to finish the failover.



8. In **Commit**, select **OK** to confirm. Commit deletes all the available recovery points for the VM in Site Recovery, and you won't be able to change the recovery point.

9. Monitor the commit progress in notifications.

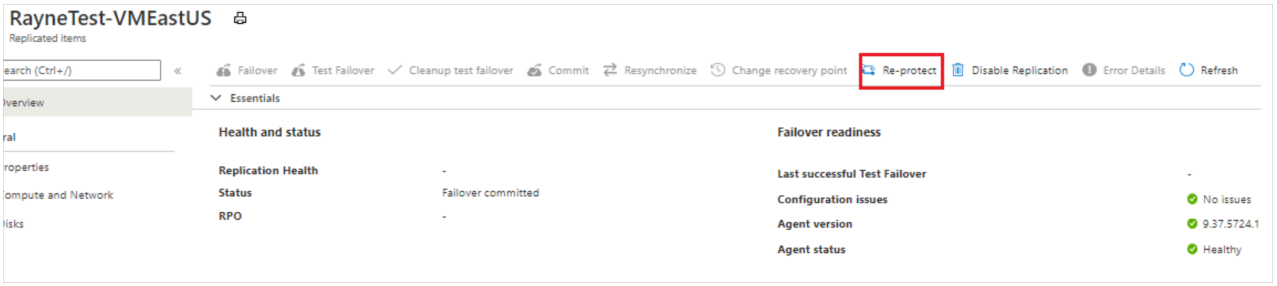


> **Note**
>
> You may also get alternative virtual machine SKU recommendation during test failover. Learn more.

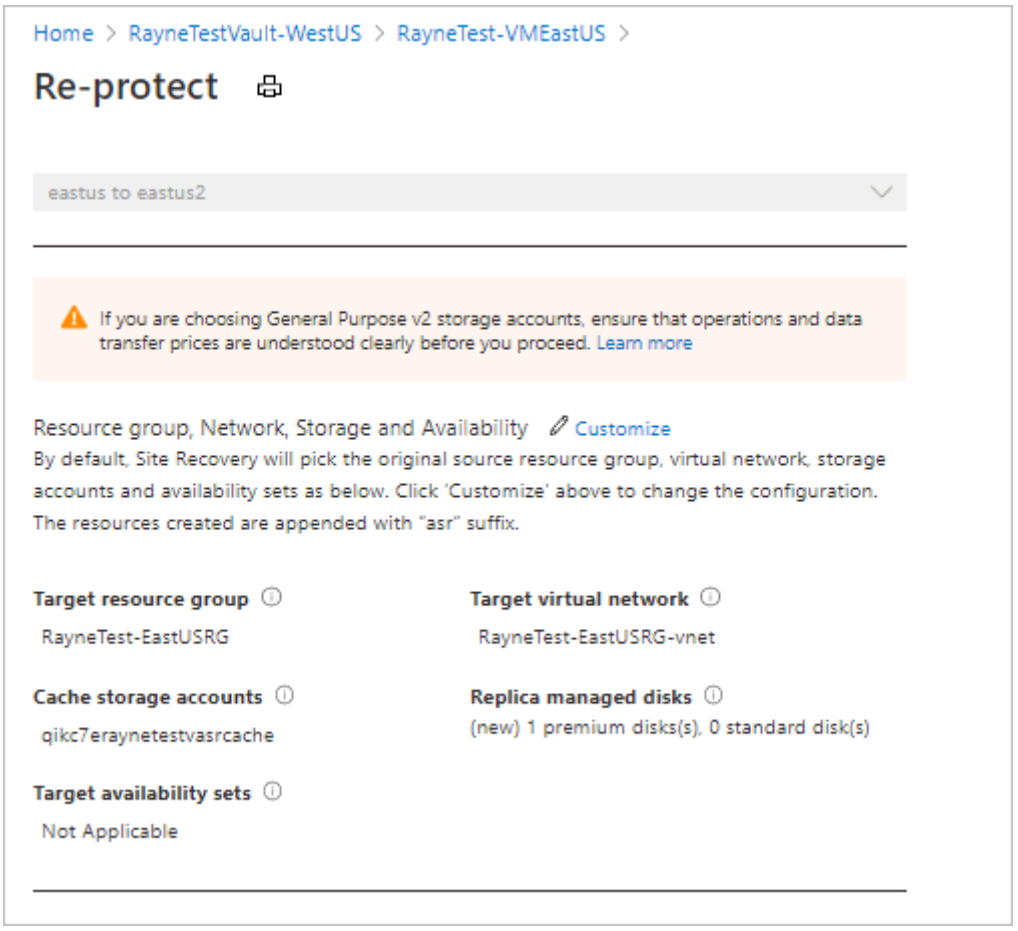
Reprotect the VM

After failover, you reprotect the VM in the secondary region, so that it replicates back to the primary region.

1. Make sure that VM **Status** is *Failover committed* before you start.
2. Check that you can access the primary region is available, and that you have permissions to create VMs in it.
3. On the VM **Overview** page, select **Re-Protect**.

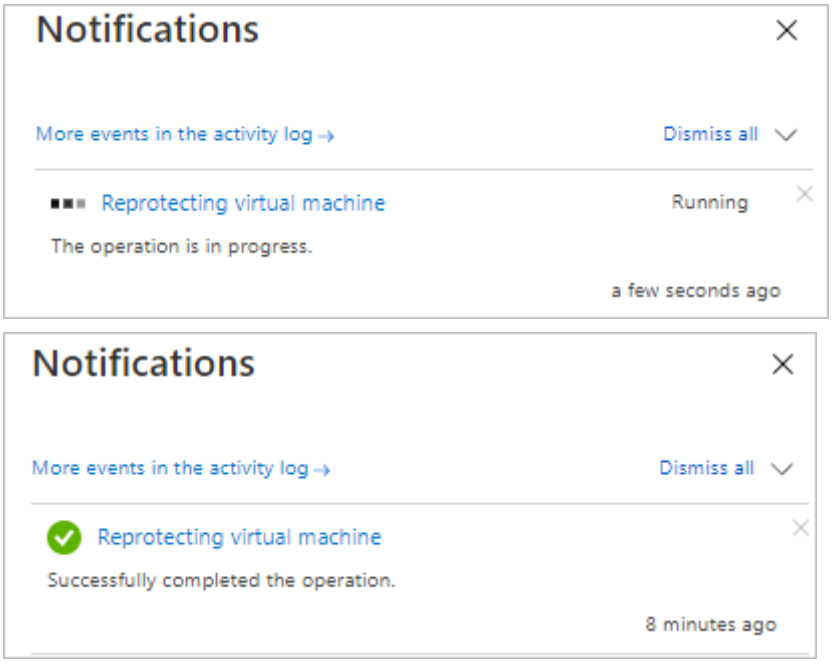


4. In **Re-protect**, verify the replication direction (secondary to primary region), and review the target settings for the primary region. Resources marked as new are created by Site Recovery as part of the reprotect operation.



5. Select **OK** to start the reprotect process. The process sends initial data to the target location, and then replicates delta information for the VMs to the target.

6. Monitor reprotect progress in the notifications.



Next steps

In this tutorial, you failed over from the primary region to the secondary, and started replicating VMs back to the primary region. Now you can fail back from the secondary region to the primary.

Tutorial: Fail back Azure VM to the primary region

Article • 02/11/2025

After failing over an Azure VM to a secondary Azure region, follow this tutorial to fail the VM to the primary Azure region, using Azure Site Recovery. In this article, you learn how to:

- ✓ Review the prerequisites.
- ✓ Fail back the VM in the secondary region.
- ✓ Reprotect primary VMs back to the secondary region.

> **Note**
>
> This tutorial shows you how to fail back with minimal steps. If you want to run a failover with full settings, learn about Azure VM networking, automation, and troubleshooting.

Prerequisites

Before you start this tutorial, you should have:

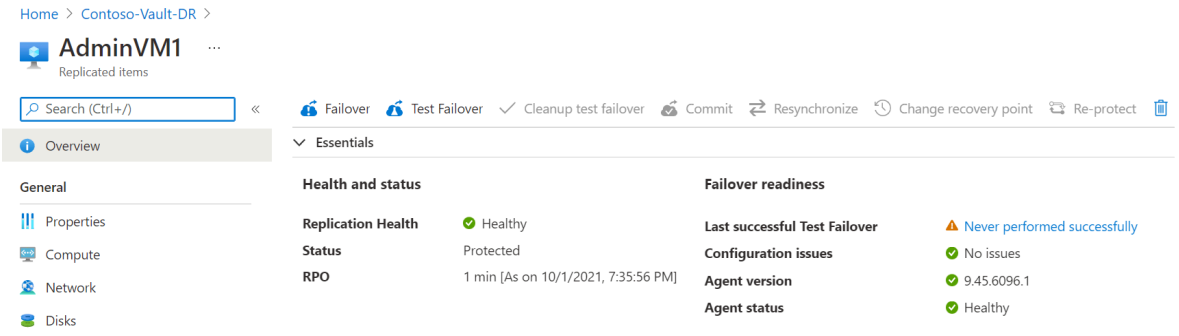
1. Set up replication for at least one Azure VM, and tried out a disaster recovery drill for it.
2. Failed over the VM from the primary region to a secondary region, and reprotected it so that it replicates from the secondary region to the primary.
3. Check that the primary region is available, and that you're able to create and access new resources in it.
4. Source region VM must be shut down before attempting a failback since, changes are synced from failover VM disks to the source region disks during failback. If the source VM is powered on then it fails and thus fails the failback operation.

Fail back to the primary region

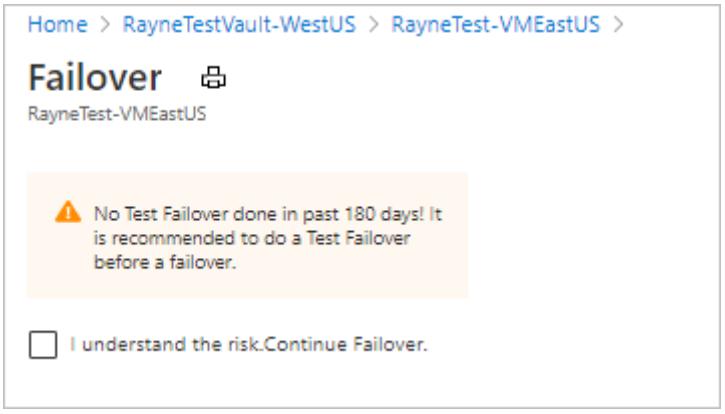
After VMs are reprotected, you can fail back to the primary region as needed.

1. In the vault > **Replicated items**, select the VM.

2. On the VM overview page, check that the VM is healthy, and that synchronization is complete, before you run a failover. The VM should be in a *Protected* state.



3. On the overview page, select **Failover**. Since we're not doing a test failover this time, we're prompted to verify.



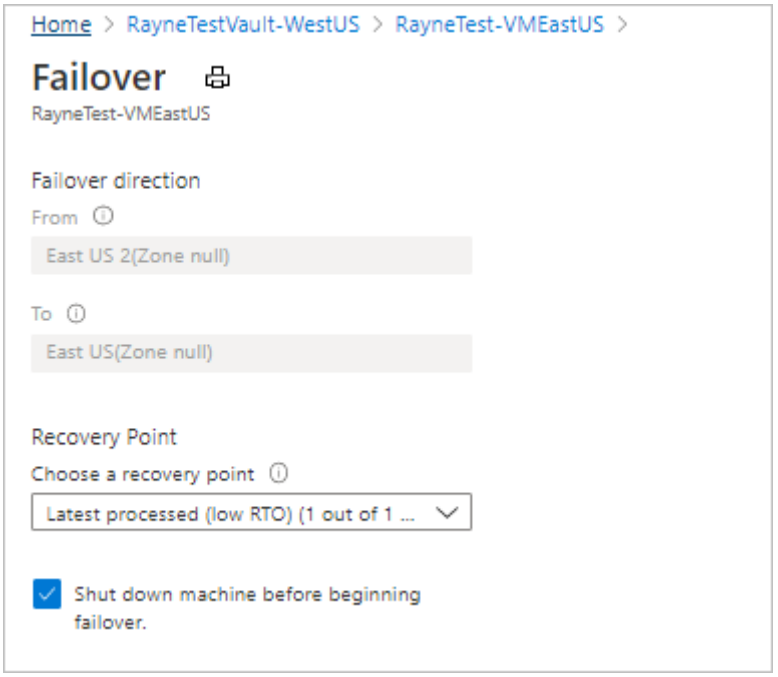
4. In **Failover**, note the direction from secondary to primary, and select a recovery point. The Azure VM in the target (primary region) is created using data from this point.

    - **Latest processed**: Uses the latest recovery point processed by Site Recovery. The time stamp is shown. No time is spent processing data, so it provides a low recovery time objective (RTO).
    - **Latest**: Processes all the data sent to Site Recovery, to create a recovery point for each VM before failing over to it. Provides the lowest recovery point objective (RPO), because all data is replicated to Site Recovery when the failover is triggered.
    - **Latest app-consistent**: This option fails over VMs to the latest app-consistent recovery point. The time stamp is shown.
    - **Custom**: Fail over to particular recovery point. Custom is only available when you fail over a single VM, and don't use a recovery plan.

> ⓘ Note

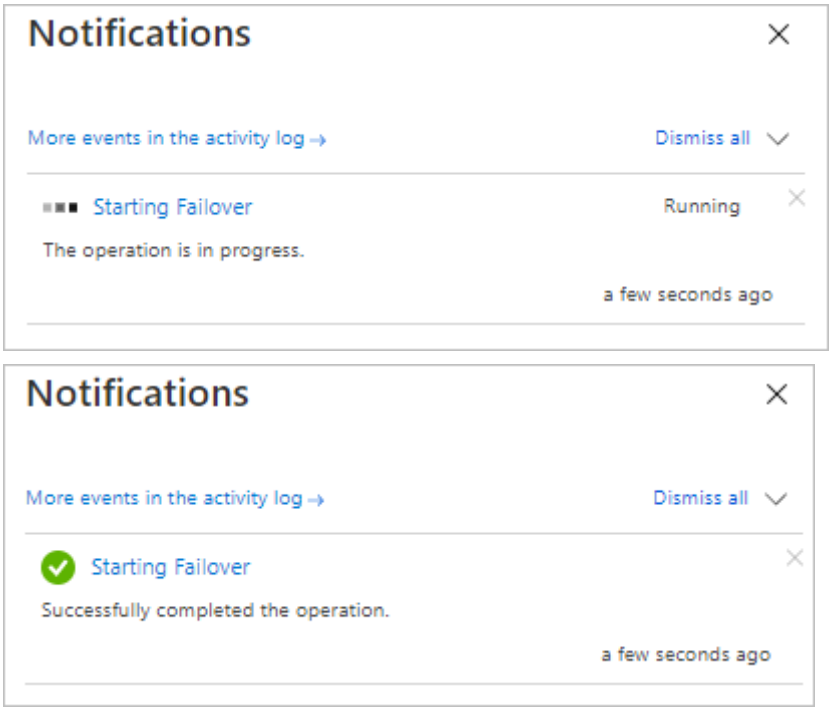
> If you fail over a VM to which you added a disk after you enabled replication for the VM, replication points will show the disks available for recovery. For example, a replication point that was created before you added a second disk will show as "1 of 2 disks".

5. Select **Shut down machine before beginning failover** if you want Site Recovery to attempt to shut down the source VMs before starting failover. Shutdown helps to ensure no data loss. Failover continues even if shutdown fails.



6. To start the failover, select **OK**.

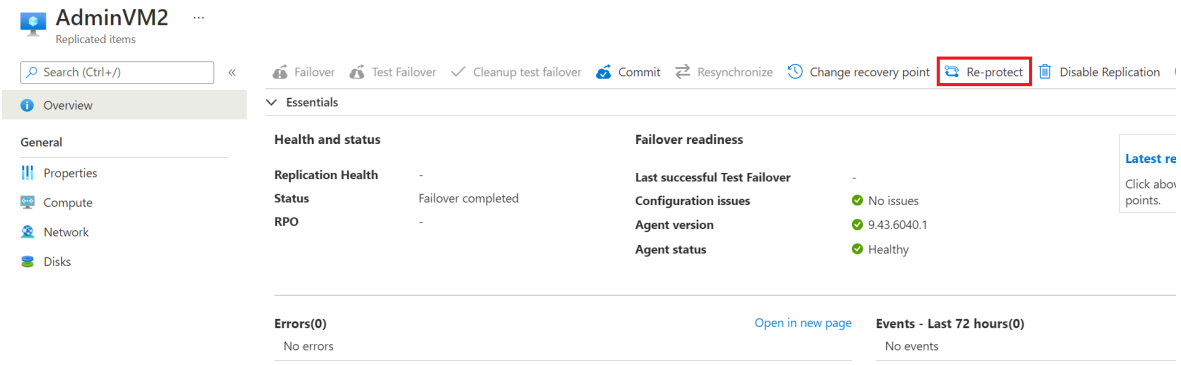
7. Monitor the failover in notifications.



Reprotect VMs

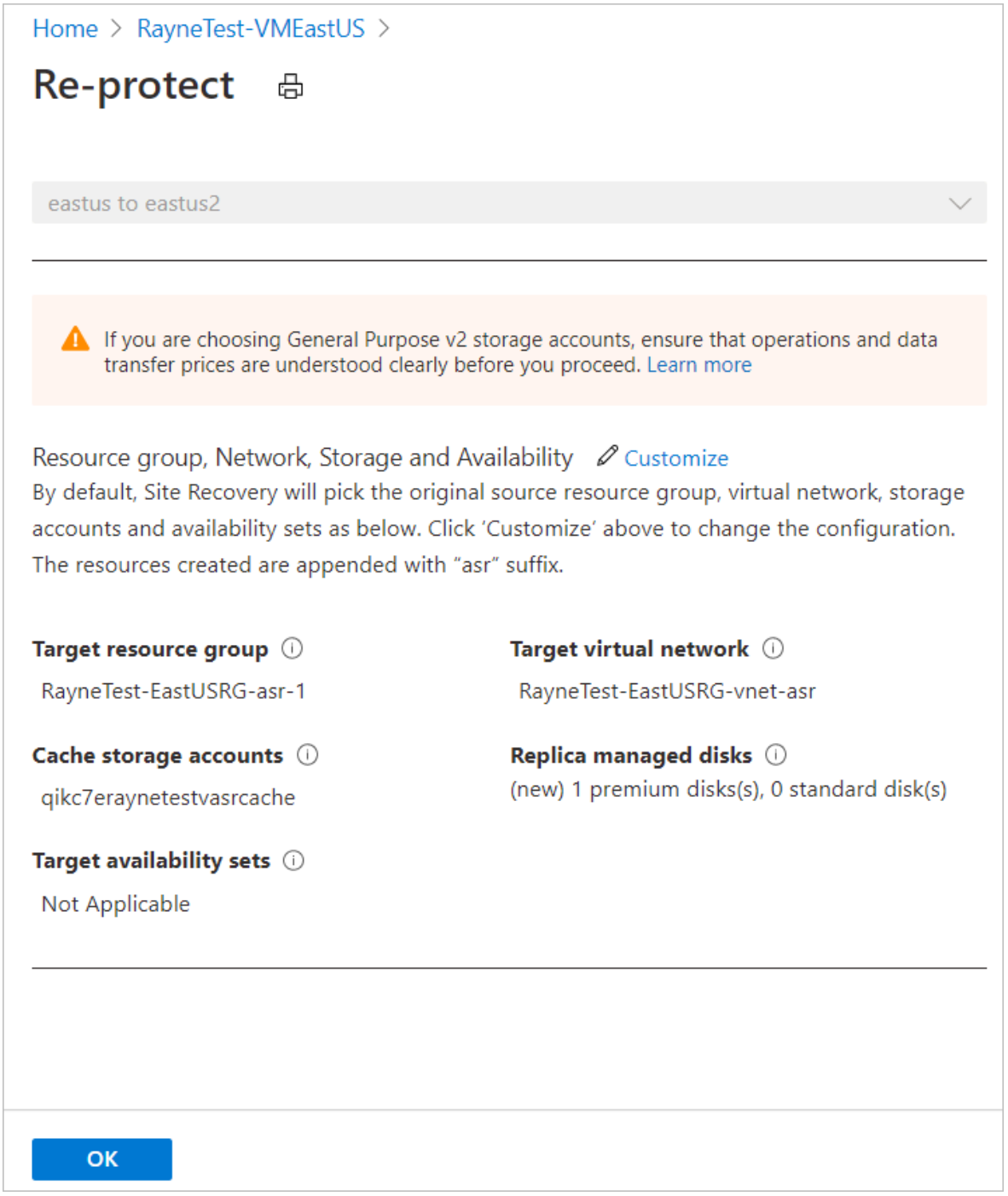
After failing back VMs to the primary region, you need to reprotect them, so that they start replicating to the secondary region again.

1. In the **Overview** page for the VM, select **Re-protect**.

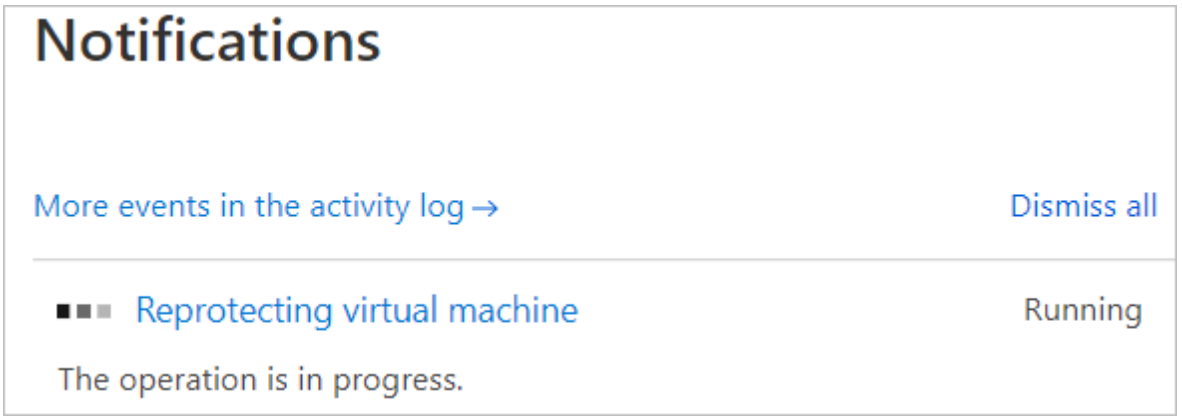


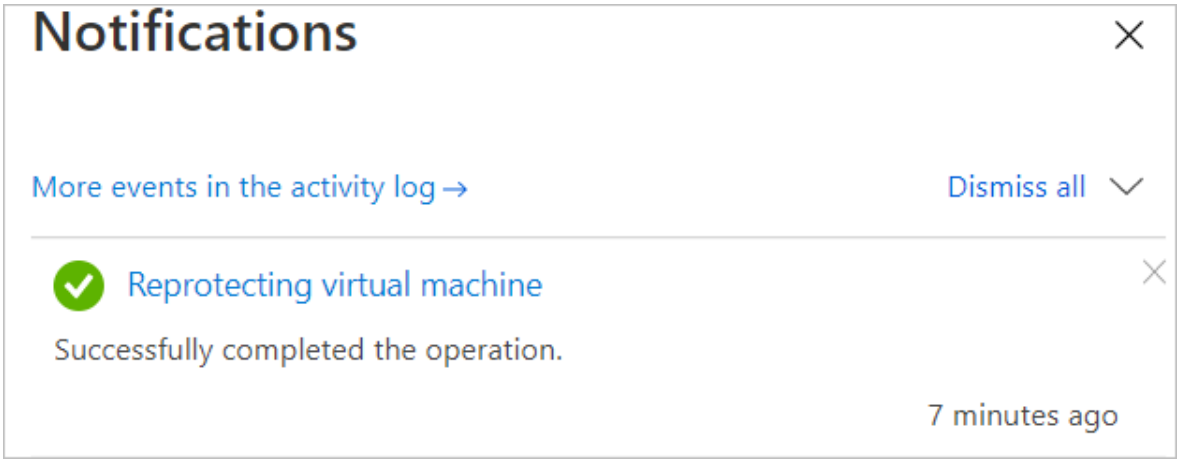
2. Review the target settings for the primary region. Resources marked as new are created by Site Recovery as part of the reprotect operation.

3. Select **OK** to start the reprotect process. The process sends initial data to the target location, and then replicates delta information for the VMs to the target.



4. Monitor reprotect progress in notifications.





Clean up resources

For VMs with managed disks, after failback is complete and VMs are reprotected for replication from primary to secondary, Site Recovery automatically cleans up machines in the secondary disaster recovery region. You don't need to manually delete VMs and NICs in the secondary region. VMs with unmanaged disks aren't cleaned up.

If you completely disable replication after failing back, Site Recovery cleans up machines protected by it. In this case, it also cleans up disks for VMs that don't use managed disks.

Next steps

In this tutorial, you failed VMs back from the secondary region to the primary. This is the last step in the process that includes enabling replication for a VM, trying out a disaster recovery drill, failing over from the primary region to the secondary, and finally failing back.

Now, try out disaster recovery to Azure for an on-premises VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Overview of VM restore points

Summarize this article for me

Business continuity and disaster recovery solutions are primarily designed to address sitewide data loss. Solutions that operate at this scale often manage and run across automated failovers and failbacks in multiple regions. You can use Azure virtual machine (VM) restore points to implement granular backup and retention policies.

You can protect your data and guard against extended downtime by creating VM restore points at regular intervals. There are several backup options available for VMs, depending on your use case. For more information, see Backup and restore options for VMs in Azure.

About VM restore points

An individual VM restore point is a resource that stores VM configuration and point-in-time application-consistent snapshots of all the managed disks attached to the VM. You can use VM restore points to easily capture multidisk-consistent backups. VM restore points contain a disk restore point for each of the attached disks. A disk restore point consists of a snapshot of an individual managed disk.

VM restore points support both application consistency and crash consistency.

Application consistency is supported for VMs running Windows operating systems and support file system consistency for VMs running Linux operating systems. Application-consistent restore points use Volume Shadow Copy Service (VSS) writers (or pre- and postscripts for Linux) to ensure the consistency of the application data before a restore point is created. To get an application-consistent restore point, the application running in the VM needs to provide a VSS writer (for Windows) or pre- and postscripts (for Linux) to achieve application consistency.

A multidisk crash-consistent VM restore point stores the VM configuration and point-in-time write-order-consistent snapshots for all managed disks attached to a VM. This information is the same as the status of data in the VM after a power outage or a crash. The `consistencyMode` optional parameter has to be set to `crashConsistent` in the creation request.

> **Note**
>
> Multidisk crash consistency can't be guaranteed in below scenarios:
>
> - For disks configured with read/write host caching, multidisk crash consistency can't be guaranteed because writes that occur while the snapshot is taken might not be

acknowledged by Azure Storage. If maintaining consistency is crucial, we recommend that you use the application-consistency mode.

- Crash-consistent restore points created for Intel V6+ (Dsv6-series, Edsv6-series, Esv6-series etc.) and AMD V7+ (Dasv7-series, Dadsv7-series, Easv7-series, Faldsv7-series etc.) Virtual Machines (VMs) with more than one data disk may not be consistent across disks. These SKUs use Azure Boost, which offloads storage operations to hardware for performance. For VM SKUs using the latest gen of Azure boost, current Crash consistent restore point orchestration does not guarantee simultaneous snapshots across multiple disks, leading to timing gaps (sub-second skew) between disks.

VM restore points are organized into restore point collections. A restore point collection is an Azure Resource Manager resource that contains the restore points for a specific VM. If you want to utilize Azure Resource Manager templates (ARM templates) for creating restore points and restore point collections, see the public Virtual-Machine-Restore-Points repository in GitHub.

The following image illustrates the relationship between restore point collections, VM restore points, and disk restore points.



VM restore points are incremental. The first restore point stores a full copy of all disks attached to the VM. For each successive restore point for a VM, only the incremental changes to your disks are backed up. To reduce your costs, you can optionally exclude any disk when you create a restore point for your VM.

Restore points for VMs inside virtual machine scale set and availability set (AvSet)

Currently, you can create restore points in only one VM at a time. You can't create a single restore point across multiple VMs. Because of this limitation, we currently support creating restore points for individual VMs with a virtual machine scale set in Flexible Orchestration mode or an availability set. If you want to back up instances within a virtual machine scale set instance or your availability set instance, you must individually create restore points for all the VMs that are part of the instance.

> **Note**
>
> A virtual machine scale set with Uniform orchestration isn't supported by restore points. You can't create restore points of VMs inside a virtual machine scale set with Uniform orchestration.

Throttling limits for Restore points

Expand table

Scope	Operation	Limit per hour
VM	RestorePoints.RestorePointOperation.PUT (Create new **Application Consistent**)	3
VM	RestorePoints.RestorePointOperation.PUT (Create new **Crash Consistent**)	3
Target restore point collection	RestorePoints.RestorePointOperation.PUT (Copy any VM Restore Point)	3

The throttling would fail with error code 429. Please retry the operation after the specified duration.

Limitations

- Restore points are supported only for managed disks.
- Ultra Disks, Premium SSD v2 disks, Write-accelerated disks, Ephemeral OS disks, and shared disks aren't supported for crash consistency mode.
- Ephemeral OS disks, and shared disks aren't supported for application consistency mode.
- The API version for an application-consistent restore point is 2021-03-01, or later.
- The API version for a crash-consistent restore point is 2021-07-01, or later.
- Crash consistent restore point is supported on all Azure public regions.

- Crash consistent restore point is supported for VM sizes that support Premium storage like DSv2.
- A maximum of 500 VM restore points can be retained at any time for a VM, irrespective of the number of restore point collections or consistency type.
- Concurrent creation of restore points for a VM isn't supported.
- Restore points for virtual machine scale sets in Uniform orchestration mode aren't supported.
- Movement of VMs between resource groups or subscriptions isn't supported when the VM has restore points. Moving the VM between resource groups or subscriptions doesn't update the source VM reference in the restore point and causes a mismatch of Resource Manager IDs between the actual VM and the restore points.

> **Note**
>
> Public preview of cross-region copying of VM restore points is available, with the following limitations:
>
> - Private links aren't supported when you copy restore points across regions or create restore points in a region other than the source VM.
> - Customer-managed key encrypted restore points, when copied to a target region, are created as platform-managed key encrypted restore points.

Troubleshoot VM restore points

Most common restore point failures are attributed to the communication with the VM agent and extension. To resolve failures, follow the steps in Troubleshoot restore point failures.

Next steps

- Create a VM restore point.
- Learn more about backup and restore options for VMs in Azure.
- Learn more about the extensions used with application consistency mode.
- Learn more about how to copy VM restore points across regions.

Last updated on 02/13/2026

Quickstart: Create VM restore points using APIs

Article • 08/22/2024

You can protect your data by taking backups at regular intervals. Azure VM restore point APIs are a lightweight option you can use to implement granular backup and retention policies. VM restore points support application consistency for VMs running Windows operating systems and support file system consistency for VMs running Linux operating system.

You can use the APIs to create restore points for your source VM in either the same region, or in other regions. You can also copy existing VM restore points between regions.

Prerequisites

- Learn more about the requirements for a VM restore point.
- Consider the limitations before creating a restore point.

Create VM restore points

The following sections outline the steps you need to take to create VM restore points with the Azure Compute REST APIs.

You can find more information in the Restore Points, PowerShell, and Restore Point Collections API documentation.

Step 1: Create a VM restore point collection

Before you create VM restore points, you must create a restore point collection. A restore point collection holds all the restore points for a specific VM. Depending on your needs, you can create VM restore points in the same region as the VM, or in a different region. To create a restore point collection, call the restore point collection's Create or Update API.

- If you're creating restore point collection in the same region as the VM, then specify the VM's region in the location property of the request body.
- If you're creating the restore point collection in a different region than the VM, specify the target region for the collection in the location property, but also specify

the source restore point collection ARM resource ID in the request body.

To create a restore point collection, call the restore point collection's Create or Update API.

Step 2: Create a VM restore point

After you create the restore point collection, the next step is to create a VM restore point within the restore point collection. For more information about restore point creation, see the Restore Points - Create API documentation. For creating crash consistent restore points (in preview) "consistencyMode" property has to be set to "CrashConsistent" in the creation request.

> **Tip**
>
> To save space and costs, you can exclude any disk from either local region or cross-region VM restore points. To exclude a disk, add its identifier to the `excludeDisks` property in the request body.

Step 3: Track the status of the VM restore point creation

Restore point creation in your local region will be completed within a few seconds. Scenarios, which involve the creation of cross-region restore points will take considerably longer. To track the status of the creation operation, follow the guidance in Get restore point copy or replication status. This is only applicable for scenarios where the restore points are created in a different region than the source VM.

Get restore point copy or replication status

Copying the first VM restore point to another region is a long running operation. The VM restore point can be used to restore a VM only after the operation is completed for all disk restore points. To track the operation's status, call the Restore Point - Get API on the target VM restore point and include the `instanceView` parameter. The return will include the percentage of data that has been copied at the time of the request.

During restore point creation, the `ProvisioningState` will appear as `Creating` in the response. If creation fails, `ProvisioningState` is set to `Failed`.

Next steps

- Learn more about managing restore points.
- Create restore points using the Azure portal, CLI, or PowerShell.
- Learn more about Backup and restore options for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

VMSnapshot extension

Article • 08/22/2024

Application-consistent restore points use the Volume Shadow Copy Service (VSS) service (or pre-/postscripts for Linux) to verify application data consistency prior to creating a restore point. Achieving an application-consistent restore point involves the VM's running application providing a VSS service (for Windows) or pre- and postscripts (for Linux).

For Windows images, the **VMSnapshot Windows** extension, and for Linux images, the **VMSnapshot Linux** extension, are used for taking application consistent restore points. When there's a create application consistent restore point request issued from a VM, Azure installs the VM snapshot extension if it's not already present. The extension is updated automatically.

> **Important**
>
> Azure begins creating a restore point only after all extensions (including but not limited to VMSnapshot) provisioning state are complete.

Extension logs

You can view logs for the VMSnapshot extension on the VM under `C:\WindowsAzure\Logs\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot` for Windows and under `/var/log/azure/Microsoft.Azure.RecoveryServices.VMSnapshotLinux/extension.log` for Linux.

Troubleshooting

Most common restore point failures are attributed to the communication with the VM agent and extension. To resolve failures, follow the steps in Troubleshoot restore point failures.

During certain VSS writer failure, Azure takes file system-consistent restore points for the next three times (irrespective of the frequency at which the restore point creation is scheduled) upon failing the initial creation request. From the fourth time onward, an application-consistent restore point is attempted.

Follow these steps to troubleshoot VSS writer issues.

> **Note**
>
> Avoid manually deleting the extension because it leads to failure of the subsequent creation of an application-consistent restore point.

Next steps

Create a VM restore point

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Overview of cross-region copy of VM restore points (in preview)

Article • 08/22/2024

Azure virtual machine (VM) restore point APIs are a lightweight option you can use to implement granular backup and retention policies. VM restore points support application consistency and crash consistency (in preview). You can copy a VM restore point from one region to another region. This capability can help you build business continuity and disaster recovery solutions for Azure VMs.

Scenarios where this API can be helpful:

- Extend multiple copies of restore points to different regions.
- Extend local restore point solutions to support disaster recovery from region failures.

> ⓘ **Note**
>
> To copy a restore point across a region, you need to precreate a `RestorePointCollection` resource in the target region.

Limitations

- Private links aren't supported when you copy restore points across regions or create restore points in a region other than the source VM.
- Azure confidential VMs aren't supported.
- The API version for a cross-region copy of the VM restore point feature is 2021-03-01, or later.
- Copy of a copy isn't supported. You can't copy a restore point that was already copied from another region. For example, if you copied RP1 from East US to West US as RRP1, you can't copy RRP1 from West US to another region (or back to East US).
- Multiple copies of the same restore point in a single target region aren't supported. A single restore point in the source region can be copied only once to a target region.
- Copying a restore point that's CMK encrypted in the source is encrypted by using CMK in the target region. This feature is currently in preview.
- The target restore point only shows the creation time when the source restore point was created.

- Currently, the replication progress is updated every 10 minutes. For disks that have low churn, there can be scenarios where only the initial (0) and the final replication progress (100) appear.
- The maximum copy time supported is two weeks. For a huge amount of data to be copied to a target region, depending on the bandwidth available between the regions, the copy time could be a couple of days. If the copy time exceeds two weeks, the copy operation is terminated automatically.
- No error details are provided when a disk restore point copy fails.
- When a disk restore point copy fails, the intermediate completion percentage where the copy failed isn't shown.
- Restoring a disk from the restore point doesn't automatically check if the disk restore point replication is completed. You need to manually check the percent completion if the replication status is 100% and then start restoring the disk.
- Restore points that are copied to the target region don't have a reference to the source VM. They have a reference to the source restore points. So, if the source restore point is deleted, there's no way to identify the source VM by using the target restore points.
- Copying restore points in a nonsequential order isn't supported. For example, you might have the three restore points RP1, RP2, and RP3. If you already successfully copied RP1 and RP3, you aren't allowed to copy RP2.
- The full snapshot on the source side should always exist and can't be deleted to save cost. For example, if RP1 (full snapshot), RP2 (incremental), and RP3 (incremental) exist in the source and are successfully copied to the target, you can delete RP2 and RP3 on the source side to save cost. Deleting RP1 on the source side results in creating a full snapshot, say RRP1, the next time, and copying also results in a full snapshot. The storage layer maintains the relationship with each pair of source and target snapshot that needs to be preserved.

Troubleshoot VM restore points

Most common restore point failures are attributed to the communication with the VM agent and extension. To resolve failures, follow the steps in Troubleshoot restore point failures.

Next steps

- Copy a VM restore point.
- Learn more about backup and restore options for VMs in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Manage VM restore points

Article • 08/22/2024

This article explains how to copy and restore a VM from a VM restore point and track the progress of the copy operation. This article also explains how to create a disk from a disk restore point and to create a shared access signature for a disk.

Copy a VM restore point between regions

The VM restore point APIs can be used to restore a VM in a different region than the source VM. Use the following steps:

Step 1: Create a destination VM restore point collection

To copy an existing VM restore point from one region to another, your first step is to create a restore point collection in the target or destination region. To do this, reference the restore point collection from the source region as detailed in Create a VM restore point collection.

Azure PowerShell

```
New-AzRestorePointCollection `
    -ResourceGroupName 'myResourceGroup' `
    -Name 'myRPCollection' `
    -Location 'WestUS' `
    -RestorePointCollectionId '/subscriptions/<SUBSCRIPTION
ID>/resourceGroups/<RG>/providers/Microsoft.Compute/restorePointCollections/
<SOURCE RESTORE POINT COLLECTION>'
```

Step 2: Create the destination VM restore point

After the restore point collection is created, trigger the creation of a restore point in the target restore point collection. Ensure that you've referenced the restore point in the source region that you want to copy and specified the source restore point's identifier in the request body. The source VM's location is inferred from the target restore point collection in which the restore point is being created. See the Restore Points - Create API documentation to create a `RestorePoint`.

Azure PowerShell

```
New-AzRestorePoint `
    -ResourceGroupName 'myResourceGroup' `
```

```
    -RestorePointCollectionName 'myRPCollection'
    -Name 'myRestorePoint'
```

Step 3: Track copy status

To track the status of the copy operation, follow the guidance in the Get restore point copy or replication status section below. This is only applicable for scenarios where the restore points are copied to a different region than the source VM.

Azure PowerShell

```
Get-AzRestorePoint `
    -ResourceGroupName 'myResourceGroup' `
    -RestorePointCollectionName 'myRPCollection'
    -Name 'myRestorePoint'
```

Get restore point copy or replication status

Copying the first VM restore point to another region is a long running operation. The VM restore point can be used to restore a VM only after the operation is completed for all disk restore points. To track the operation's status, call the Restore Point - Get API on the target VM restore point and include the `instanceView` parameter. The return will include the percentage of data that has been copied at the time of the request.

During restore point creation, the `ProvisioningState` will appear as `Creating` in the response. If creation fails, `ProvisioningState` is set to `Failed`.

Create a disk using disk restore points

You can use the VM restore points APIs to restore a VM disk, which can then be used to create a new VM. Use the following steps:

Step 1: Retrieve disk restore point identifiers

Call the Restore Point Collections - Get API on the restore point collection to get access to associated restore points and their IDs. Each VM restore point will in turn contain individual disk restore point identifiers.

Step 2: Create a disk

After you have the list of disk restore point IDs, you can use the Disks - Create Or Update API to create a disk from the disk restore points. You can choose a zone while creating the disk. The zone can be different from zone in which the disk restore point exists.

Restore a VM with a restore point

To restore a full VM from a VM restore point, you must restore individual disks from each disk restore point. This process is described in the Create a disk section. After you restore all the disks, create a new VM and attach the restored disks to the new VM. You can also use the ARM template to restore a full VM along with all the disks.

Get a shared access signature for a disk

To create a Shared Access Signature (SAS) for a disk within a VM restore point, pass the ID of the disk restore points via the `BeginGetAccess` API. If no active SAS exists on the restore point snapshot, a new SAS is created. The new SAS URL is returned in the response. If an active SAS already exists, the SAS duration is extended, and the pre-existing SAS URL is returned in the response.

For more information about granting access to snapshots, see the Grant Access API documentation.

Next steps

Learn more about Backup and restore options for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machine restore points using Azure CLI

Article • 08/22/2024

You can protect your data and guard against extended downtime by creating VM restore points at regular intervals. You can create VM restore points, and exclude disks while creating the restore point, using Azure CLI. Azure CLI is used to create and manage Azure resources using command line or scripts. Alternatively, you can create VM restore points using the Azure portal or using PowerShell.

The az restore-point module is used to create and manage restore points from the command line or in scripts.

In this tutorial, you learn how to:

- ✓ Create a VM restore point collection
- ✓ Create a VM restore point
- ✓ Track the progress of Copy operation
- ✓ Restore a VM

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

- Learn more about the support requirements and limitations before creating a restore point.

Step 1: Create a VM restore point collection

Use the az restore-point collection create command to create a VM restore point collection, as shown below:

```
az restore-point collection create --location "norwayeast" --source-id
"/subscriptions/{subscription-
id}/resourceGroups/ExampleRg/providers/Microsoft.Compute/virtualMachines/Exa
mpleVM" --tags myTag1="tagValue1" --resource-group "ExampleRg" --collection-
name "ExampleRpc"
```

Step 2: Create a VM restore point

Create a VM restore point with the az restore-point create command as follows:

```
az restore-point create --resource-group "ExampleRg" --collection-name
"ExampleRpc" --name "ExampleRp"
```

To create a crash consistent restore point set the optional parameter "consistency-mode" to "CrashConsistent". This feature is currently in preview.

Exclude disks when creating a restore point

Exclude the disks that you do not want to be a part of the restore point with the `--exclude-disks` parameter, as follows:

```
az restore-point create --exclude-disks "/subscriptions/{subscription-
id}/resourceGroups/ExampleRg/providers/Microsoft.Compute/disks/ExampleDisk1"
--resource-group "ExampleRg" --collection-name "ExampleRpc" --name
"ExampleRp"
```

Step 3: Track the status of the VM restore point creation

Use the az restore-point show command to track the progress of the VM restore point creation.

```
az restore-point show --resource-group "ExampleRg" --collection-name
"ExampleRpc" --name "ExampleRp"
```

Restore a VM from VM restore point

To restore a VM from a VM restore point, first restore individual disks from each disk restore point. You can also use the ARM template to restore a full VM along with all the disks.

```
# Create Disks from disk restore points
$osDiskRestorePoint = az restore-point show --resource-group "ExampleRg" --
collection-name "ExampleRpc" --name "ExampleRp" --query
"sourceMetadata.storageProfile.dataDisks[0].diskRestorePoint.id"
$dataDisk1RestorePoint = az restore-point show --resource-group "ExampleRg"
--collection-name "ExampleRpcTarget" --name "ExampleRpTarget" –query
"sourceMetadata.storageProfile.dataDisks[0].diskRestorePoint.id"
$dataDisk2RestorePoint = az restore-point show --resource-group "ExampleRg"
--collection-name "ExampleRpcTarget" --name "ExampleRpTarget" –query
"sourceMetadata.storageProfile.dataDisks[0].diskRestorePoint.id"

az disk create --resource-group “ExampleRg” --name “ExampleOSDisk” --sku
Premium_LRS --size-gb 128 --source $osDiskRestorePoint

az disk create --resource-group “ExampleRg” --name “ExampleDataDisk1” --sku
Premium_LRS --size-gb 128 --source $dataDisk1RestorePoint

az disk create --resource-group “ExampleRg” --name “ExampleDataDisk1” --sku
Premium_LRS --size-gb 128 --source $dataDisk2RestorePoint
```

Once you have created the disks, create a new VM and attach these restored disks to the newly created VM.

Next steps

Learn more about Backup and restore options for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machine restore points using PowerShell

Article • 08/22/2024

> **Note**
>
> We recommend that you use the Azure Az PowerShell module to interact with Azure. To get started, see Install Azure PowerShell. To learn how to migrate to the Az PowerShell module, see Migrate Azure PowerShell from AzureRM to Az.

You can create Virtual Machine restore points using PowerShell scripts. The Azure PowerShell Az module is used to create and manage Azure resources from the command line or in scripts.

You can protect your data and guard against extended downtime by creating VM restore points at regular intervals. This article shows you how to create VM restore points, and exclude disks from the restore point, using the Az.Compute module. Alternatively, you can create VM restore points using the Azure CLI or in the Azure portal.

In this tutorial, you learn how to:

- ✓ Create a VM restore point collection
- ✓ Create a VM restore point
- ✓ Track the progress of Copy operation
- ✓ Restore a VM

Prerequisites

- Learn more about the support requirements and limitations before creating a restore point.

Step 1: Create a VM restore point collection

Use the New-AzRestorePointCollection cmdlet to create a VM restore point collection.

```
New-AzRestorePointCollection -ResourceGroupName ExampleRG -Name ExampleRPC -
VmId “/subscriptions/{SubscriptionId}/resourcegroups/
```

```
ExampleRG/providers/microsoft.compute/virtualmachines/Example-vm-1” -
Location “WestEurope”
```

Step 2: Create a VM restore point

Create a VM restore point with the New-AzRestorePoint cmdlet as shown below:

```
New-AzRestorePoint -ResourceGroupName ExampleRG -RestorePointCollectionName
ExampleRPC -Name ExampleRP
```

To create a crash consistent restore point set the optional parameter "ConsistencyMode" to "CrashConsistent". This feature is currently in preview.

Exclude disks from the restore point

Exclude certain disks that you do not want to be a part of the restore point with the `-DisksToExclude` parameter, as follows:

```
New-AzRestorePoint -ResourceGroupName ExampleRG -RestorePointCollectionName
ExampleRPC -Name ExampleRP -DisksToExclude
“/subscriptions/{SubscriptionId}/resourcegroups/
ExampleRG/providers/Microsoft.Compute/disks/example-vm-1-data_disk_1”
```

Step 3: Track the status of the VM restore point creation

You can track the progress of the VM restore point creation using the Get-AzRestorePoint cmdlet, as follows:

```
Get-AzRestorePoint -ResourceGroupName ExampleRG -RestorePointCollectionName
ExampleRPC -Name ExampleRP
```

Restore a VM from VM restore point

To restore a VM from a VM restore point, first restore individual disks from each disk restore point. You can also use the ARM template to restore a full VM along with all the disks.

```
# Create Disks from disk restore points
$restorePoint = Get-AzRestorePoint -ResourceGroupName ExampleRG -
RestorePointCollectionName ExampleRPC -Name ExampleRP

$osDiskRestorePoint =
$restorePoint.SourceMetadata.StorageProfile.OsDisk.DiskRestorePoint.Id
$dataDisk1RestorePoint =
$restorePoint.sourceMetadata.storageProfile.dataDisks[0].diskRestorePoint.id
$dataDisk2RestorePoint =
$restorePoint.sourceMetadata.storageProfile.dataDisks[1].diskRestorePoint.id

New-AzDisk -DiskName “ExampleOSDisk” (New-AzDiskConfig  -Location eastus -
CreateOption Restore -SourceResourceId $osDiskRestorePoint) -
ResourceGroupName ExampleRg

New-AzDisk -DiskName “ExampleDataDisk1” (New-AzDiskConfig  -Location eastus
-CreateOption Restore -SourceResourceId $dataDisk1RestorePoint) -
ResourceGroupName ExampleRg

New-AzDisk -DiskName “ExampleDataDisk2” (New-AzDiskConfig  -Location eastus
-CreateOption Restore -SourceResourceId $dataDisk2RestorePoint) -
ResourceGroupName ExampleRg
```

After you create the disks, create a new VM and attach these restored disks to the newly created VM.

Next steps

Learn more about Backup and restore options for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Create virtual machine restore points using Azure portal

Article • 08/22/2024

You can create virtual machine restore points through the Azure portal. You can protect your data and guard against extended downtime by creating VM restore points at regular intervals. This article shows you how to create VM restore points using the Azure portal. Alternatively, you can create VM restore points using the Azure CLI or using PowerShell.

In this tutorial, you learn how to:

- ✓ Create a VM restore point collection
- ✓ Create a VM restore point
- ✓ Track the progress of Copy operation
- ✓ Restore a VM

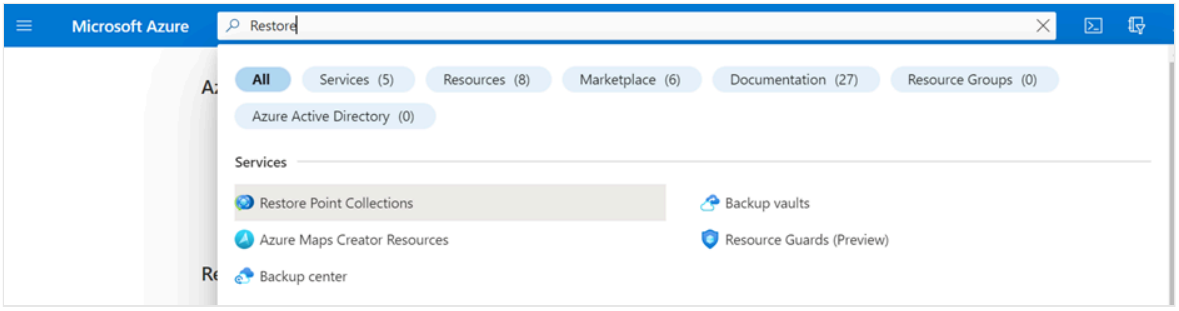
Prerequisites

- Learn more about the support requirements and limitations before creating a restore point.

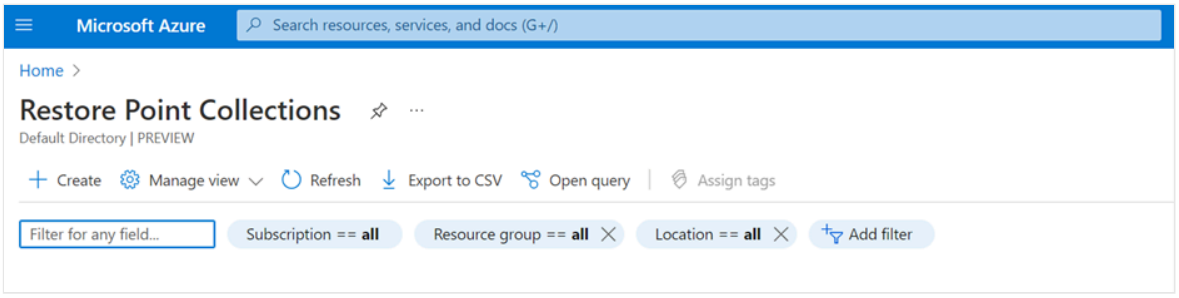
Step 1: Create a VM restore point collection

Use the following steps to create a VM restore points collection:

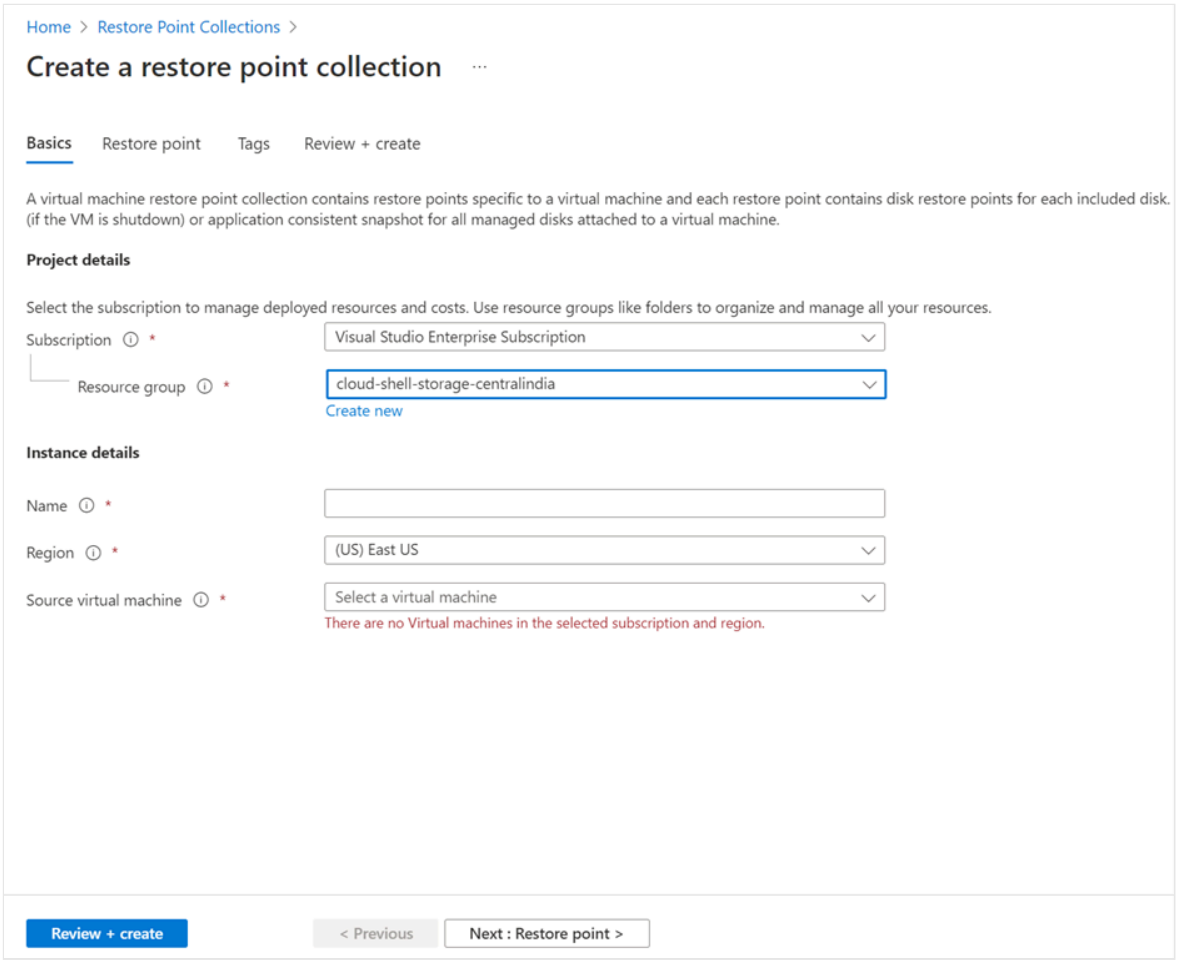
1. Sign in to the Azure portal.
2. In the Search box, enter **Restore Point Collections**.



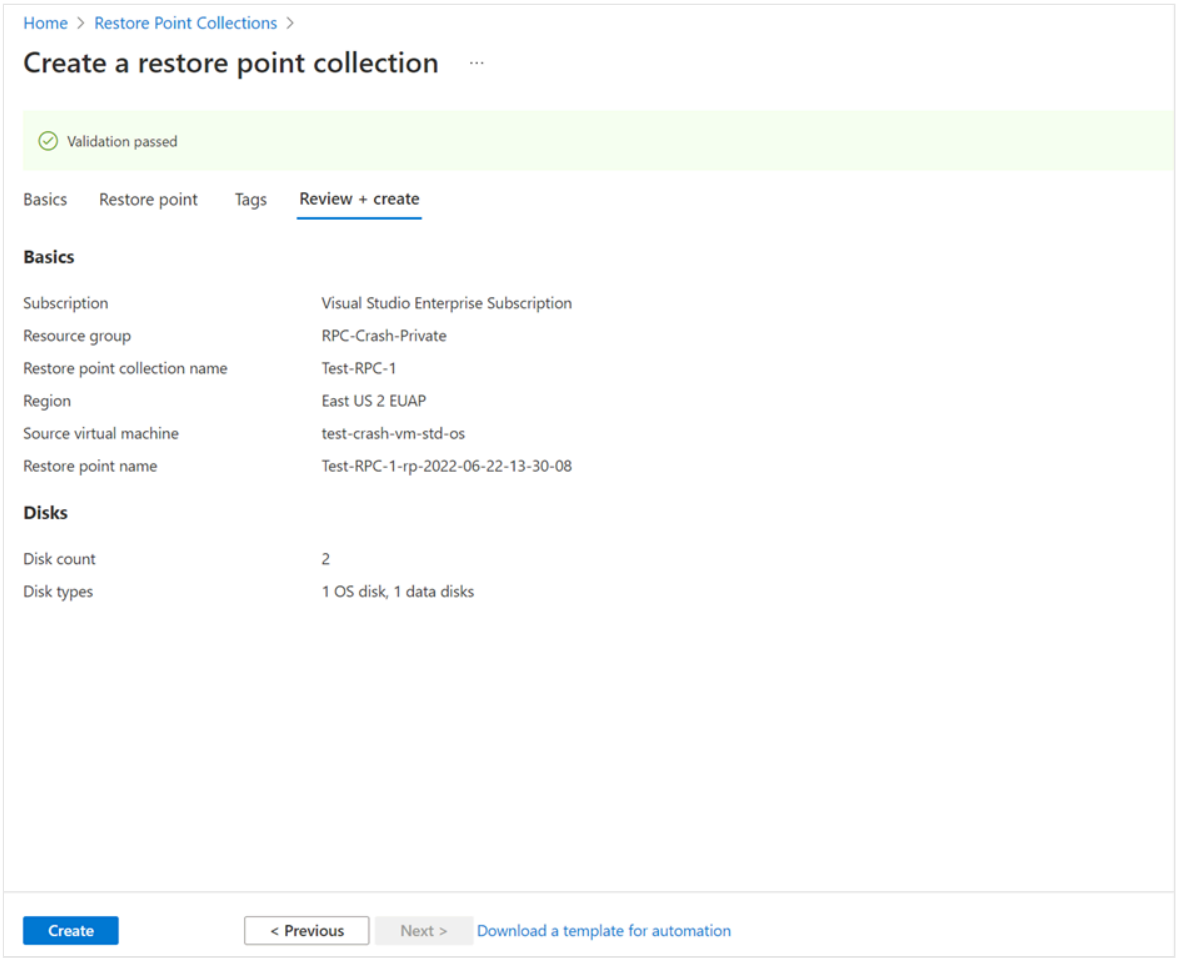
3. Select + **Create** to create a new Restore Point Collection.



4. Enter the details and select the VM for which you want to create a restore point collection.



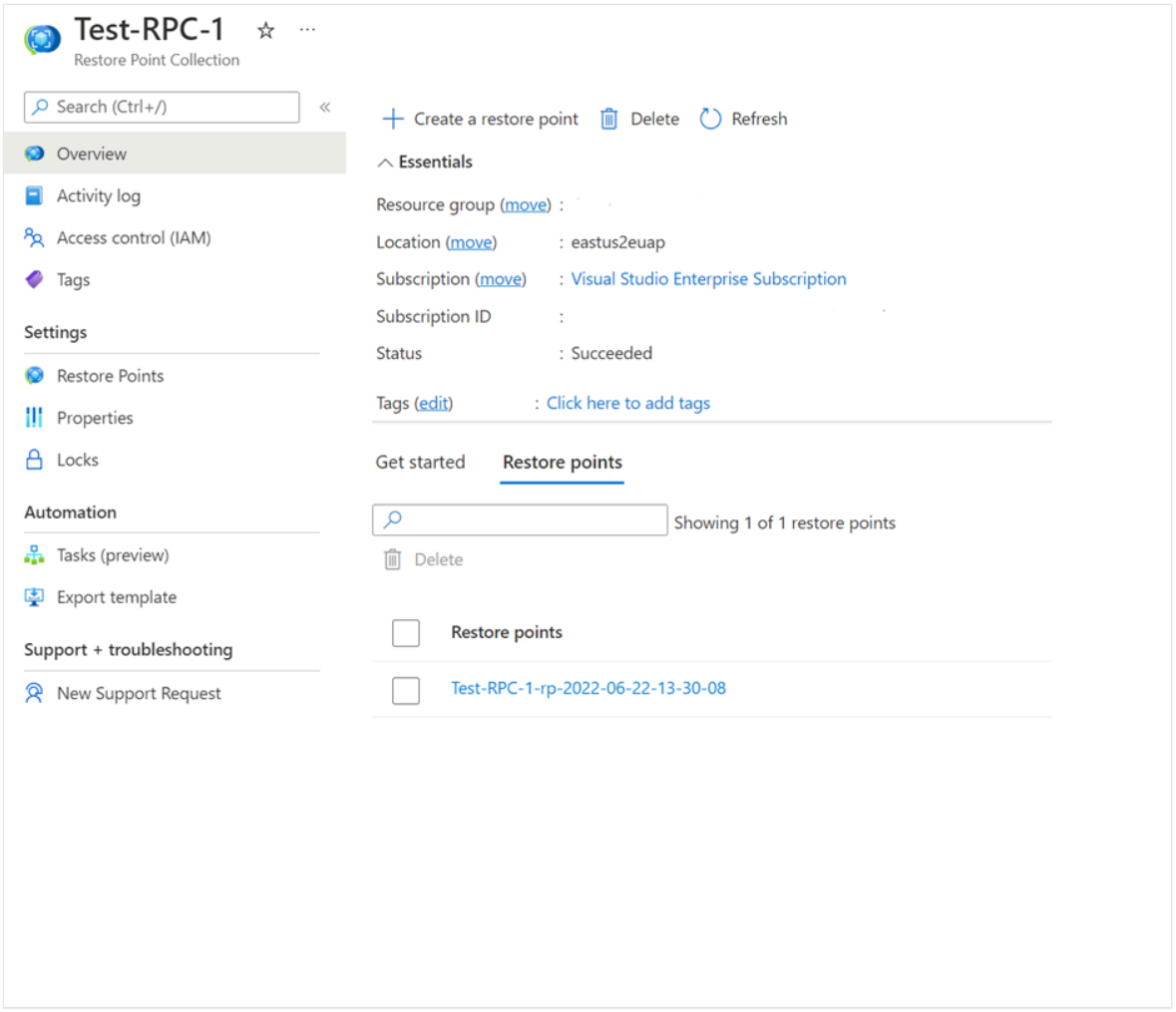
5. Select **Next: Restore Point** to create your first restore point or select **Review + Create** to create an empty restore point collection.



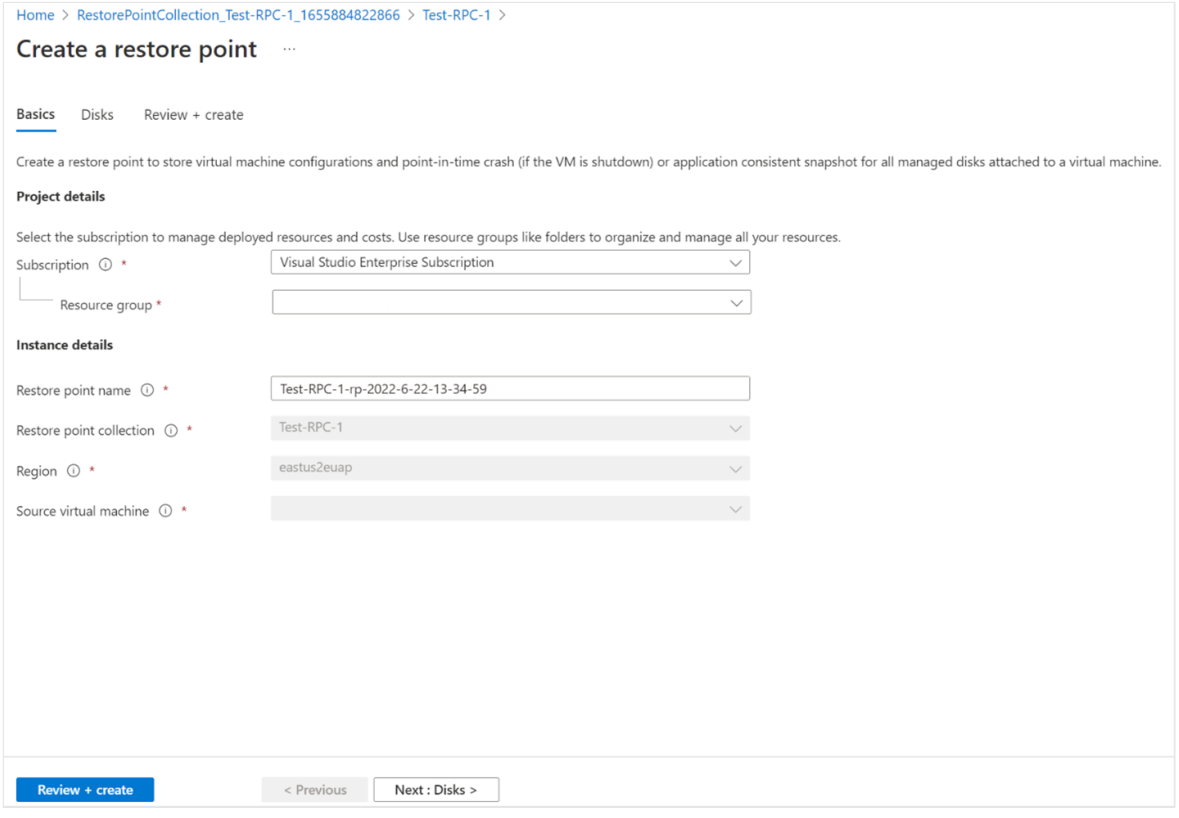
Step 2: Create a VM restore point

Use the following steps to create a VM restore point:

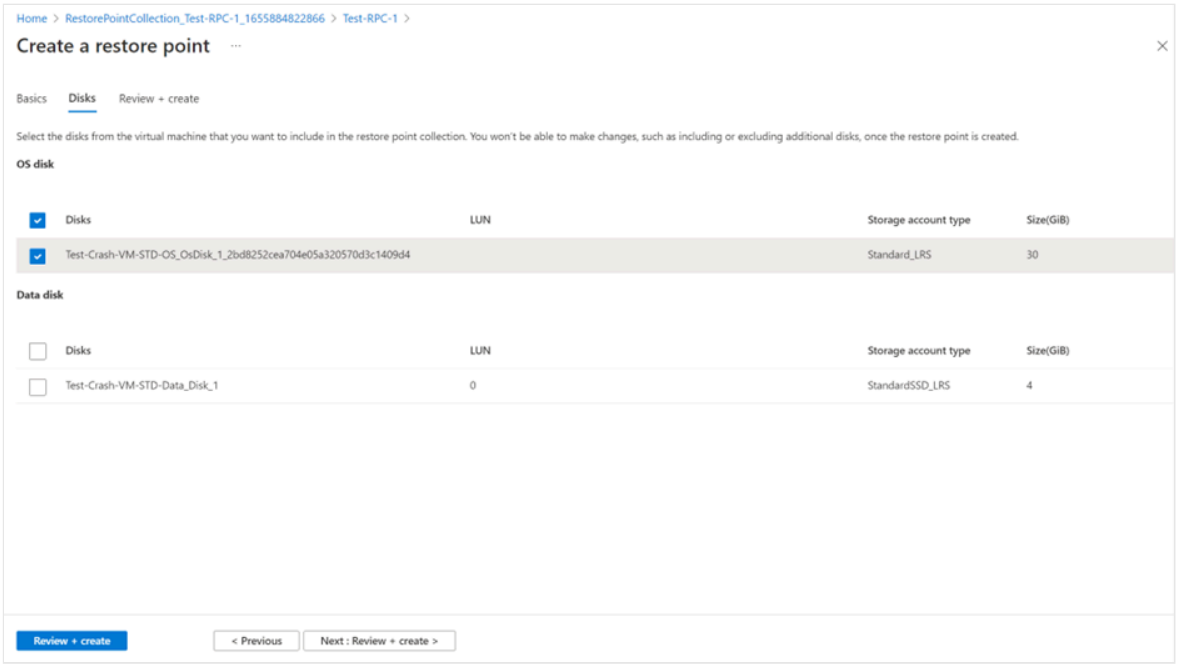
1. Navigate to the restore point collection where you want to create restore points and select + **Create a restore point** to create new restore point for the VM.



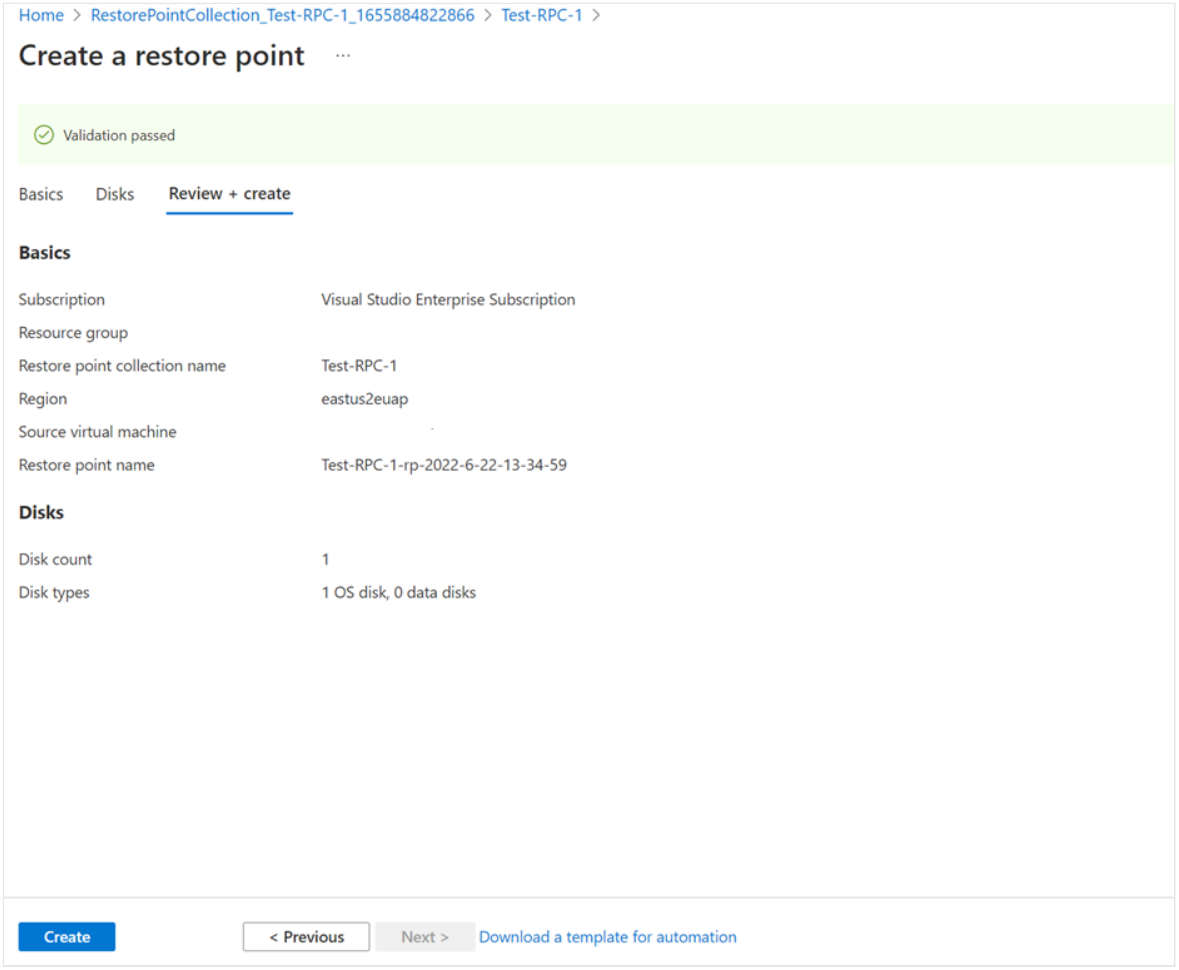
2. Enter a name for the restore point and other required details and select **Next: Disks >**.



3. Select the disks to be included in the restore point.

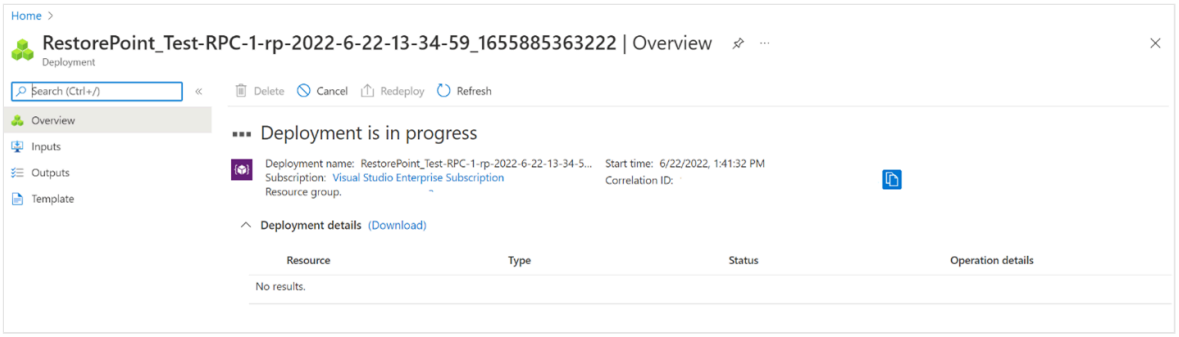


4. Select **Review + create** to validate the settings. Once validation is completed, select **Create** to create the restore point.



Step 3: Track the status of the VM restore point creation

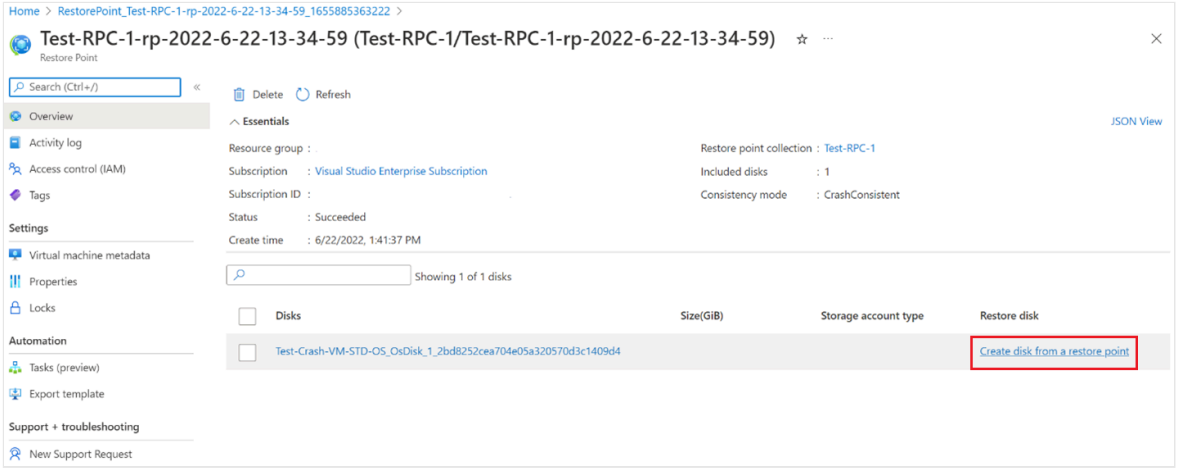
1. Select the notification to track the progress of the restore point creation.



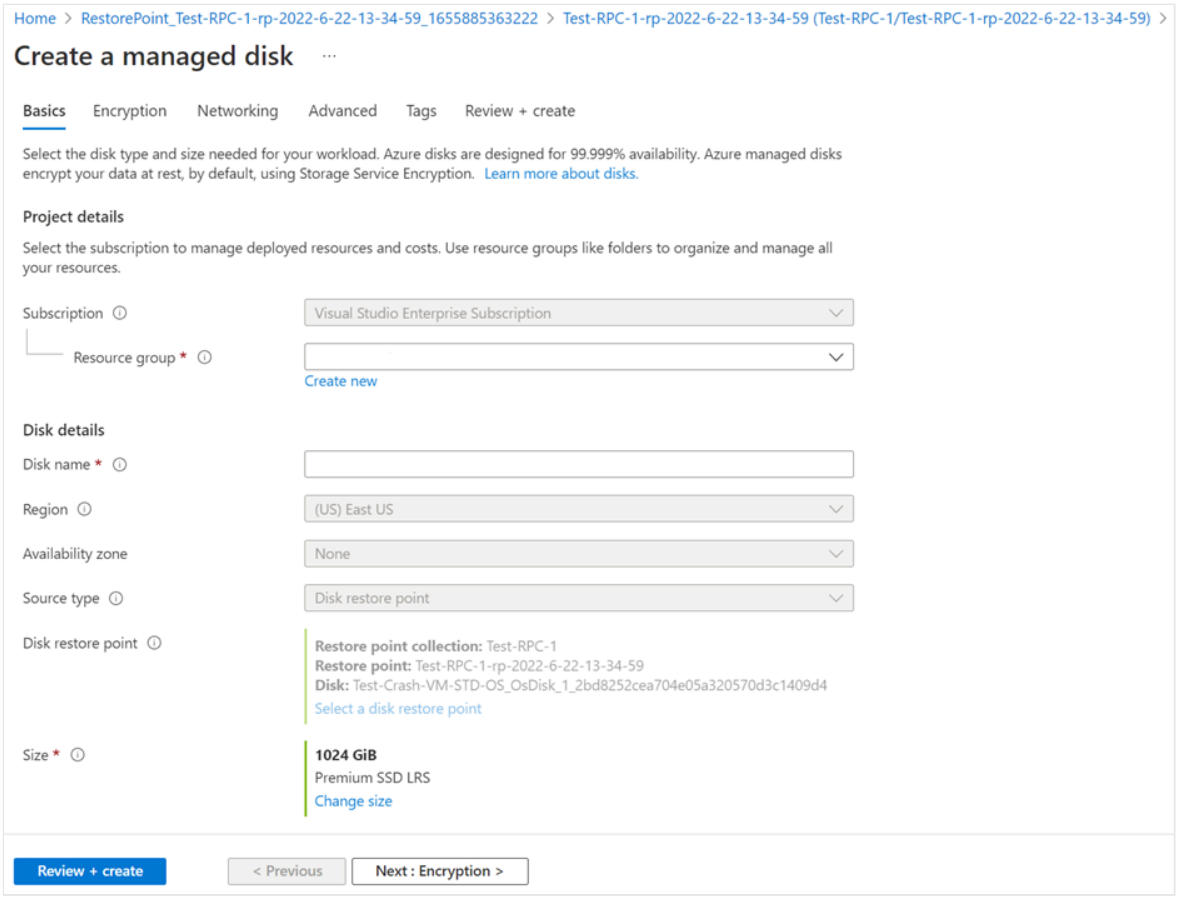
Restore a VM from a restore point

To restore a VM from a VM restore point, first restore individual disks from each disk restore point. You can also use the ARM template to restore a VM along with all the disks.

1. Select **Create a disk from a restore point** to restore a disk from a disk restore point. Do this for all the disks that you want to restore.



2. Enter the details in the **Create a managed disk** dialog to create disks from the restore points. Once the disks are created, create a new VM and attach these restored disks to the newly created VM.



Next steps

Learn more about Backup and restore options for virtual machines in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Cross-region copy of VM restore points

Article • 08/22/2024

In this article, you learn how to copy virtual machine (VM) restore points to another region.

Prerequisites

To copy a restore point across a region, you need to precreate a `restorePointCollection` resource in the target region.

Learn more about cross-region copy and its limitation before you copy restore points.

Create a restore point collection in a target region

The first step in copying an existing VM restore point from one region to another is to create a `restorePointCollection` resource in the target region by referencing `restorePointCollection` from the source region.

URI request

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/restorePointCollections/{rest
orePointCollectionName}&api-version={api-version}
```

Request body

```
{
    "name": "name of target restorePointCollection resource",
    "location": "location of target restorePointCollection resource",
    "tags": {
        "department": "finance"
    },
    "properties": {
         "source": {
               "id":
"/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microso
```

```
ft.compute/restorePointCollections/{restorePointCollectionName}"
          }
    }
}
```

Response

The request response includes a status code and a set of response headers.

Status code

The operation returns 201 during creation and 200 during the update.

Response body

```
{
    "name": "name of the copied restorePointCollection resource",
    "id": "CSM Id of copied restorePointCollection resource",
    "type": "Microsoft.Compute/restorePointCollections",
    "location": "location of the copied restorePointCollection resource",
    "tags": {
        "department": "finance"
    },
    "properties": {
        "source": {
            "id":
"/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microso
ft.compute/restorePointCollections/{restorePointCollectionName}",
            "location": "location of source RPC"
        }
    }
}
```

Create a VM restore point in a target region

The next step is to trigger the copy of a restore point in the target `RestorePointCollection` resource by referencing the restore point in the source region that needs to be copied.

URI request

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/restorePointCollections/{rest
orePointCollectionName}/restorePoints/{restorePointName}&api-version={api-
version}
```

Request body

```
{
    "name": "name of the restore point resource",
    "properties": {
        "sourceRestorePoint": {
            "id":
"/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microso
ft.compute/restorePointCollections/{restorePointCollectionName}/restorePoint
s/{restorePointName}"
        }
    }
}
```

> ⓘ **Note**
>
> The location of `sourceRestorePoint` is inferred from that of the source `RestorePointCollection`.

Response

The request response includes a status code and a set of response headers.

Status code

This operation is long running, so the operation returns 201 during creation. The client is expected to poll for the status by using the operation. Both the `location` and `Azure-AsyncOperation` headers are provided for this purpose.

During restore point creation, `ProvisioningState` appears as `Creating` in the GET restore point API response. If creation fails, `ProvisioningState` appears as `Failed`. `ProvisioningState` is set to `Succeeded` when the data copy across regions is initiated.

> **Note**
>
> You can track the copy status by calling GET instance view (`?$expand=instanceView`) on the target VM restore point. For steps on how to do this, see the section "Get the VM restore points Copy/Replication status." The VM restore point is considered usable (can be used to restore a VM) only when a copy of all the disk restore points are successful.

Response body

```
{
    "id": "CSM Id of the restore point",
    "name": "name of the restore point",
    "properties": {
        "optionalProperties": "opaque bag of properties to be passed to
extension",
        "sourceRestorePoint": {
            "id":
"/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microso
ft.compute/restorePointCollections/{restorePointCollectionName}/restorePoint
s/{restorePointName}"
        },
        "consistencyMode": "CrashConsistent | FileSystemConsistent |
ApplicationConsistent",
        "sourceMetadata": {
            "vmId": "Unique Guid of the VM from which the restore point was
created",
            "location": "source VM location",
            "hardwareProfile": {
                "vmSize": "Standard_A1"
            },
            "osProfile": {
                "computername": "",
                "adminUsername": "",
                "secrets": [
                    {
                        "sourceVault": {
                            "id":
"/subscriptions/<subId>/resourceGroups/<rgName>/providers/Microsoft.KeyVault
/vaults/<keyvault-name>"
                        },
                        "vaultCertificates": [
                            {
                                "certificateUrl": "https://<keyvault-
name>.vault.azure.net/secrets/<secret-name>/<secret-version>",
                                "certificateStore": "certificateStoreName on
Windows"
```

```
                        }
                    ]
                }
            ],
            "customData": "",
            "windowsConfiguration": {
                "provisionVMAgent": "true|false",
                "winRM": {
                    "listeners": [
                        {
                            "protocol": "http"
                        },
                        {
                            "protocol": "https",
                            "certificateUrl": ""
                        }
                    ]
                },
                "additionalUnattendContent": [
                    {
                        "pass": "oobesystem",
                        "component": "Microsoft-Windows-Shell-Setup",
                        "settingName": "FirstLogonCommands|AutoLogon",
                        "content": "<XML unattend content>"
                    }
                ],
                "enableAutomaticUpdates": "true|false"
            },
            "linuxConfiguration": {
                "disablePasswordAuthentication": "true|false",
                "ssh": {
                    "publicKeys": [
                        {
                            "path": "Path-Where-To-Place-Public-Key-On-
VM",
                            "keyData": "PEM-Encoded-public-key-file"
                        }
                    ]
                }
            }
        },
        "storageProfile": {
            "osDisk": {
                "osType": "Windows|Linux",
                "name": "OSDiskName",
                "diskSizeGB": "10",
                "caching": "ReadWrite",
                "managedDisk": {
                    "id": "CSM Id of the managed disk",
                    "storageAccountType": "Standard_LRS"
                },
                "diskRestorePoint": {
                    "id":
"/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpc
Name>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>"
```

```
                    }
                },
                "dataDisks": [
                    {
                        "lun": "0",
                        "name": "datadisk0",
                        "diskSizeGB": "10",
                        "caching": "ReadWrite",
                        "managedDisk": {
                            "id": "CSM Id of the managed disk",
                            "storageAccountType": "Standard_LRS"
                        },
                        "diskRestorePoint": {
                            "id":
"/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpc
Name>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>"
                        }
                    }
                ]
            },
            "diagnosticsProfile": {
                "bootDiagnostics": {
                    "enabled": true,
                    "storageUri": "
http://storageaccount.blob.core.windows.net/"
                }
            }
        },
        "provisioningState": "Succeeded | Failed | Creating | Deleting",
        "provisioningDetails": {
            "creationTime": "Creation Time of Restore point in UTC"
        }
    }
}
```

Get the VM restore points Copy/Replication status

After the copy of VM restore points is initiated, you can track the copy status by calling the GET instance view (`?$expand=instanceView`) on the target VM restore point.

URI request

```
GET
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{
resourceGroupName}/providers/Microsoft.Compute/restorePointCollections/{rest
orePointCollectionName}/restorePoints/{restorePointName}?
$expand=instanceView&api-version={api-version}
```

Response

```
{
    "id": "CSM Id of the restore point",
    "name": "name of the restore point",
    "properties": {
        "optionalProperties": "opaque bag of properties to be passed to
extension",
        "sourceRestorePoint": {
            "id":
"/subscriptions/{subid}/resourceGroups/{resourceGroupName}/providers/microso
ft.compute/restorePointCollections/{restorePointCollectionName}/restorePoint
s/{restorePointName}"
        },
        "consistencyMode": "CrashConsistent | FileSystemConsistent |
ApplicationConsistent",
        "sourceMetadata": {
            "vmId": "Unique Guid of the VM from which the restore point was
created",
            "location": "source VM location",
            "hardwareProfile": {
                "vmSize": "Standard_A1"
            },
            "osProfile": {
                "computername": "",
                "adminUsername": "",
                "secrets": [
                    {
                        "sourceVault": {
                            "id":
"/subscriptions/<subId>/resourceGroups/<rgName>/providers/Microsoft.KeyVault
/vaults/<keyvault-name>"
                        },
                        "vaultCertificates": [
                            {
                                "certificateUrl": "https://<keyvault-
name>.vault.azure.net/secrets/<secret-name>/<secret-version>",
                                "certificateStore": "certificateStoreName on
Windows"
                            }
                        ]
                    }
                ],
                "customData": "",
                "windowsConfiguration": {
                    "provisionVMAgent": "true|false",
                    "winRM": {
                        "listeners": [
                            {
                                "protocol": "http"
                            },
                            {
```

```
                                "protocol": "https",
                                "certificateUrl": ""
                            }
                        ]
                    },
                    "additionalUnattendContent": [
                        {
                            "pass": "oobesystem",
                            "component": "Microsoft-Windows-Shell-Setup",
                            "settingName": "FirstLogonCommands|AutoLogon",
                            "content": "<XML unattend content>"
                        }
                    ],
                    "enableAutomaticUpdates": "true|false"
                },
                "linuxConfiguration": {
                    "disablePasswordAuthentication": "true|false",
                    "ssh": {
                        "publicKeys": [
                            {
                                "path": "Path-Where-To-Place-Public-Key-On-
VM",
                                "keyData": "PEM-Encoded-public-key-file"
                            }
                        ]
                    }
                }
            },
            "storageProfile": {
                "osDisk": {
                    "osType": "Windows|Linux",
                    "name": "OSDiskName",
                    "diskSizeGB": "10",
                    "caching": "ReadWrite",
                    "managedDisk": {
                        "id": "CSM Id of the managed disk",
                        "storageAccountType": "Standard_LRS"
                    },
                    "diskRestorePoint": {
                        "id":
"/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpc
Name>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>"
                    }
                },
                "dataDisks": [
                    {
                        "lun": "0",
                        "name": "datadisk0",
                        "diskSizeGB": "10",
                        "caching": "ReadWrite",
                        "managedDisk": {
                            "id": "CSM Id of the managed disk",
                            "storageAccountType": "Standard_LRS"
                        },
                        "diskRestorePoint": {
```

```
                            "id":
"/subscriptions/<subId>/resourceGroups/<rgName>/restorePointCollections/<rpc
Name>/restorePoints/<rpName>/diskRestorePoints/<diskRestorePointName>"
                        }
                    }
                ]
            },
            "diagnosticsProfile": {
                "bootDiagnostics": {
                    "enabled": true,
                    "storageUri": "
http://storageaccount.blob.core.windows.net/"
                }
            }
        },
        "provisioningState": "Succeeded | Failed | Creating | Deleting",
        "provisioningDetails": {
            "creationTime": "Creation Time of Restore point in UTC"
        },
        "instanceView": {
            "statuses": [
                {
                    "code": "ReplicationState/succeeded",
                    "level": "Info",
                    "displayStatus": "Replication succeeded"
                }
            ],
            "diskRestorePoints": [
                {
                    "id": "<diskRestorePoint Arm Id>",
                    "replicationStatus": {
                        "status": {
                            "code": "ReplicationState/succeeded",
                            "level": "Info",
                            "displayStatus": "Replication succeeded"
                        },
                        "completionPercent": "<completion percentage of the
replication>"
                    }
                }
            ]
        }
    }
}
```

Next steps

- Create a VM restore point.
- Learn more about backup and restore options for VMs in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Support matrix for VM restore points

Summarize this article for me

This article summarizes the support matrix and limitations of using VM restore points.

VM restore points support matrix

The following table summarizes the support matrix for VM restore points.

Expand table

Scenarios	Supported by VM restore points
VMs using Managed disks	Yes
VMs using unmanaged disks	No
VM sizes	**Crash consistency** is supported for VM SKUs that support Premium storage. Crash-consistent restore points created for Intel V6+ (Dsv6-series, Edsv6-series, Esv6-series, etc.) and AMD V7+ (Dasv7-series, Dadsv7-series, Easv7-series, Faldsv7-series, etc.) Virtual Machines (VMs) with **more than one data disk might not be consistent across disks**. **Application consistency** is supported for all VM SKUs.
VMs using Ultra Disks	Yes. Supported for application consistency. Not supported for crash consistency. Exclude these disks and create a VM restore point when using crash consistency.
VMs using Premium SSD v2 disks	Yes. Supported for application consistency. Not supported for crash consistency. Exclude these disks and create a VM restore point when using crash consistency.
VMs using Ephemeral OS Disks	No. Exclude these disks and create a VM restore point.
VMs using shared disks	No. Exclude these disks and create a VM restore point.
VMs with extensions	Yes
VMs with trusted launch OS disk	Yes
Data disks restored from Trusted launch enabled OS Disk	No
Confidential VMs	Supported for application consistency only with size v6-series in UAE North and Korea Central (**preview**) v5-series isn't supported. Not supported for crash consistency.

Scenarios	Supported by VM restore points
Generation 2 VMs (UEFI boot)	Yes
VMs with NVMe disks (Storage optimized - Lsv2-series)	Yes for **Application consistent**. **Crash-consistent** restore points created for Intel V6+ (Dsv6-series, Edsv6-series, Esv6-series, etc.) and AMD V7+ (Dasv7-series, Dadsv7-series, Easv7-series, Faldsv7-series, etc.) Virtual Machines (VMs) with more than one data disk might not be consistent across disks.
VMs in Proximity placement groups	Yes
VMs in an availability set	Yes. You can create VM restore points for individual VMs within an availability set. You need to create restore points for all the VMs within an availability set to protect an entire availability set instance.
VMs inside VMSS with uniform orchestration	No
VMs inside VMSS with flexible orchestration	Yes. You can create VM restore points for individual VMs within the virtual machine scale set flex. However, you need to create restore points for all the VMs within the virtual machine scale set flex to protect an entire virtual machine scale set flex instance.
Spot VMs (Low priority VMs)	Yes
VMs with dedicated hosts	Yes
VMs with Host caching enabled	Yes
VMs created from marketplace images	Yes
VMs created from custom images	Yes
VM with HUB (Hybrid Use Benefit) license	Yes
VMs migrated from on-prem using Azure Migrate	Yes
VMs with RBAC policies	Yes
Temporary disk in VMs	Yes. You can create VM restore point for VMs with temporary disks. However, the restore points created don't contain the data from the temporary disks.
VMs with Standard HDDs	Yes
VMs with Standard SSDs	Yes

Scenarios	Supported by VM restore points
VMs with Premium SSDs	Yes
VMs with ZRS disks	Yes
VMs with server-side encryption using service-managed keys	Yes. The encryption of source disk won't be enabled on the restore point.
VMs with server-side encryption using customer-managed keys	Yes. The encryption of source disk won't be enabled on the restore point.
VMs with double encryption at rest	Yes. The encryption of source disk won't be enabled on the restore point.
VMs with Host based encryption enabled with PMK/CMK/Double encryption	Yes. The encryption of source disk won't be enabled on the restore point.
VMs with ADE (Azure Disk Encryption)	Yes. The encryption of source disk won't be enabled on the restore point.
VMs using Accelerated Networking	Yes
Azure Boost compatible Virtual machine sizes	Yes for **Application consistent**. **Crash-consistent** restore points created for Intel V6+ (Dsv6-series, Edsv6-series, Esv6-series, etc.) and AMD V7+ (Dasv7-series, Dadsv7-series, Easv7-series, Faldsv7-series, etc.) Virtual Machines (VMs) with more than one data disk may not be consistent across disks.
Minimum Frequency at which App consistent restore point can be taken	3 hours
Minimum Frequency at which crash consistent restore points can be taken	1 hour
API version for Application consistent restore point	2021-03-01 or later
API version for Crash consistent restore point	2021-07-01 or later

> ⓘ **Note**

Restore Points (App consistent or crash consistent) can be created by customer at the minimum supported frequency as mentioned above. Taking restore points at a frequency lower than supported would result in failure.

Operating system support for application consistency

Windows

The following Windows operating systems are supported when creating restore points for Azure VMs running on Windows.

- Windows 10 Client (64 bit only)
- Windows Server 2022 (Datacenter/Datacenter Core/Standard)
- Windows Server 2019 (Datacenter/Datacenter Core/Standard)
- Windows Server 2016 (Datacenter/Datacenter Core/Standard)
- Windows Server 2012 R2 (Datacenter/Standard)
- Windows Server 2012 (Datacenter/Standard)
- OS that have reached extended security update won't be supported. Check your product's lifecycle here

Restore points don't support 32-bit operating systems.

Linux

For Azure VM Linux VMs, restore points support the list of Linux distributions endorsed by Azure. Note the following:

- Restore points don't support Core OS Linux.
- Restore points don't support 32-bit operating systems.
- Other bring-your-own Linux distributions might work as long as the Azure VM agent for Linux is available on the VM, and as long as Python is supported.
- Restore points don't support a proxy-configured Linux VM if it doesn't have Python version 2.7 or higher installed.
- Restore points don't back up NFS files that are mounted from storage, or from any other NFS server, to Linux or Windows machines. It only backs up disks that are locally attached to the VM.

Operating system support for crash consistency

- All Operating systems are supported.

Other limitations

- Restore points are supported only for managed disks.
- Ephemeral OS disks, and Shared disks aren't supported via both consistency modes.
- Restore points APIs require an API of version 2021-03-01 or later for application consistency.
- Restore points APIs require an API of version 2021-03-01 or later for crash consistency.
- A maximum of 10,000 restore point collections can be retained at per subscription per region level.
- A maximum of 500 VM restore points can be retained at any time for a VM, irrespective of the number of restore point collections.
- Concurrent creation of restore points for a VM isn't supported.
- Movement of virtual machines between resource groups or subscriptions is supported when VM has restore points. New restore point creation will fail on the previous VM as the VM no longer exists after the movement. You need to clean up the restore point collection and restore points of the old VM if no longer needed.

Confidential Virtual Machines (Preview)

Expand table

Scenarios	Supported by VM restore points
Consistency type	Application consistent restore points
Regions supported	UAE North and Korea Central
VM Sizes	v6-series is supported. v5-series isn't supported.
Key rotation	Supported. When key rotation occurs on a confidential virtual machine, the keys for the VM disks, related restore points, and snapshots update automatically. **Known issue:** The key rotation in this preview release might have performance issues or fail in the following scenarios: - More than 40 disks are attached to one DES when (only) restore points are associated with these disks. If you also directly create disk snapshots outside of Azure backup for these disks connected to the same DES, this lowers the safe threshold of 40 disks to DES mapping. **Recommendation:** Keep the number of disks connected to each DES to a minimum until the issue is resolved.

Scenarios	Supported by VM restore points
Prerequisites	Before you configure backup for CVM, ensure that the following prerequisites are met: Register for the preview feature in your Azure subscription - Name: `RestorePointSupportForConfidentialVMV2` Provider: `Microsoft.Compute`. The registration is autoapproved. You can follow the steps here to do this on the portal. You can also run the following PowerShell cmdlet. `azurepowershell-interactive Register-AzProviderFeature -FeatureName "RestorePointSupportForConfidentialVMV2" -ProviderNamespace "Microsoft.Compute"` Identify or create a Confidential VM (CVM) in a supported region. See the supported regions.

> **Note**
>
> Public preview of cross-region copy of VM restore points is available, with the following limitations:
>
> - Private links aren't supported when copying restore points across regions or creating restore points in a region other than the source VM.
> - Customer-managed key encrypted restore points, when copied to a target region or created directly in the target region are created as platform-managed key encrypted restore points.
> - No portal support for cross region copy and cross region creation of restore points

Next steps

- Learn how to create VM restore points using CLI, Azure portal, and PowerShell.

Last updated on 03/10/2026

Troubleshoot restore point failures: Issues with the agent or extension

Article • 08/22/2024

This article provides troubleshooting steps that can help you resolve restore point errors related to communication with the VM agent and extension.

If your Azure issue is not addressed in this article, visit the Azure forums on Microsoft Q & A and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

Step-by-step guide to troubleshoot restore point failures

Most common restore point failures can be resolved by following the troubleshooting steps listed below:

Step 1: Check the health of Azure VM

- **Ensure Azure VM provisioning state is 'Running'**: If the VM provisioning state is in the **Stopped/Deallocated/Updating** state, it interferes with the restore point operation. In the Azure portal, go to **Virtual Machines** > **Overview** and ensure the VM status is **Running** and retry the restore point operation.
- **Review pending OS updates or reboots**: Ensure there are no pending OS updates or pending reboots on the VM.

Step 2: Check the health of Azure VM Guest Agent service

Ensure Azure VM Guest Agent service is started and up-to-date:

On a Windows VM

- Navigate to **services.msc** and ensure **Windows Azure VM Guest Agent service** is up and running. Also, ensure the latest version is installed. Learn more.
- The Azure VM Agent is installed by default on any Windows VM deployed from an Azure Marketplace image from the portal, PowerShell, Command Line

Interface, or an Azure Resource Manager template. A manual installation of the Agent may be necessary when you create a custom VM image that's deployed to Azure.

- Review the support matrix to check if VM runs on the supported Windows operating system.

Step 3: Check the health of Azure VM Extension

- **Ensure all Azure VM Extensions are in 'provisioning succeeded' state**: If any extension is in a failed state, then it can interfere with the restore point operation.
  - In the Azure portal, go to **Virtual machines** > **Settings** > **Extensions** > **Extensions status** and check if all the extensions are in **provisioning succeeded** state.
  - Ensure all extension issues are resolved and retry the restore point operation.
- **Ensure COM+ System Application** is up and running. Also, the **Distributed Transaction Coordinator service** should be running as **Network Service account**.

Follow the troubleshooting steps in troubleshoot COM+ and MSDTC issues in case of issues.

Step 4: Check the health of Azure VM Snapshot Extension

Restore points use the VM Snapshot Extension to take an application consistent snapshot of the Azure virtual machine. Restore points install the extension as part of the first restore point creation operation.

- **Ensure VMSnapshot extension isn't in a failed state**: Follow the steps in Troubleshooting to verify and ensure the Azure VM snapshot extension is healthy.

- **Check if antivirus is blocking the extension**: Certain antivirus software can prevent extensions from executing.

  At the time of the restore point failure, verify if there are log entries in **Event Viewer Application logs** with *faulting application name: IaaSBcdrExtension.exe*. If you see entries, the antivirus configured in the VM could be restricting the execution of the VMSnapshot extension. Test by excluding the following directories in the antivirus configuration and retry the restore point operation.
  - `C:\Packages\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot`
  - `C:\WindowsAzure\Logs\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot`

- **Check if network access is required**: Extension packages are downloaded from the Azure Storage extension repository and extension status uploads are posted to Azure Storage. Learn more.
  - If you're on a non-supported version of the agent, you need to allow outbound access to Azure storage in that region from the VM.
  - If you've blocked access to `168.63.129.16` using the guest firewall or with a proxy, extensions will fail regardless of the above. Ports 80, 443, and 32526 are required. Learn more.

- **Ensure DHCP is enabled inside the guest VM**: This is required to get the host or fabric address from DHCP for the restore point to work. If you need a static private IP, you should configure it through the **Azure portal**, or **PowerShell** and make sure the DHCP option inside the VM is enabled. Learn more.

- **Ensure the VSS writer service is up and running**: Follow these steps to troubleshoot VSS writer issues.

Common issues

DiskRestorePointUsedByCustomer - There is an active shared access signature outstanding for disk restore point

Error code: DiskRestorePointUsedByCustomer

Error message: There is an active shared access signature outstanding for disk restore point. Call EndGetAccess before deleting the restore point.

You can't delete a restore point if there are active Shared Access Signatures (SAS) on any of the underlying disk restore points. End the shared access on the disk restore points and retry the operation.

OperationNotAllowed - Changes were made to the Virtual Machine while the operation 'Create Restore Point' was in progress.

Error code: OperationNotAllowed

Error message: Changes were made to the Virtual Machine while the operation 'Create Restore Point' was in progress. Operation Create Restore Point cannot be completed at this time. Please try again later.

Restore point creation fails if there are changes being made in parallel to the VM model, for example, a new disk being attached or an existing disk being detached. This is to ensure data integrity of the restore point that is created. Retry creating the restore point once the VM model has been updated.

OperationNotAllowed - Operation 'Create Restore Point' is not allowed as disk(s) have not been allocated successfully.

Error code: OperationNotAllowed

Error message: Operation 'Create Restore Point' is not allowed as disk(s) have not been allocated successfully. Please exclude these disk(s) using excludeDisks property and retry.

If any one of the disks attached to the VM isn't allocated properly, the restore point fails. You must exclude these disks before triggering creation of restore points for the VM. If you're using ARM processor API to create a restore point, to exclude a disk, add its identifier to the excludeDisks property in the request body. If you're using CLI, PowerShell, or Portal, set the respective parameters.

OperationNotAllowed - Creation of Restore Point of a Virtual Machine with Shared disks is not supported.

Error code: VMRestorePointClientError

Error message: Creation of Restore Point of a Virtual Machine with Shared disks is not supported. You may exclude this disk from the restore point via excludeDisks property.

Restore points are currently not supported for shared disks. You need to exclude these disks before triggering creation of restore point for the VM. If you are using ARM processor API to create restore point, to exclude a disk, add its identifier to the excludeDisks property in the request body. If you are using CLI, PowerShell, or Portal, follow the respective steps.

VMAgentStatusCommunicationError - VM agent unable to communicate with compute service

Error code: VMAgentStatusCommunicationError

Error message: VM has not reported status for VM agent or extensions.

The Azure VM agent might be stopped, outdated, in an inconsistent state, or not installed. These states prevent the creation of restore points.

- In the Azure portal, go to **Virtual Machines** > **Settings** > **Properties** and ensure that the VM **Status** is **Running** and **Agent status** is **Ready**. If the VM agent is stopped or is in an inconsistent state, restart the agent.
  - Restart the Guest Agent for Windows VMs.
  - Restart the Guest Agent for Linux VMs.
- In the Azure portal, go to **Virtual Machines** > **Settings** > **Extensions** and ensure all extensions are in **provisioning succeeded** state. If not, follow these steps to resolve the issue.

VMRestorePointInternalError - Restore Point creation failed due to an internal execution error while creating VM snapshot. Please retry the operation after some time.Internal

Error code: VMRestorePointInternalError

Error message: Restore Point creation failed due to an internal execution error while creating VM snapshot. Please retry the operation after some time.

After you trigger a restore point operation, the compute service starts the job by communicating with the VM backup extension to take a point-in-time snapshot. Any of the following conditions might prevent the snapshot from being triggered. If the snapshot isn't triggered, restore point creation will fail. Complete the following troubleshooting steps in the order listed, and then retry your operation:

Cause 1: **The agent is installed in the VM, but it's unresponsive (for Windows VMs)**

Cause 2: **The agent installed in the VM is out of date (for Linux VMs)**

Cause 3: **The snapshot status can't be retrieved, or a snapshot can't be taken**

Cause 4: **VM-Agent configuration options aren't set (for Linux VMs)**

Cause 5: **Application control solution is blocking IaaSBcdrExtension.exe**

This error could also occur when one of the extension failures puts the VM into provisioning failed state. If the above steps didn't resolve your issue, then do the following:

In the Azure portal, go to **Virtual Machines** > **Settings** > **Extensions** and ensure all extensions are in **provisioning succeeded** state. Learn more about Provisioning states.

- If any extension is in a failed state, it can interfere with the restore point operation. Ensure the extension issues are resolved and retry the restore point operation.
- If the VM provisioning state is in an updating state, it can interfere with the restore point operation. Ensure that it's healthy and retry the restore point operation.

VMRestorePointClientError - Restore Point creation failed due to COM+ error.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed due to COM+ error. Please restart windows service "COM+ System Application" (COMSysApp). If the issue persists, restart the VM.

Restore point operations fail if the COM+ service is not running or if there are any errors with this service. Restart the COM+ System Application, and restart the VM and retry the restore point operation.

VMRestorePointClientError - Restore Point creation failed due to insufficient memory available in COM+ memory quota.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed due to insufficient memory available in COM+ memory quota. Please restart windows service "COM+ System Application" (COMSysApp). If the issue persists, restart the VM.

Restore point operations fail if there's insufficient memory in the COM+ service. Restarting the COM+ System Application service and the VM usually frees up the memory. Once restarted, retry the restore point operation.

VMRestorePointClientError - Restore Point creation failed due to VSS Writers in bad state.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed due to VSS Writers in bad state. Restart VSS Writer services and reboot VM.

Restore point creation invokes VSS writers to flush in-memory IOs to the disk before taking snapshots to achieve application consistency. If the VSS writers are in bad state, it

affects the restore point creation operation. Restart the VSS writer service and restart the VM before retrying the operation.

VMRestorePointClientError - Restore Point creation failed due to failure in installation of Visual C++ Redistributable for Visual Studio 2012.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed due to failure in installation of Visual C++ Redistributable for Visual Studio 2012. Please install Visual C++ Redistributable for Visual Studio 2012. If you are observing issues with installation or if it is already installed and you are observing this error, please restart the VM to clean installation issues.

Restore point operations require Visual C++ Redistributable for Visual Studio 2021. Download Visual C++ Redistributable for Visual Studio 2012 and restart the VM before retrying the restore point operation.

VMRestorePointClientError - Restore Point creation failed as the maximum allowed snapshot limit of one or more disk blobs has been reached. Please delete some existing restore points of this VM and then retry.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed as the maximum allowed snapshot limit of one or more disk blobs has been reached. Please delete some existing restore points of this VM and then retry.

The number of restore points across the restore point collections and resource groups for a VM can't exceed 500. To create a new restore point, delete the existing restore points.

VMRestorePointClientError - Restore Point creation failed with the error "COM+ was unable to talk to the Microsoft Distributed Transaction Coordinator".

Error code: VMRestorePointClientError

Error message: Restore Point creation failed with the error "COM+ was unable to talk to the Microsoft Distributed Transaction Coordinator".

Follow these steps to resolve this error:

- Open services.msc from an elevated command prompt
- Make sure that **Log On As** value for **Distributed Transaction Coordinator** service is set to **Network Service** and the service is running.
- If this service fails to start, reinstall this service.

VMRestorePointClientError - Restore Point creation failed due to inadequate VM resources.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed due to inadequate VM resources. Increase VM resources by changing the VM size and retry the operation. To resize the virtual machine, refer https://azure.microsoft.com/blog/resize-virtual-machines/ .

Creating a restore point requires enough compute resource to be available. If you get the above error when creating a restore point, you need resize the VM and choose a higher VM size. Follow the steps in how to resize your VM . Once the VM is resized, retry the restore point operation.

VMRestorePointClientError - Restore point creation failed due to no network connectivity on the virtual machine.

Error code: VMRestorePointClientError

Error message: Restore Point creation failed due to no network connectivity on the virtual machine. Ensure that VM has network access. Either allowlist the Azure datacenter IP ranges or set up a proxy server for network access. For more information, see https://go.microsoft.com/fwlink/?LinkId=800034 . If you are already using proxy server, make sure that proxy server settings are configured correctly.

After you trigger creation of restore point, the compute service starts communicating with the VM snapshot extension to take a point-in-time snapshot. Any of the following conditions might prevent the snapshot from being triggered. If the snapshot isn't triggered, a restore point failure might occur. Complete the following troubleshooting step, and then retry your operation:

[The snapshot status can't be retrieved, or a snapshot can't be taken].(#the-snapshot-status-cant-be-retrieved-or-a-snapshot-cant-be-taken)

VMRestorePointClientError - RestorePoint creation failed since a concurrent 'Create RestorePoint' operation was triggered on the VM.

Error code: VMRestorePointClientError

Error message: RestorePoint creation failed since a concurrent 'Create RestorePoint' operation was triggered on the VM.

Your recent restore point creation failed because there's already an existing restore point being created. You can't create a new restore point until the current restore point is fully created. Ensure the restore point creation operation currently in progress is completed before triggering another restore point creation operation.

To check the restore points in progress, do the following steps:

1. Sign in to the Azure portal, select **All services**. Enter **Recovery Services** and select **Restore point collection**. The list of Restore point collections appears.
2. From the list of Restore point collections, select a Restore point collection in which the restore point is being created.
3. Select **Settings** > **Restore points** to view all the restore points. If a restore point is in progress, wait for it to complete.
4. Retry creating a new restore point.

DiskRestorePointClientError - Keyvault associated with DiskEncryptionSet is not found.

Error code: DiskRestorePointClientError

Error message: Keyvault associated with DiskEncryptionSet not found. The resource may have been deleted due to which Restore Point creation failed. Please retry the operation after re-creating the missing resource with the same name.

If you are creating restore points for a VM that has encrypted disks, you must ensure the keyvault where the keys are stored, is available. We use the same keys to create encrypted restore points.

BadRequest - This request can be made with api-version '2021-03-01' or newer

Error code: BadRequest

Error message: This request can be made with api-version '2022-03-01' or newer.

Restore points are supported only with API version 2022-03-01 or later. If you are using REST APIs to create and manage restore points, use the specified API version when calling the restore point API.

InternalError / InternalExecutionError / InternalOperationError / InternalDiskRestorePointError - An internal execution error occurred. Please retry later.

Error code: InternalError / InternalExecutionError / InternalOperationError / InternalDiskRestorePointError

Error message: An internal execution error occurred. Please retry later.

After you trigger creation of restore point, the compute service starts communicating with the VM snapshot extension to take a point-in-time snapshot. Any of the following conditions might prevent the snapshot from being triggered. If the snapshot isn't triggered, a restore point failure might occur. Complete the following troubleshooting steps in the order listed, and then retry your operation:

- **Cause 1:** **The agent is installed in the VM, but it's unresponsive (for Windows VMs)**.
- **Cause 2:** **The agent installed in the VM is out of date (for Linux VMs)**.
- **Cause 3:** **The snapshot status can't be retrieved, or a snapshot can't be taken**.
- **Cause 4:** **Compute service does not have permission to delete the old restore points because of a resource group lock**.
- **Cause 5**: There's an extension version/bits mismatch with the Windows version you're running, or the following module is corrupt:

  C:\Packages\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot\<extension version>\iaasvmprovider.dll

  To resolve this issue, check if the module is compatible with x86 (32-bit)/x64 (64-bit) version of *regsvr32.exe*, and then follow these steps:

  1. In the affected VM, go to **Control panel** > **Program and features**.
  2. Uninstall **Visual C++ Redistributable x64** for **Visual Studio 2013**.
  3. Reinstall **Visual C++ Redistributable** for **Visual Studio 2013** in the VM. To install, follow these steps:

a. Go to the folder: **C:\Packages\Plugins\Microsoft.Azure.RecoveryServices.VMSnapshot\<LatestVersion>**.

b. Search and run the **vcredist2013_x64** file to install.

4. Retry the restore point operation.

OSProvisioningClientError - Restore points operation failed due to an error. For details, see restore point provisioning error Message details

Error code: OSProvisioningClientError

Error message: OS Provisioning did not finish in the allotted time. This error occurred too many times consecutively from image. Make sure the image has been properly prepared (generalized).

This error is reported from the IaaS VM. Take necessary actions as described in the error message and retry the operation.

AllocationFailed - Restore points operation failed due to an error. For details, see restore point provisioning error Message details

Error code: AllocationFailed

Error message: Allocation failed. If you are trying to add a new VM to an Availability Set or update/resize an existing VM in an Availability Set, please note that such Availability Set allocation is scoped to a single cluster, and it is possible that the cluster is out of capacity. Learn more about improving likelihood of allocation success.

This error is reported from the IaaS VM. Take necessary actions as described in the error message and retry the operation.

Causes and solutions

The agent is installed in the VM, but it's unresponsive (for Windows VMs)

Solution

The VM agent might have been corrupted, or the service might have been stopped. Reinstalling the VM agent helps get the latest version. It also helps restart communication with the service.

1. Determine whether the Microsoft Azure Guest Agent service is running in the VM services (services.msc). Try to restart the Microsoft Azure Guest Agent service and initiate the restore point operation.
2. If the Microsoft Azure Guest Agent service isn't visible in services, in Control Panel, go to **Programs and Features** to determine whether the Microsoft Azure Guest Agent service is installed.
3. If the Microsoft Azure Guest Agent appears in **Programs and Features**, uninstall the Microsoft Azure Guest Agent.
4. Download and install the latest version of the agent MSI. You must have Administrator rights to complete the installation.
5. Verify that the Microsoft Azure Guest Agent services appear in services.
6. Retry the restore point operation.

Also, verify that Microsoft .NET 4.5 is installed in the VM. .NET 4.5 is required for the VM agent to communicate with the service.

The agent installed in the VM is out of date (for Linux VMs)

Solution

Most agent-related or extension-related failures for Linux VMs are caused by issues that affect an outdated VM agent. To troubleshoot this issue, follow these general guidelines:

1. Follow the instructions for updating the Linux VM agent.

   > **Note**
   >
   > We *strongly recommend* that you update the agent only through a distribution repository. We don't recommend downloading the agent code directly from GitHub and updating it. If the latest agent for your distribution is not available, contact distribution support for instructions on how to install it. To check for the most recent agent, go to the Windows Azure Linux agent page in the GitHub repository.

2. Ensure that the Azure agent is running on the VM by running the following command: `ps -e`

If the process isn't running, restart it by using the following commands:

- For Ubuntu/Debian:

```
Bash

sudo systemctl start walinuxagent
```

- For other Linux distributions:

```
Bash

sudo systemctl start waagent
```

3. Configure the auto restart agent.
4. Retry the restore point operation. If the failure persists, collect the following logs from the VM:
   - /var/lib/waagent/*.xml
   - /var/log/waagent.log
   - /var/log/azure/*

If you require verbose logging for waagent, follow these steps:

1. In the `/etc/waagent.conf` file, locate the following line: **Enable verbose logging (y|n)**.
2. Change the **Logs.Verbose** value from *n* to *y*.
3. Save the change, and then restart waagent by completing the steps described earlier in this section.

VM-Agent configuration options are not set (for Linux VMs)

A configuration file (`/etc/waagent.conf`) controls the actions of waagent. Configuration File Options **Extensions.Enable** should be set to **y** and **Provisioning.Agent** should be set to **auto** for restore points to work. For the full list of VM-Agent Configuration File Options, see https://github.com/Azure/WALinuxAgent#configuration-file-options.

Application control solution is blocking IaaSBcdrExtension.exe

If you're running AppLocker (or another application control solution), and the rules are publisher or path based, they may block the **IaaSBcdrExtension.exe** executable from running.

Solution

Exclude the `/var/lib` path or the **IaaSBcdrExtension.exe** executable from AppLocker (or other application control software.)

The snapshot status can't be retrieved, or a snapshot can't be taken

Restore points rely on issuing a snapshot command to the underlying storage account. Restore point can fail either because it has no access to the storage account, or because the execution of the snapshot task is delayed.

Solution

The following conditions might cause the snapshot task to fail:

Expand table

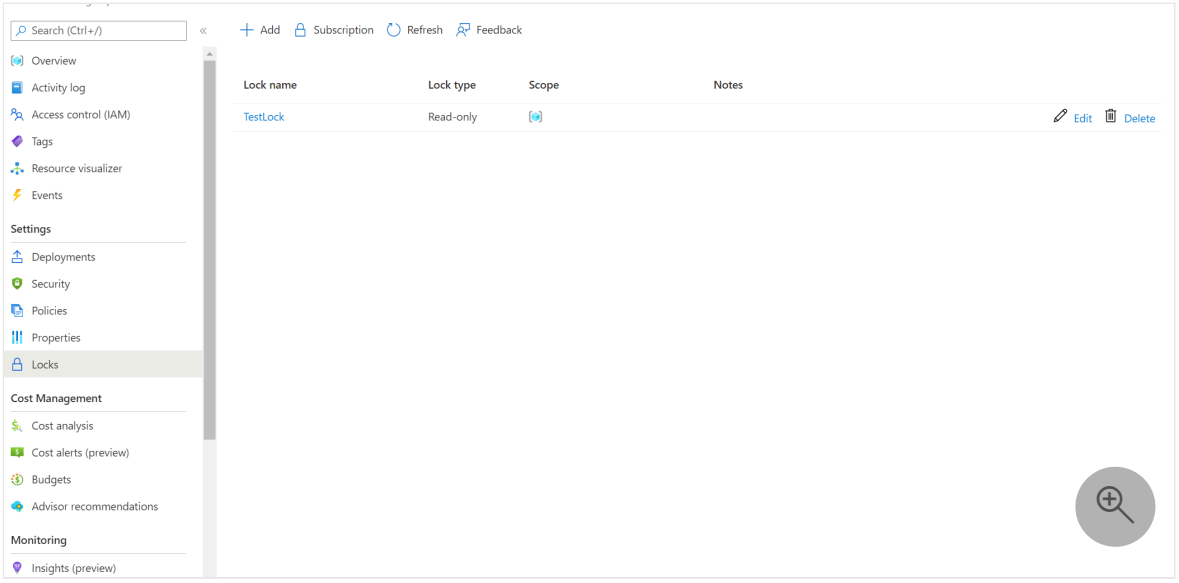
Cause	Solution
The VM status is reported incorrectly because the VM is shut down in Remote Desktop Protocol (RDP).	If you shut down the VM in RDP, check the portal to determine whether the VM status is correct. If it's not correct, shut down the VM in the portal by using the **Shutdown** option on the VM dashboard.
The VM can't get the host or fabric address from DHCP.	DHCP must be enabled inside the guest for restore point to work. If the VM can't get the host or fabric address from DHCP response 245, it can't download or run any extensions. If you need a static private IP, you should configure it through the **Azure portal**, or **PowerShell** and make sure the DHCP option inside the VM is enabled. Learn more about setting up a static IP address with PowerShell.

Remove lock from the recovery point resource group

1. Sign in to the Azure portal.
2. Go to **All Resources**, select the restore point collection resource group.

3. In the **Settings** section, select **Locks** to display the locks.

4. To remove the lock, select **Delete**.



Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Move Azure resources to a new resource group or subscription

07/29/2025

This article explains how to move Azure resources between resource groups within the same subscription or across different subscriptions. If the move involves different subscriptions, both subscriptions must be part of the same Microsoft Entra ID tenant. You can use tools like the Azure portal, Azure PowerShell, Azure CLI, the REST API, or Python to move the resources.

During the move operation, both the source and target resource groups are locked. You can't create, delete, or update resources within these resource groups while the move is in progress. However, existing resources remain fully operational. For example, if you move a virtual machine from one resource group to another, you can't delete it or modify its properties (such as its size) during the move. Despite this restriction, the virtual machine continues to operate normally, and services relying on it don't experience any downtime. The lock can last up to four hours. Most moves are completed faster, and the lock is removed accordingly.

Only top-level (parent) resources should be specified in the move request. Child resources move automatically with their parent but can't be moved independently. For example, you can move a parent resource like `Microsoft.Compute/virtualMachines`, and its child resource such as `Microsoft.Compute/virtualMachines/extensions` moves with it. However, you can't move the child resource on its own.

While moving a resource preserves its dependencies with child resources, dependencies with other resources can break and might need to be configured again. Moving a resource only changes its associated resource group and doesn't alter the physical region of the resource.

> **Note**
>
> Azure resources can't be moved if a read-only lock exists on the source, destination resource group, or subscription.

Changed resource ID

When you move a resource, you change its resource ID. The standard format for a resource ID is `/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/{resourceProviderNamespace}/{resourceType}/{resourceName}`. When you move a resource to a new resource group or subscription, you change one or more values in that path.

If you use the resource ID anywhere, change that value. For example, if you have a custom dashboard in the portal that references a resource ID, update that value. Look for any scripts or templates that need to be updated for the new resource ID.

Checklist before moving resources

Some important steps precede moving a resource. You can avoid errors if you verify these conditions.

1. The source and destination subscriptions must be active. If you have trouble enabling an account that's disabled, create an Azure support request. Select **Subscription Management** for the issue type.

2. The source and destination subscriptions must exist within the same Microsoft Entra tenant. Use the Azure CLI or PowerShell to check that both subscriptions have the same tenant ID.

   Azure CLI

   Azure CLI

   ```
   az account show --subscription <your-source-subscription> --query
   tenantId
   az account show --subscription <your-destination-subscription> --query
   tenantId
   ```

   If the tenant IDs for the source and destination subscriptions don't match, use the following methods to reconcile them:

   - Transfer billing ownership of an Azure subscription.
   - Associate or add an Azure subscription to your Microsoft Entra tenant.

3. To move resources to or from a Cloud Solution Provider (CSP) partner, see Transfer Azure subscriptions between subscribers and CSPs.

4. The resources you want to move must support the move operation. For a list of which resources support move operations, see Azure resource types for move operations.

5. Some services have specific limitations or requirements when moving resources. Check the following move guidance before moving resources within these services:

   - If you're using Azure Stack Hub, you can't move resources between groups.
     - Azure App Services

- Azure DevOps Services
- Classic deployment model for classic compute, storage, virtual networks, and cloud services
- Cloud Services (extended support)
- Networking
- Azure Recovery Services
- Virtual machines
- To move an Azure subscription to a new management group, see Move subscriptions.

6. The destination subscription must be registered for the resource provider of the resource you're moving. If it's not, you receive an error stating that the **subscription isn't registered for a resource type**. You might see this error when moving a resource to a new subscription, but you didn't previously use the resource type in the subscription.

   Azure CLI

   To get the registration status:

   Azure CLI

   ```
   az account set -s <destination-subscription-name-or-id>
   az provider list --query "[].{Provider:namespace,
   Status:registrationState}" --out table
   ```

   To register a resource provider:

   Azure CLI

   ```
   az provider register --namespace Microsoft.Batch
   ```

7. Before starting a move operation, check the subscription quota for the subscription to which you're moving resources. Verify if you can request an increase in a quota that would cause a destination subscription to exceed its limit. For detailed guidance about limits and how to request an increase, see Azure subscription and service limits, quotas, and constraints.

8. The account moving the resources must have at least the following permissions:

   - At the source resource group: **Microsoft.Resources/subscriptions/resourceGroups/moveResources/action**

- At the destination resource group:
  Microsoft.Resources/subscriptions/resourceGroups/write

9. If you move a resource with an active Azure role assignment (or its child resource with this same assignment), the role assignment doesn't move and becomes orphaned. You must create the role assignment again after the move. Although the system automatically removes the orphaned role assignment, we recommend that you remove it before the move.

   To learn more about how to manage role assignments, see List Azure role assignments and Assign Azure roles.

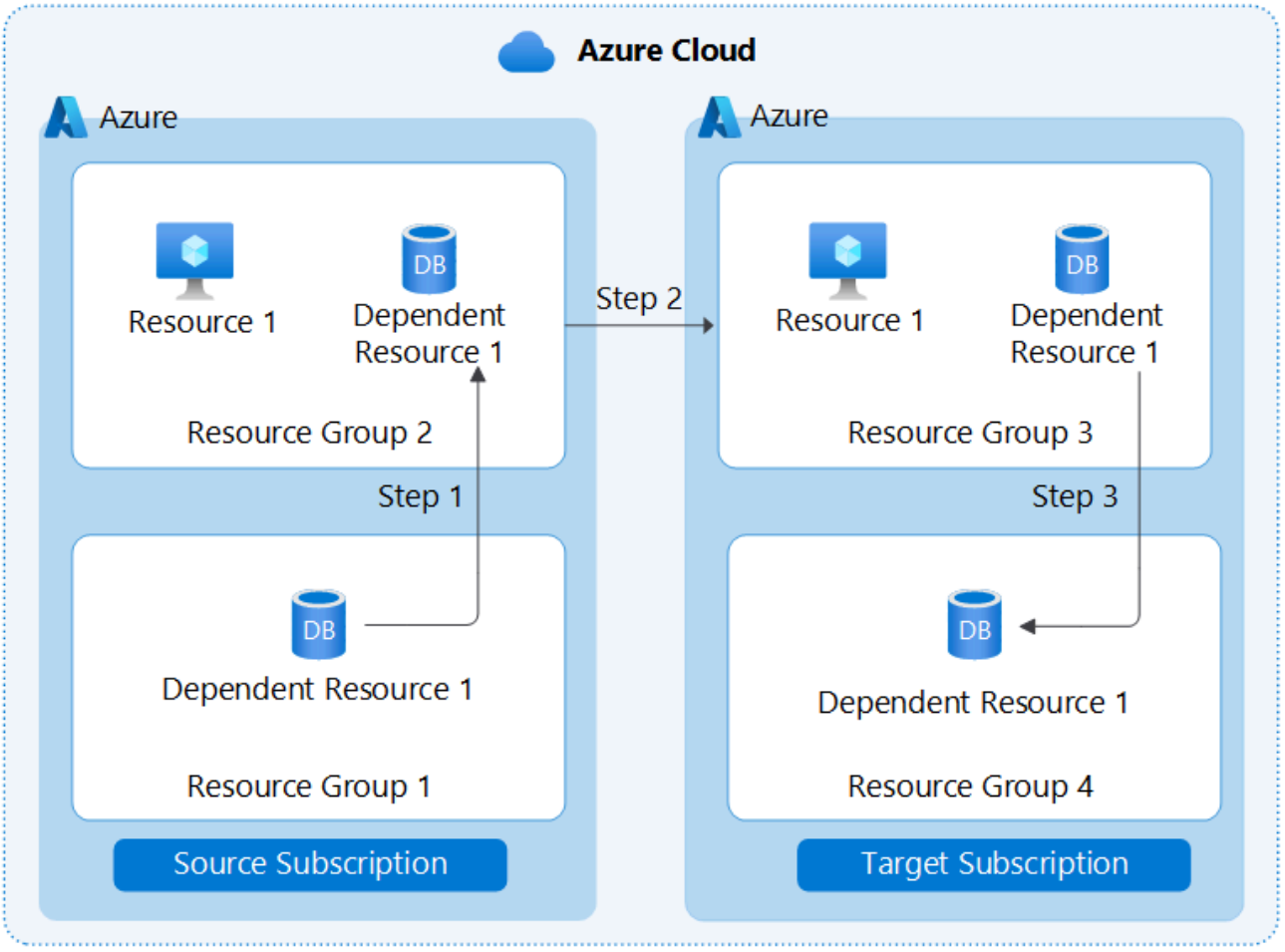
10. **For a move across subscriptions, the resource and its dependent resources must be located in the same resource group and they must be moved together.** For example, a virtual machine with managed disks requires you to move the virtual machine, managed disks, and other dependent resources together.

    If you're moving a resource to a new subscription, check if the resource has any dependent resources and if they're located in the same resource group. If the resources aren't in the same resource group, check if you can combine them into the same resource group. If you can, use one move operation across resource groups to consolidate all the resources into the same resource group.

    For more information, see Scenario for move across subscriptions.

Scenario for moving across subscriptions

Moving resources from one subscription to another is a three-step process. To illustrate these steps, the following diagram depicts only one dependent resource:

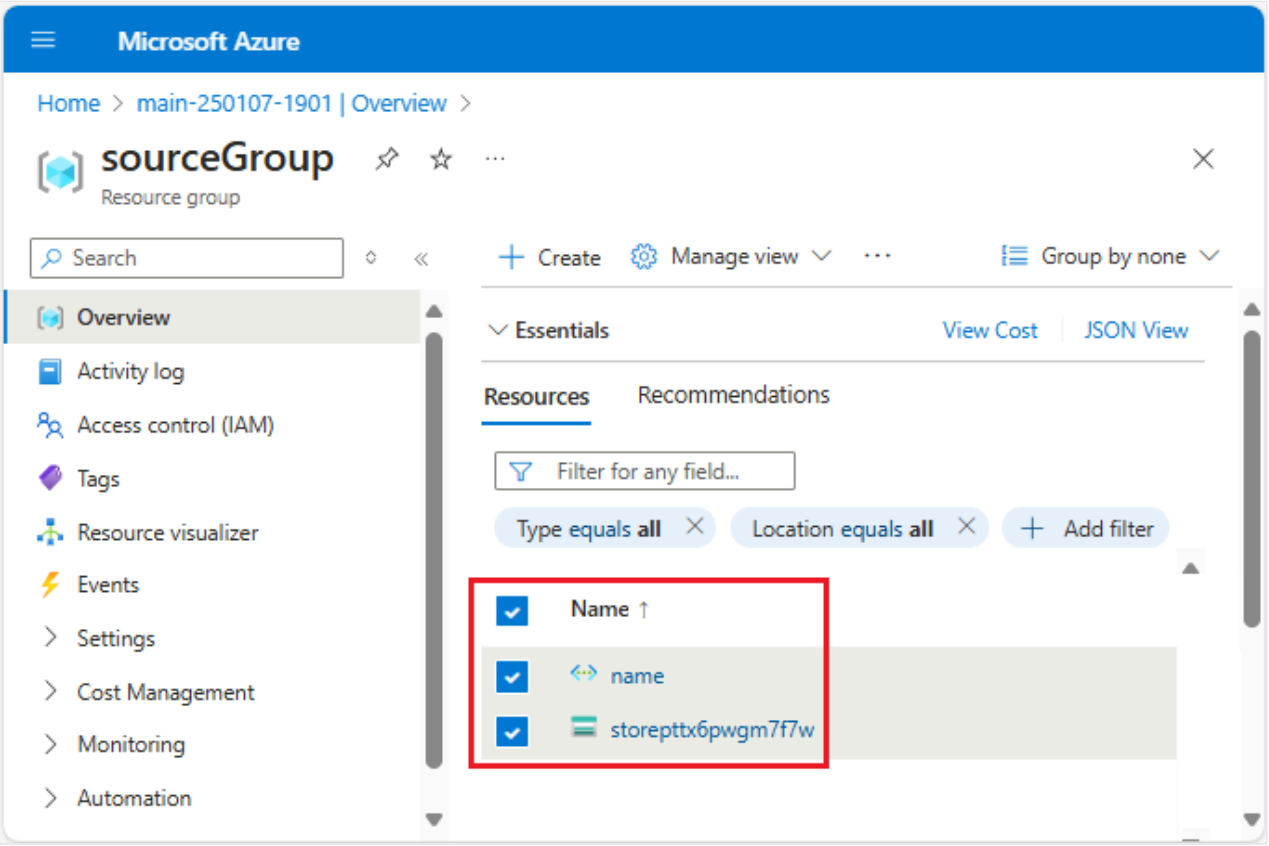


- Step 1: If dependent resources are distributed across different resource groups, first move them into one resource group.
- Step 2: Move the resource and dependent resources together from the source subscription to the target subscription.
- Step 3: Optionally, redistribute the dependent resources to different resource groups within the target subscription.

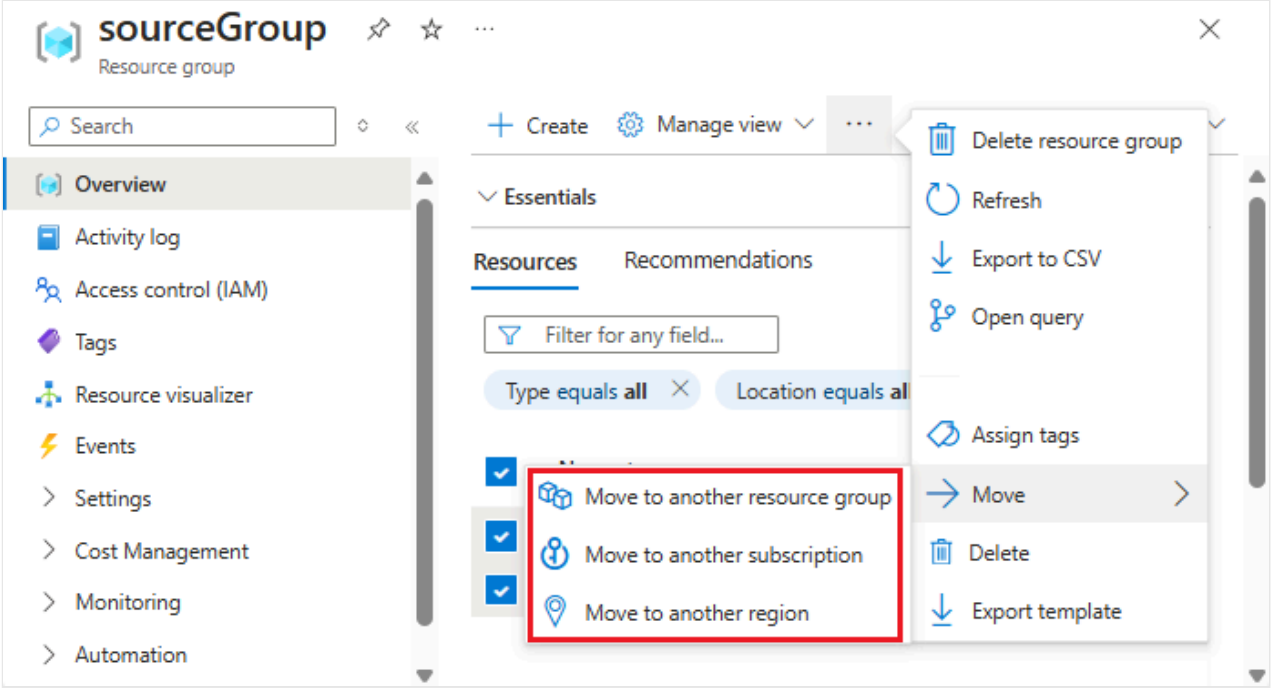
Move resources

Use the Azure portal

1. To move resources, select the resource group that contains those resources.
2. Select the resources that you want to move. To move all of the resources, select the checkbox at the top of list. Or, select resources individually.



3. Select the **Move** button.

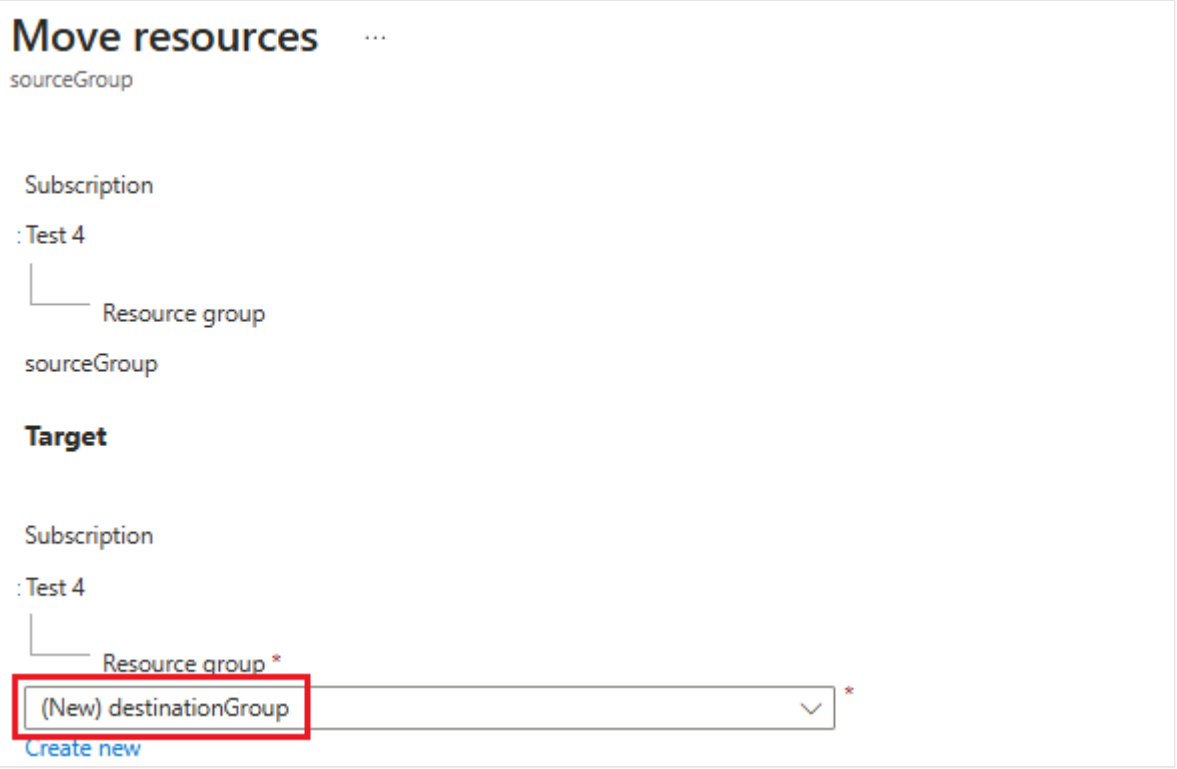


This button gives you three options:

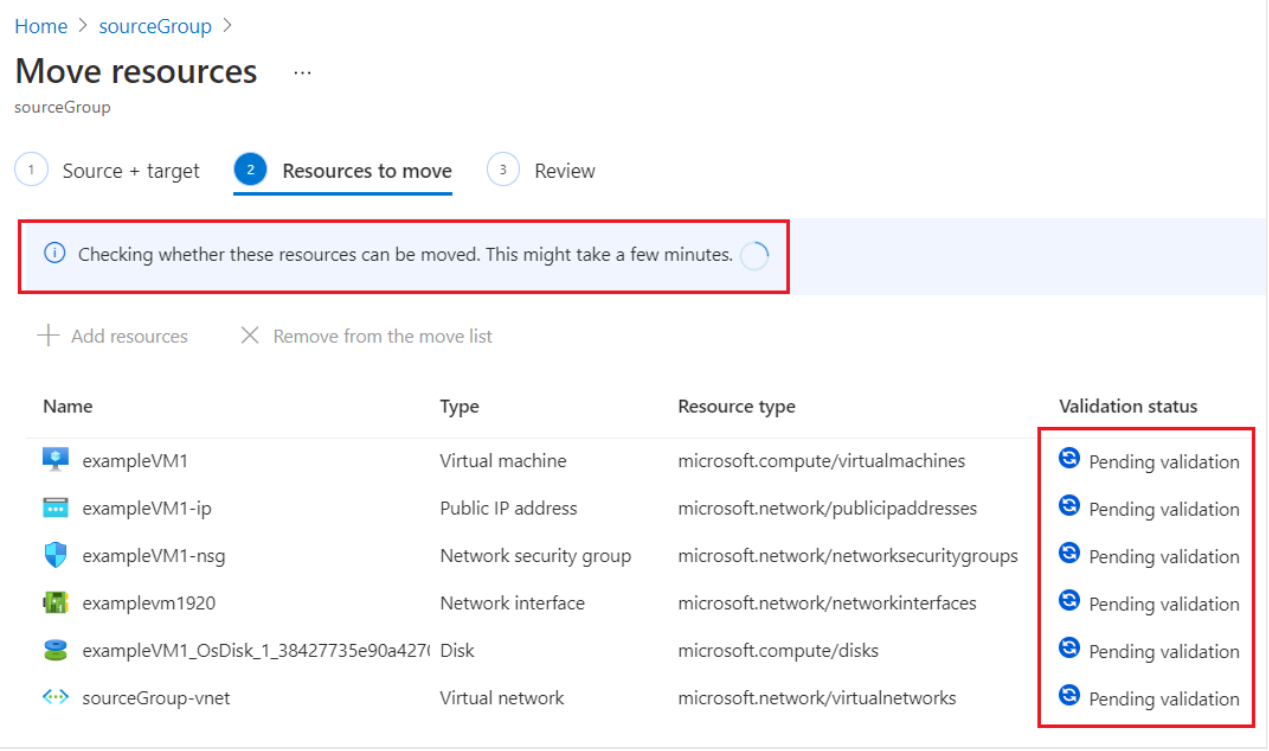
- Move to a new resource group.
- Move to a new subscription.
- Move to a new region. To change regions, see Move resources across regions (from resource group) with Azure Resource Mover.

4. Select if you're moving the resources to a new resource group or subscription.

5. The source resource group sets automatically. Specify the destination resource group. If you're moving to a new subscription, specify this option. Select **Next**.

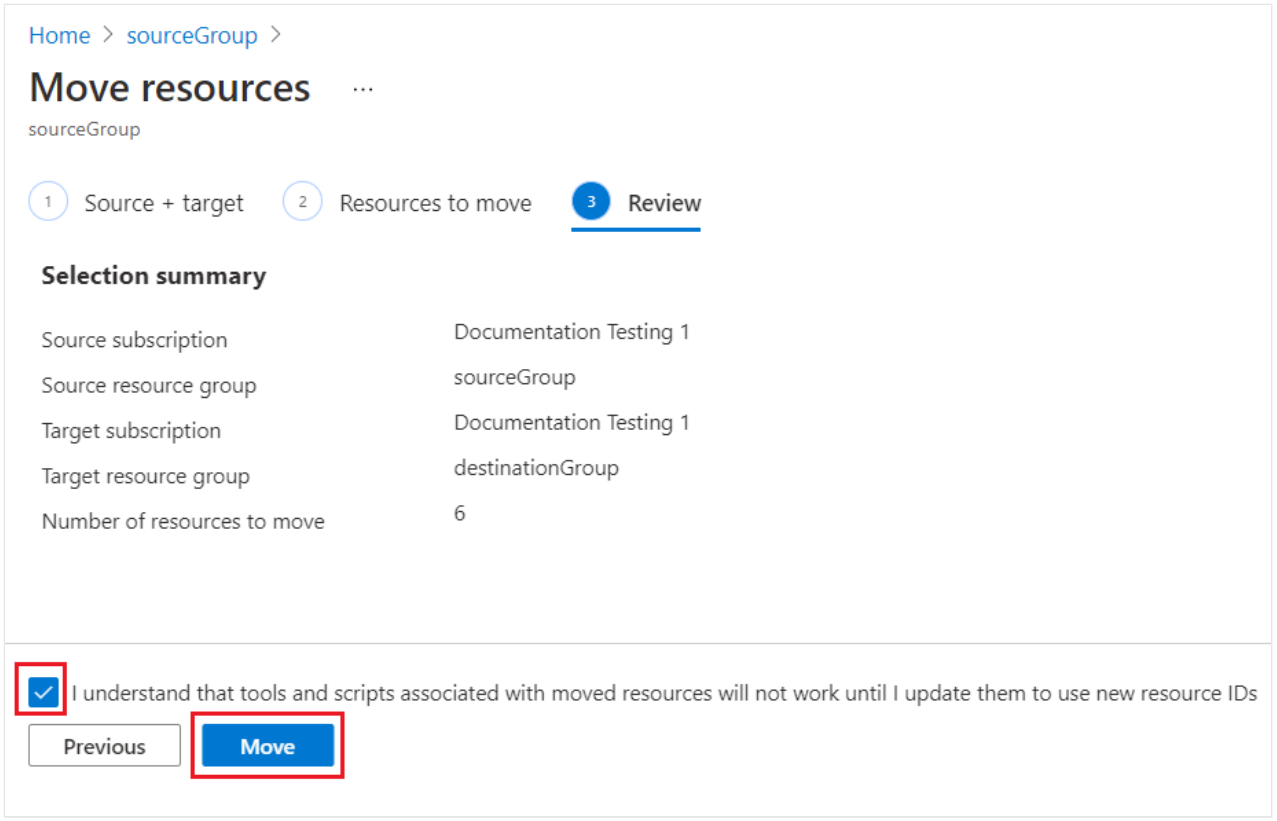


6. The portal validates that the resources can be moved. Wait for validation to complete.

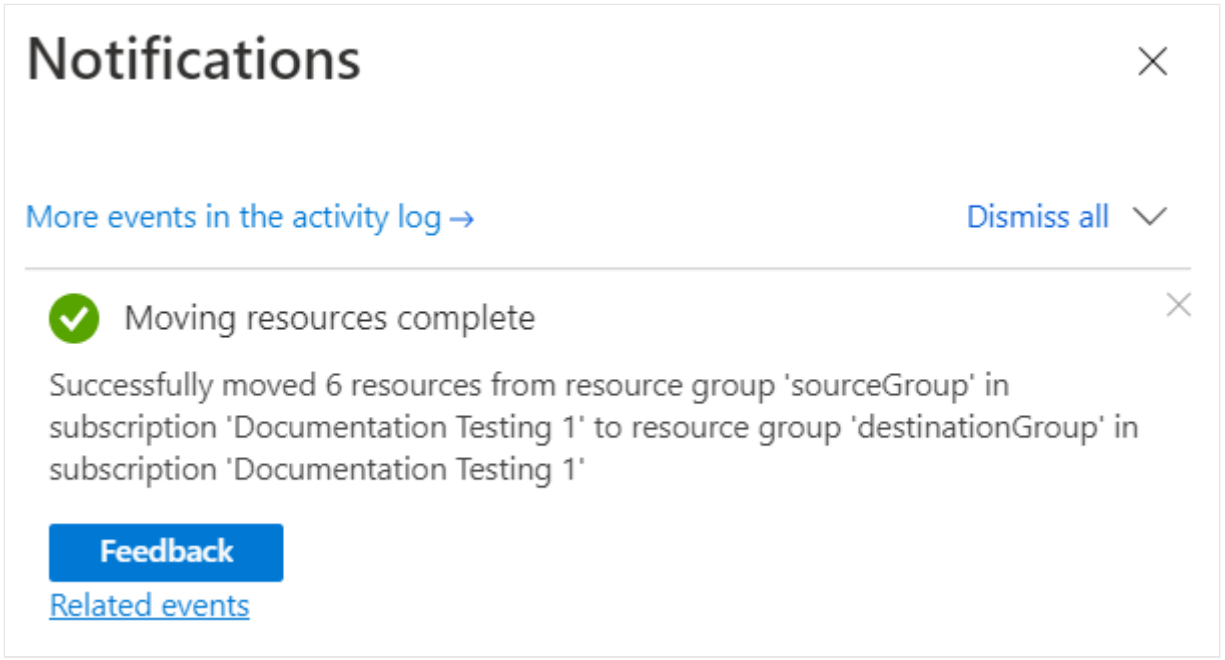


7. When validation completes successfully, select **Next**.

8. Acknowledge that you need to update tools and scripts for these resources. To start moving the resources, select **Move**.



9. The Azure portal notifies you when the move completes.



Use the Azure CLI

Validate

To test your move scenario without actually moving resources in real time, use the az resource invoke-action command. Use this command only when you need to model the results without following through. To run this operation, you need the resource ID of the source resource group, target resource group, and each resource that you're moving.

Use `\"` to escape double quotes in the body of the request.

Azure CLI

```
az resource invoke-action --action validateMoveResources \
  --ids "/subscriptions/{subscription-id}/resourceGroups/{source-rg}" \
  --request-body "{  \"resources\": [\"/subscriptions/{subscription-
id}/resourceGroups/{source-rg}/providers/{resource-provider}/{resource-
type}/{resource-name}\", \"/subscriptions/{subscription-
id}/resourceGroups/{source-rg}/providers/{resource-provider}/{resource-
type}/{resource-name}\", \"/subscriptions/{subscription-
id}/resourceGroups/{source-rg}/providers/{resource-provider}/{resource-
type}/{resource-name}\"],\"targetResourceGroup\":\"/subscriptions/{subscription-
id}/resourceGroups/{destination-rg}\" }"
```

If validation passes, you see:

Azure CLI

```
{} Finished ..
```

If validation fails, you see an error message that explains why you can't move the resources.

Move

To move existing resources to another resource group or subscription, use the az resource move command. In the `--ids` parameter, provide a space-separated list of the resource IDs to move.

The following commands show how to move several resources to a new resource group. They work with the Azure CLI in a Bash terminal or in an Azure PowerShell console. To move resources to a new subscription, provide the `--destination-subscription-id` parameter.

Azure CLI

Azure CLI

```
webapp=$(az resource show -g OldRG -n ExampleSite --resource-type
"Microsoft.Web/sites" --query id --output tsv)
plan=$(az resource show -g OldRG -n ExamplePlan --resource-type
"Microsoft.Web/serverfarms" --query id --output tsv)
az resource move --destination-group newgroup --ids $webapp $plan
```

Use Azure PowerShell

Validate

To test your move scenario without actually moving resources in real time, use the Invoke-AzResourceAction command in Azure PowerShell. Use this command only when you need to model the results without following through.

Azure PowerShell

```
$sourceName = "sourceRG"
$destinationName = "destinationRG"
$resourcesToMove = @("app1", "app2")

$sourceResourceGroup = Get-AzResourceGroup -Name $sourceName
$destinationResourceGroup = Get-AzResourceGroup -Name $destinationName

$resources = Get-AzResource -ResourceGroupName $sourceName | Where-Object {
$_.Name -in $resourcesToMove }

Invoke-AzResourceAction -Action validateMoveResources `
  -ResourceId $sourceResourceGroup.ResourceId `
  -Parameters @{
  resources = $resources.ResourceId;  # Wrap in an @() array if providing a single
resource ID string.
  targetResourceGroup = $destinationResourceGroup.ResourceId
  }
```

An output doesn't display if the validation succeeds. However, if the validation fails, an error message explains why you can't move the resources.

Move

To move existing resources to another resource group or subscription, use the Move-AzResource command. The following example shows how to move several resources to a new resource group.

Azure PowerShell

```
$sourceName = "sourceRG"
$destinationName = "destinationRG"
$resourcesToMove = @("app1", "app2")

$resources = Get-AzResource -ResourceGroupName $sourceName | Where-Object {
$_.Name -in $resourcesToMove }
```

```
Move-AzResource -DestinationResourceGroupName $destinationName -ResourceId
$resources.ResourceId
```

To move to a new subscription, include a value for the `DestinationSubscriptionId` parameter.

Use Python

Validate

To test your move scenario without actually moving resources in real time, use the ResourceManagementClient.resources.begin_validate_move_resources method. Use this method only when you need to model the results without following through.

Python

```python
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ResourceManagementClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

resource_client = ResourceManagementClient(credential, subscription_id)

source_name = "sourceRG"
destination_name = "destinationRG"
resources_to_move = ["app1", "app2"]

destination_resource_group = resource_client.resource_groups.get(destination_name)

resources = [
  resource for resource in
resource_client.resources.list_by_resource_group(source_name)
  if resource.name in resources_to_move
]

resource_ids = [resource.id for resource in resources]

validate_move_resources_result =
resource_client.resources.begin_validate_move_resources(
  source_name,
  {
  "resources": resource_ids,
  "target_resource_group": destination_resource_group.id
  }
).result()

print("Validate move resources result: {}".format(validate_move_resources_result))
```

An output doesn't display if the validation succeeds. However, if the validation fails, an error message explains why you can't move the resources.

Move

To move existing resources to another resource group or subscription, use the ResourceManagementClient.resources.begin_move_resources method in Python. The following example shows how to move several resources to a new resource group.

Python

```
import os
from azure.identity import AzureCliCredential
from azure.mgmt.resource import ResourceManagementClient

credential = AzureCliCredential()
subscription_id = os.environ["AZURE_SUBSCRIPTION_ID"]

resource_client = ResourceManagementClient(credential, subscription_id)

source_name = "sourceRG"
destination_name = "destinationRG"
resources_to_move = ["app1", "app2"]

destination_resource_group = resource_client.resource_groups.get(destination_name)

resources = [
  resource for resource in
resource_client.resources.list_by_resource_group(source_name)
  if resource.name in resources_to_move
]

resource_ids = [resource.id for resource in resources]

resource_client.resources.begin_move_resources(
  source_name,
  {
  "resources": resource_ids,
  "target_resource_group": destination_resource_group.id
  }
)
```

Use REST API

Validate

The validate move operation operation tests your move scenario without actually moving resources. Use this operation to check if the move can succeed. Validation is automatically

called when you send a move request. Use this operation only when you need to model the results without following through. To run this operation, you need the:

- Name of the source resource group
- Resource ID of the target resource group
- Resource ID of each resource to move
- The access token for your account

Send the following request:

HTTP

```
POST https://management.azure.com/subscriptions/<subscription-id>/resourceGroups/<source-group>/validateMoveResources?api-version=2019-05-10
Authorization: Bearer <access-token>
Content-type: application/json
```

With a request body:

JSON

```
{
 "resources": ["<resource-id-1>", "<resource-id-2>"],
 "targetResourceGroup": "/subscriptions/<subscription-id>/resourceGroups/<target-group>"
}
```

If the request is formatted correctly, the operation returns:

HTTP

```
Response Code: 202
cache-control: no-cache
pragma: no-cache
expires: -1
location: https://management.azure.com/subscriptions/<subscription-id>/operationresults/<operation-id>?api-version=2018-02-01
retry-after: 15
...
```

A **202** status code indicates that the validation request was accepted, but it hasn't yet determined if the move operation will succeed. The `location` value contains a URL that you use to check the status of the long-running operation.

To check the status, send the following request:

HTTP

```
GET <location-url>
Authorization: Bearer <access-token>
```

You continue to receive a **202** status code while the operation runs. Wait the number of seconds indicated in the `retry-after` value before trying again. You receive a **204** status code if the move validation succeeds. If the move validation fails, you receive an error message that resembles:

JSON

```
{"error":{"code":"ResourceMoveProviderValidationFailed","message":"<message>"...}}
```

Move

To move existing resources to another resource group or subscription, use the Move resources operation.

HTTP

```
POST https://management.azure.com/subscriptions/{source-subscription-id}/resourcegroups/{source-resource-group-name}/moveResources?api-version={api-version}
```

Specify the target resource group and resources to move in the body of the request.

JSON

```
{
 "resources": ["<resource-id-1>", "<resource-id-2>"],
 "targetResourceGroup": "/subscriptions/<subscription-id>/resourceGroups/<target-group>"
}
```

Frequently asked questions

My resource move operation that usually takes a few minutes is running for almost an hour. Is something wrong?

Moving a resource is a complex operation with different phases. It can involve more than just the resource provider of the resource you're trying to move. Azure Resource Manager allows a move operation four hours to finish because of the dependencies between resource providers.

This duration gives them time to recover from transient issues. If your move request is within the four-hour period, the operation keeps trying to complete and might succeed. The operation locks the source and destination resource groups during this time to avoid consistency issues.

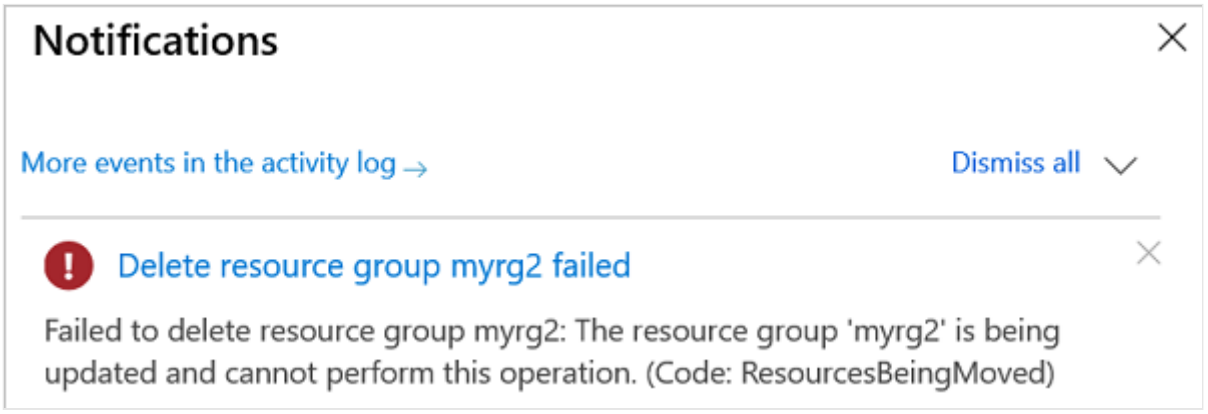
Why is my resource group locked for four hours during resource move?

- Move operations are allowed four hours to complete. The operation locks the source and destination resource groups during this time to prevent them from being modified.
- There are two phases in a move request. Resources move during the first phase, and resource providers that depend on the resources being moved are notified during the second phase. A resource group can be locked for all four hours when a resource provider fails either phase. Resource Manager initiates any failed steps during the span of the move operation.
- Resource Manager unlocks both resource groups if a resource doesn't move within four hours. Resources that move successfully are in the destination resource group. Resources that fail to move remain in the source resource group.

What are the implications of the source and destination resource groups being locked during the resource move?

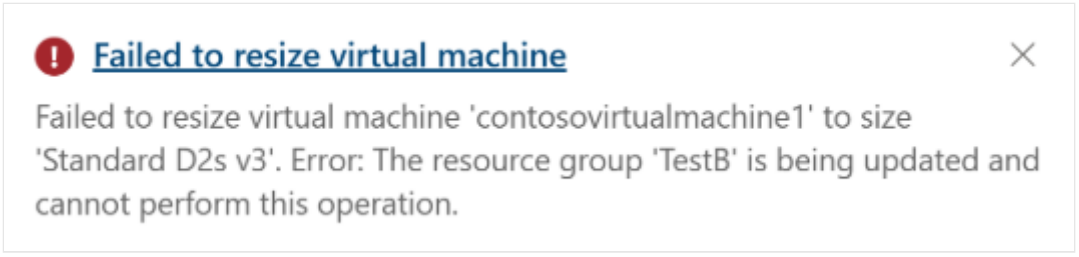
The lock prevents you from deleting either resource group. The lock also prevents you from creating a new resource, deleting a resource, or updating a resource's properties within each resource group (for example, changing a virtual machine's size).

The following image shows an error message from the Azure portal when you try to delete a resource group that's part of an ongoing move:



In the preceding image, the virtual machine resource belongs to a resource group ("TestB") that's currently undergoing a move operation. When you attempt to update a virtual machine's

property (such as its size), the Azure portal returns an error message. This error occurs because the resource group is locked during the move, which protects its resources from being modified.



Additionally, the source and the destination resource groups can't participate in other move operations simultaneously during a resource move. For example, if you're moving resources from resource group A to resource group B, both resource group A and resource group B can't be involved in another move operation at the same time. You can't simultaneously move resources to or from resource group C. This restriction prevents multiple conflicting operations from locking resource groups during the move process.

What does the error code "MissingMoveDependentResources" mean?

When you move a resource, its dependent resources must exist in the destination resource group or subscription, or be included in the move request. You get the ### MissingMoveDependentResources** error code when a dependent resource doesn't meet this requirement. The error message provides details about the dependent resource that you need to include in the move request.

For example, moving a virtual machine could require moving seven resource types with three different resource providers. Those resource providers and types are:

- Microsoft.Compute
  - virtualMachines
  - disks
- Microsoft.Network
  - networkInterfaces
  - publicIPAddresses
  - networkSecurityGroups
  - virtualNetworks
- Microsoft.Storage
  - storageAccounts

Another common example involves moving a virtual network where you might have to move several other resources associated with that virtual network. The move request could require moving public IP addresses, route tables, virtual network gateways, network security groups, and other resources. A virtual network gateway should be in the same resource group as its virtual network since they can't move separately.

What does the error code "RequestDisallowedByPolicy" mean?

Resource Manager validates your move request before attempting a move. This validation includes checking policies defined for the resources involved in the move. For example, the validation fails when you're attempting to move a key vault but your organization has a policy to deny creating a key vault in the target resource group. The returned error code is ### RequestDisallowedByPolicy**.

For more information about policies, see What is Azure Policy?.

Why can't I move some resources in Azure?

Not all Azure resources allow move operations.

How many resources can I move in one operation?

Separate large moves into different move operations when possible. Resource Manager immediately returns an error when there are more than 800 resources in a single operation. However, moving fewer than 800 resources can also fail by timing out.

What is the meaning of the error that a resource isn't in a "succeeded" state?

When you get an error message indicating that you can't move a resource because it isn't in a **Succeeded** state, it might be because a dependent resource is blocking the move. Typically, the error code is **MoveCannotProceedWithResourcesNotInSucceededState**.

If the source or target resource group contains a virtual network, the states of all resources that depend on that virtual network are checked during the move. This check includes resources that directly and indirectly depend on the network. The move is blocked if any resources aren't in a succeeded state. For example, if a virtual machine using a virtual network that doesn't report a succeeded state, the move is blocked. The move is blocked even when the virtual machine isn't one of the resources being moved. The move is also blocked even when the virtual machine isn't in the source or destination resource group.

To resolve this issue, move your resources to a resource group that doesn't have a virtual network or contact support.

Can I move a resource group to a different subscription?

No, you can't move a resource group to a new subscription. But, you can move all resources in a resource group to a resource group in another subscription. Settings such as tags, role assignments, and policies don't transfer automatically from the original resource group to the destination resource group. You need to apply these settings manually to the new resource group.

Next steps

To verify which Azure resources support move operations, see Move operation support for resources.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

Move a Marketplace Azure Virtual Machine to another subscription

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

To move a Marketplace virtual machine to a different subscription, you must move the OS disk to that subscription and then recreate the virtual machine.

You don't need this procedure to move a data disk to a new subscription. Instead, create a new virtual machine in the new subscription from the Marketplace, then move and attach the data disk.

This script demonstrates three operations:

- Create a snapshot of an OS disk.
- Move the snapshot to a different subscription.
- Create a virtual machine based on that snapshot.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

Sample script

To move a Marketplace virtual machine to a different subscription, you must create a new virtual machine for the same Marketplace offer from the moved OS disk.

> **Note**
>
> If the virtual machine plan is no longer available in the Marketplace, you can't use this procedure.

Azure CLI

```
#!/bin/bash
# Set variable values before proceeding.

# Variables
sourceResourceGroup= Resource group for the current virtual machine
sourceSubscription= Subscription for the current virtual machine
vmName= Name of the current virtual machine

destinationResourceGroup= Resource group for the new virtual machine, create
if necessary
destinationSubscription= Subscription for the new virtual machine

# Set your current subscription for the source virtual machine
az account set --subscription $sourceSubscription

# Load variables about your virtual machine
# osType = windows or linux
osType=$(az vm get-instance-view --resource-group $sourceResourceGroup \
    --name $vmName --subscription $sourceSubscription \
    --query 'storageProfile.osDisk.osType' --output tsv)

# offer = Your offer in Marketplace
offer=$(az vm get-instance-view --resource-group $sourceResourceGroup \
    --name $vmName --query 'storageProfile.imageReference.offer' --output
tsv)

# plan = Your plan in Marketplace
plan=$(az vm get-instance-view --resource-group $sourceResourceGroup \
    --name $vmName --query 'plan' --output tsv)

# publisher = Your publisher in Marketplace
publisher=$(az vm get-instance-view --resource-group $sourceResourceGroup \
    --name $vmName --query 'storageProfile.imageReference.publisher' --
output tsv)

# Get information to create new virtual machine
planName=$(az vm get-instance-view --resource-group $sourceResourceGroup \
    --subscription $sourceSubscription --query 'plan.name' --name $vmName)
planProduct=$(az vm get-instance-view --resource-group $sourceResourceGroup
\
    --subscription $sourceSubscription --query 'plan.product' --name
```

```
$vmName)
planPublisher=$(az vm get-instance-view --resource-group
$sourceResourceGroup \
    --subscription $sourceSubscription --query 'plan.publisher' --name
$vmName)

# Get the name of the OS disk
osDiskName=$(az vm show --resource-group $sourceResourceGroup --name $vmName
\
    --query 'storageProfile.osDisk.name' --output tsv)

# Verify the terms for your market virtual machine
az vm image terms show --offer $offer --plan '$plan' --publisher $publisher
\
    --subscription $sourceSubscription

# Deallocate the virtual machine
az vm deallocate --resource-group $sourceResourceGroup --name $vmName

# Create a snapshot of the OS disk
az snapshot create --resource-group $sourceResourceGroup --name
MigrationSnapshot \
    --source
"/subscriptions/$sourceSubscription/resourceGroups/$sourceResourceGroup/prov
iders/Microsoft.Compute/disks/$osDiskName"

# Move the snapshot to your destination resource group
az resource move --destination-group $destinationResourceGroup \
    --destination-subscription-id $destinationSubscription \
    --ids
"/subscriptions/$sourceSubscription/resourceGroups/$sourceResourceGroup/prov
iders/Microsoft.Compute/snapshots/MigrationSnapshot"

# Set your subscription to the destination value
az account set --subscription $destinationSubscription

# Accept the terms from the Marketplace
az vm image terms accept --offer $offer --plan '$plan' --publisher
$publisher \
    --subscription $destinationSubscription

# Create disk from the snapshot
az disk create --resource-group $destinationResourceGroup --name
DestinationDisk \
    --source
"/subscriptions/$destinationSubscription/resourceGroups/$destinationResource
Group/providers/Microsoft.Compute/snapshots/MigrationSnapshot" \
    --os-type $osType

# Create virtual machine from disk
az vm create --resource-group $destinationResourceGroup --name $vmName \
    --plan-name $planName --plan-product $planProduct  --plan-publisher
$planPublisher \
    --attach-os-disk
"/subscriptions/$destinationSubscription/resourceGroups/$destinationResource
```

```
Group/providers/Microsoft.Compute/disks/DestinationDisk" \
    --os-type $osType
```

Clean up resources

After the sample has been run, use the following commands to remove the resource groups and all associated resources:

Azure CLI

```
az group delete --name $sourceResourceGroup --subscription
$sourceSubscription
az group delete --name $destinationResourceGroup --subscription
$destinationSubscription
```

Azure CLI references used in this article

- az account set
- az disk create
- az group delete
- az resource move
- az snapshot create
- az vm create
- az vm deallocate
- az vm delete
- az vm get-instance-view
- az vm image terms accept
- az vm image terms show
- az vm show

Next steps

- Move VMs to another Azure region
- Move a VM to another subscription or resource group

Feedback

Was this page helpful? Yes No

Move VMs to another Azure region

There are scenarios in which you'd want to move your existing Azure IaaS virtual machines (VMs) from one region to another. For example, you want to improve the reliability and availability of your existing VMs, improve manageability, or move for governance reasons. For more information, see the Azure VM move overview.

You can use Azure Site Recovery service to move Azure VMs to a secondary region.

In this tutorial, you learn how to:

- ✓ Verify prerequisites for the move
- ✓ Prepare the source VMs and the target region
- ✓ Copy the data and enable replication
- ✓ Test the configuration and perform the move
- ✓ Delete the resources in the source region

> **ⓘ Important**
>
> To move Azure VMs to another region, we recommend using Azure Resource Mover. Resource Mover provides:
>
> - A single hub for moving resources across regions.
> - Reduced move time and complexity. Everything you need is in a single location.
> - A simple and consistent experience for moving different types of Azure resources.
> - An easy way to identify dependencies across resources you want to move. This helps you to move related resources together, so that everything works as expected in the target region, after the move.
> - Automatic cleanup of resources in the source region, if you want to delete them after the move.
> - Testing. You can try out a move, and then discard it if you don't want to do a full move.

> **ⓘ Note**
>
> This tutorial shows you how to move Azure VMs from one region to another as is. If you need to improve availability by moving VMs in an availability set to zone pinned VMs in a different region, see the Move Azure VMs into Availability Zones tutorial.

Prerequisites

- Make sure that the Azure VMs are in the Azure region from which you want to move.
- Verify that your choice of source region - target region combination is supported, and make an informed decision about the target region.
- Make sure that you understand the scenario architecture and components.
- Review the support limitations and requirements.
- Verify account permissions. If you created your free Azure account, you're the administrator of your subscription. If you're not the subscription administrator, work with the administrator to assign the permissions that you need. To enable replication for a VM and essentially copy data by using Azure Site Recovery, you must have:
    - Permission to create a VM in Azure resources. The Virtual Machine Contributor built-in role has these permissions, which include:
    - Permission to create a VM in the selected resource group
    - Permission to create a VM in the selected virtual network
    - Permission to write to the selected storage account
    - Permission to manage Azure Site Recovery operations. The Site Recovery Contributor role has all the permissions that are required to manage Site Recovery operations in a Recovery Services vault.
- Make sure that all the latest root certificates are on the Azure VMs that you want to move. If the latest root certificates aren't on the VM, security constraints will prevent the data copy to the target region.
- For Windows VMs, install all the latest Windows updates on the VM, so that all the trusted root certificates are on the machine. In a disconnected environment, follow the standard Windows Update and certificate update processes for your organization.
- For Linux VMs, follow the guidance provided by your Linux distributor to get the latest trusted root certificates and certificate revocation list on the VM.
- Make sure that you're not using an authentication proxy to control network connectivity for VMs that you want to move.
- If the VM that you're trying to move doesn't have access to the internet, or it's using a firewall proxy to control outbound access, check the requirements.

- Identify the source networking layout and all the resources that you're currently using. This includes but isn't limited to load balancers, network security groups (NSGs), and public IPs.
- Verify that your Azure subscription allows you to create VMs in the target region that's used for disaster recovery. Contact support to enable the required quota.
- Make sure that your subscription has enough resources to support VMs with sizes that match your source VMs. If you're using Site Recovery to copy data to the target, Site Recovery chooses the same size or the closest possible size for the target VM.
- Make sure that you create a target resource for every component that's identified in the source networking layout. This step is important to ensure that your VMs have all the functionality and features in the target region that you had in the source region.

  > **Note**
  >
  > Azure Site Recovery automatically discovers and creates a virtual network when you enable replication for the source VM. You can also pre-create a network and assign it to the VM in the user flow to enable replication. As mentioned later, you need to manually create any other resources in the target region.

  To create the most commonly used network resources that are relevant for you based on the source VM configuration, see the following documentation:
  - Network security groups
  - Load balancers
  - Public IP
  - For any other networking components, see the networking documentation.

Prepare

The following steps show how to prepare the virtual machine for the move using Azure Site Recovery as a solution.

Create the vault in any region, except the source region

1. Sign in to the Azure portal.
2. In search, type Recovery Services > click Recovery Services vaults
3. In the Recovery Services vaults menu, click + **Add**.
4. In **Name**, specify the friendly name **ContosoVMVault**. If you have more than one subscription, select the appropriate one.

5. Create the resource group **ContosoRG**.
6. Specify an Azure region. To check supported regions, see geographic availability in Azure Site Recovery pricing details.
7. In **Recovery Services vaults**, select **ContosoVMVault** > **Replicated items** > **+Replicate**.
8. In the dropdown, select **Azure Virtual Machines**.
9. In **Source location**, select the source Azure region where your VMs are currently running.
10. Select the Resource Manager deployment model. Then select the **Source subscription** and **Source resource group**.
11. Select **OK** to save the settings.

Enable replication for Azure VMs and start copying the data

Site Recovery retrieves a list of the VMs that are associated with the subscription and resource group.

1. In the next step, select the VM that you want to move, then select **OK**.
2. In **Settings**, select **Disaster recovery**.
3. In **Configure disaster recovery** > **Target region**, select the target region to which you'll replicate.
4. For this tutorial, accept the other default settings.
5. Select **Enable replication**. This step starts a job to enable replication for the VM.

Move

The following steps show how to perform the move to the target region.

1. Go to the vault. In **Settings** > **Replicated items**, select the VM and then select **Failover**.
2. In **Failover**, select **Latest**.
3. Select **Shut down machine before beginning failover**. Site Recovery attempts to shut down the source VM before triggering the failover. Failover continues even if shutdown fails. You can follow the failover progress on the **Jobs** page.
4. After the job is finished, check that the VM appears in the target Azure region as expected.

Discard

In case you checked the moved VM and need to make changes to the point of failover or want to go back to a previous point, in the **Replicated items**, right-select the VM > **Change recovery point**. This step provides you the option to specify a different recovery point and failover to that one.

Commit

Once you have checked the moved VM and are ready to commit the change, in the **Replicated items**, right-select the VM > **Commit**. This step finishes the move process to the target region. Wait until the commit job finishes.

Clean up

The following steps will guide you through how to clean up the source region as well as related resources that were used for the move.

For all resources that were used for the move:

- Go to the VM. Select **Disable Replication**. This step stops the process from copying the data for the VM.

  > **Important**
  >
  > It's important to perform this step to avoid being charged for Azure Site Recovery replication.

If you have no plans to reuse any of the source resources, complete these additional steps:

1. Delete all the relevant network resources in the source region that you identified in the prerequisites.
2. Delete the corresponding storage account in the source region.

Next steps

In this tutorial, you moved an Azure VM to a different Azure region. Now you can configure disaster recovery for the VM that you moved.

- Learn more about how to Set up disaster recovery after migration.

Last updated on 10/22/2025

Move Azure virtual machines into Availability Zones

Summarize this article for me

This article describes how to move Azure virtual machines to an availability zone in a different region. If you want to move to a different zone in the same region, review this article.

Availability Zones in Azure help protect your applications and data from datacenter failures. Each Availability Zone is made up of one or more datacenters equipped with independent power, cooling, and networking. To ensure resiliency, there's a minimum of three separate zones in all enabled regions. The physical separation of Availability Zones within a region helps protect applications and data from datacenter failures. With Availability Zones, Azure offers a service-level agreement (SLA) of 99.99% for uptime of virtual machines (VMs). Availability Zones are supported in select regions, as mentioned in Regions with availability zones.

In a scenario where you deploy your virtual machines as *single instance* into a specific region, and you want to improve your availability by moving these virtual machines into an Availability Zone, you can use Azure Site Recovery. This action can further be categorized into:

- Move single-instance virtual machines into Availability Zones in a target region
- Move virtual machines in an availability set into Availability Zones in a target region

Important

To move Azure virtual machines to an availability zone in a different region, use Azure Resource Mover. Resource Mover is in public preview and provides:

- A single hub for moving resources across regions.
- Reduced move time and complexity. Everything you need is in a single location.
- A simple and consistent experience for moving different types of Azure resources.
- An easy way to identify dependencies across resources you want to move. This feature helps you move related resources together, so that everything works as expected in the target region after the move.
- Automatic cleanup of resources in the source region, if you want to delete them after the move.
- Testing. You can try out a move, and then discard it if you don't want to do a full move.

Check prerequisites

- Check whether the target region has support for availability zones. Check that your choice of source region/target region combination is supported. Make an informed decision on the target region.
- Make sure that you understand the scenario architecture and components.
- Review the support limitations and requirements.
- Check account permissions. If you just created your free Azure account, you're the admin of your subscription. If you're not the subscription admin, work with the admin to assign the permissions you need. To enable replication for a virtual machine and eventually copy data to the target by using Azure Site Recovery, you must have:

    1. Permission to create a virtual machine in Azure resources. The *Virtual Machine Contributor* built-in role has these permissions, which include:
        - Permission to create a virtual machine in the selected resource group
        - Permission to create a virtual machine in the selected virtual network
        - Permission to write to the selected storage account
    2. Permission to manage Azure Site Recovery tasks. The *Site Recovery Contributor* role has all permissions required to manage Site Recovery actions in a Recovery Services vault.

Prepare the source virtual machines

1. Your virtual machines should use managed disks if you want to move them to an Availability Zone by using Site Recovery. You can convert existing Windows virtual machines that use unmanaged disks to use managed disks. Follow the steps at Convert a Windows virtual machine from unmanaged disks to managed disks. Ensure that the availability set is configured as *managed*.
2. Check that all the latest root certificates are present on the Azure virtual machines you want to move. If the latest root certificates aren't present, security constraints prevent enabling data copy to the target region.
3. For Windows virtual machines, install all the latest Windows updates on the virtual machine, so that all the trusted root certificates are on the machine. In a disconnected environment, follow the standard Windows update and certificate update processes for your organization.

4. For Linux virtual machines, follow the guidance provided by your Linux distributor to get the latest trusted root certificates and certificate revocation list on the virtual machine.

5. Make sure you don't use an authentication proxy to control network connectivity for virtual machines that you want to move.

6. Verify outbound connectivity requirements for virtual machines.

7. Identify the source networking layout and the resources you currently use for verification, including load balancers, NSGs, and public IP.

Prepare the target region

1. Check that your Azure subscription lets you create virtual machines in the target region used for disaster recovery. If necessary, contact support to enable the required quota.

2. Make sure your subscription has enough resources to support virtual machines with sizes that match your source virtual machines. If you use Site Recovery to copy data to the target, it picks the same size or the closest possible size for the target virtual machine.

3. Create a target resource for every component identified in the source networking layout. This action ensures that after you cut over to the target region, your virtual machines have all the functionality and features that you had in the source.

   > **Note**
   >
   > When you enable replication for the source virtual machine, Azure Site Recovery automatically discovers and creates a virtual network and storage account. You can also pre-create these resources and assign them to the virtual machine as part of the enable replication step. For any other resources, you need to manually create them in the target region.

   The following documents explain how to create the most commonly used network resources that are relevant to you, based on the source virtual machine configuration.

   - Network security groups
   - Load balancers
   - Public IP

   For any other networking components, refer to the networking documentation.

   > **Important**

Ensure that you use a zone-redundant load balancer in the target. You can read more at Standard Load Balancer and Availability Zones.

4. Create a non-production network in the target region if you want to test the configuration before you cut over to the target region. This approach causes minimal interference with the production environment.

Enable replication

The following steps guide you through using Azure Site Recovery to enable replication of data to the target region before you move the data into Availability Zones.

> **Note**
>
> These steps are for a single virtual machine. You can extend the same steps to multiple virtual machines. Go to the Recovery Services vault, select + **Replicate**, and select the relevant virtual machines together.

1. In the Azure portal, select **Virtual machines**, and select the virtual machine you want to move into Availability Zones.
2. In **Backup + disaster recovery**, select **Disaster recovery**.
3. In **Configure disaster recovery** > **Target region**, select the target region to which you'll replicate. Ensure this region supports availability zones.
4. Select **Next: Advanced settings**.
5. Choose the appropriate values for the target subscription, target virtual machine resource group, and virtual network.
6. In the **Availability** section, choose the Availability Zone into which you want to move the virtual machine.

   > **Note**
   >
   > If you don't see the option for availability set or Availability Zone, ensure that the prerequisites are met and the preparation of source virtual machines is complete.

7. Select **Enable Replication**. This action starts a job to enable replication for the virtual machine.

Check settings

After the replication job finishes, you can check the replication status, modify replication settings, and test the deployment.

1. In the virtual machine menu, select **Disaster recovery**.
2. You can check replication health, the recovery points that are created, and the source and target regions on the map.

Test the configuration

1. In the virtual machine menu, select **Disaster recovery**.
2. Select the **Test Failover** icon.
3. In **Test Failover**, select a recovery point to use for the failover:
   - **Latest processed**: Fails the virtual machine over to the latest recovery point that the Site Recovery service processes. The time stamp is shown. By using this option, the service spends no time processing data, so it provides a low recovery time objective (RTO).
   - **Latest app-consistent**: This option fails over all virtual machines to the latest app-consistent recovery point. The time stamp is shown.
   - **Custom**: Select any recovery point.
4. Select the test target Azure virtual network to which you want to move the Azure virtual machines to test the configuration.

   > **ⓘ Important**
   >
   > Use a separate Azure virtual machine network for the test failure, and not the production network in the target region into which you want to move your virtual machines.

5. Select **OK** to start testing the move. To track progress, select the virtual machine to open its properties. Or, you can select the **Test Failover** job in the vault name > **Settings** > **Jobs** > **Site Recovery jobs**.
6. After the failover finishes, the replica Azure virtual machine appears in the Azure portal > **Virtual Machines**. Make sure that the virtual machine is running, sized appropriately, and connected to the appropriate network.

7. If you want to delete the virtual machine created as part of testing the move, select **Cleanup test failover** on the replicated item. In **Notes**, record and save any observations associated with the test.

Move to the target region and confirm

1. In the virtual machine menu, select **Disaster recovery**.
2. Select the **Failover** icon.
3. In **Failover**, select **Latest**.
4. Select **Shut down machine before beginning failover**. Site Recovery attempts to shut down the source virtual machine before triggering the failover. Failover continues even if shutdown fails. You can follow the failover progress on the **Jobs** page.
5. After the job finishes, check that the virtual machine appears in the target Azure region as expected.
6. In **Replicated items**, right-click the virtual machine > **Commit**. This action finishes the move process to the target region. Wait until the commit job finishes.

Discard the resource in the source region

Go to the virtual machine. Select **Disable Replication**. This action stops the process of copying the data for the virtual machine.

> **Important**
>
> To avoid charges for Site Recovery replication after the move, perform the preceding step. The source replication settings are cleaned up automatically. The Site Recovery extension that you install as part of the replication isn't removed and needs to be removed manually.

Next steps

In this tutorial, you increased the availability of an Azure virtual machine by moving into an availability set or Availability Zone. Now you can set disaster recovery set up disaster recovery after migration for the moved virtual machine.

Last updated on 02/12/2026

Move a Maintenance Control configuration to another region

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Follow this article to move a Maintenance Control configuration to a different Azure region. You might want to move a configuration for a number of reasons. For example, to take advantage of a new region, to deploy features or services available in a specific region, to meet internal policy and governance requirements, or in response to capacity planning.

Maintenance control, with customized maintenance configurations, allows you to control how platform updates are applied to VMs, and to Azure Dedicated Hosts. There are a couple of scenarios for moving maintenance control across regions:

- To move your maintenance control configuration, but not the resources associated with the configuration, follow the instructions in this article.
- To move the resources associated with a maintenance configuration, but not the configuration itself, follow these instructions.
- To move both the maintenance configuration and the resources associated with it, first follow the instructions in this article. Then, follow these instructions.

Prerequisites

Before you begin moving a maintenance control configuration:

- Maintenance configurations are associated with Azure VMs or Azure Dedicated Hosts. Make sure that VM/host resources exist in the new region before you begin.
- Identify:
  - Existing maintenance control configurations.
  - The resource groups in which existing configurations currently reside.
  - The resource groups to which the configurations will be added after moving to the new region.
  - The resources associated with the maintenance configuration you want to move.
  - Check that the resources in the new region are the same as those associated with the current maintenance configurations. The configurations can have the same names in the new region as they did in the old, but this isn't required.

Prepare and move

1. Retrieve all of the maintenance configurations in each subscription. Run the CLI az maintenance configuration list command to do this, replacing $subId with your subscription ID.

```
az maintenance configuration list --subscription $subId --query "[*].
{Name:name, Location:location, ResGroup:resourceGroup}" --output table
```

2. Review the returned table list of configuration records within the subscription. Here's an example. Your list will contain values for your specific environment.

Expand table

Name	Location	Resource group
Skip Maintenance	eastus2	configuration-resource-group
IgniteDemoConfig	eastus2	configuration-resource-group
defaultMaintenanceConfiguration-eastus	eastus	test-configuration

3. Save your list for reference. As you move the configurations, it helps you to verify that everything's been moved.

4. As a reference, map each configuration/resource group to the new resource group in the new region.

5. Create new maintenance configurations in the new region using PowerShell, or CLI.

6. Associate the configurations with the resources in the new region, using PowerShell, or CLI.

Verify the move

After moving the configurations, compare configurations and resources in the new region with the table list you prepared.

Clean up source resources

After the move, consider deleting the moved maintenance configurations in the source region, PowerShell, or CLI.

Next steps

Follow these instructions if you need to move resources associated with maintenance configurations.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Move resources in a Maintenance Control configuration to another region

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

Follow this article to move resources associated with a Maintenance Control configuration to a different Azure region. You might want to move a configuration for a number of reasons. For example, to take advantage of a new region, to deploy features or services available in a specific region, to meet internal policy and governance requirements, or in response to capacity planning.

Maintenance control, with customized maintenance configurations, allows you to control how platform updates are applied to VMs, and to Azure Dedicated Hosts. There are a couple of scenarios for moving maintenance control across regions:

- To move the resources associated with a maintenance configuration, but not the configuration itself, follow this article.
- To move your maintenance control configuration, but not the resources associated with the configuration, follow these instructions.
- To move both the maintenance configuration and the resources associated with it, first follow these instructions. Then, follow the instructions in this article.

Prerequisites

Before you begin moving the resources associated with a Maintenance Control configuration:

- Make sure that the resources you're moving exist in the new region before you begin.
- Verify the Maintenance Control configurations associated with the Azure VMs and Azure Dedicated Hosts that you want to move. Check each resource individually. There's currently no way to retrieve configurations for multiple resources.
- When retrieving configurations for a resource:
  - Make sure you use the subscription ID for the account, not an Azure Dedicated Host ID.
  - CLI: The --output table parameter is used for readability only, and can be deleted or changed.

- PowerShell: The Format-Table Name parameter is used for readability only, and can be deleted or changed.
- If you use PowerShell, you get an error if you try to list configurations for a resource that doesn't have any associated configurations. The error will be similar to: "Operation failed with status: 'Not Found'. Details: 404 Client Error: Not Found for url".

Prepare to move

1. Before you start, define these variables. We've provided an example for each.

Expand table

Variable	Details	Example
$subId	ID for subscription containing the maintenance configurations	"our-subscription-ID"
$rsrcGroupName	Resource group name (Azure VM)	"VMResourceGroup"
$vmName	VM resource name	"myVM"
$adhRsrcGroupName	Resource group (Dedicated hosts)	"HostResourceGroup"
$adh	Dedicated host name	"myHost"
$adhParentName	Parent resource name	"HostGroup"

2. To retrieve the maintenance configurations using the PowerShell Get-AZConfigurationAssignment command:

   - For Azure Dedicated Hosts, run:

```
Get-AzConfigurationAssignment -ResourceGroupName $adhRsrcGroupName
-ResourceName $adh -ResourceType hosts -ProviderName
Microsoft.Compute -ResourceParentName $adhParentName -
ResourceParentType hostGroups | Format-Table Name
```

   - For Azure VMs, run:

```
Get-AzConfigurationAssignment -ResourceGroupName $rgName -
ResourceName $vmName -ProviderName Microsoft.Compute -ResourceType
```

```
virtualMachines | Format-Table Name
```

3. To retrieve the maintenance configurations using the CLI az maintenance assignment command:

   - For Azure Dedicated Hosts:

     ```
     az maintenance assignment list --subscription $subId --resource-group $adhRsrcGroupName --resource-name $adh --resource-type hosts --provider-name Microsoft.Compute --resource-parent-name $adhParentName --resource-parent-type hostGroups --query "[].{HostResourceGroup:resourceGroup,ConfigName:name}" --output table
     ```

   - For Azure VMs:

     ```
     az maintenance assignment list --subscription $subId --provider-name Microsoft.Compute --resource-group $rsrcGroupName --resource-name $vmName --resource-type virtualMachines --query "[].{HostResourceGroup:resourceGroup, ConfigName:name}" --output table
     ```

Move

1. Follow these instructions to move the Azure VMs to the new region.
2. After the resources are moved, reapply maintenance configurations to the resources in the new region as appropriate, depending on whether you moved the maintenance configurations. You can apply a maintenance configuration to a resource using PowerShell or CLI.

Verify the move

Verify resources in the new region, and verify associated configurations for the resources in the new region.

Clean up source resources

After the move, consider deleting the moved resources in the source region.

Next steps

Follow these instructions if you need to move maintenance configurations.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate from Amazon EC2 to Azure Virtual Machines

Summarize this article for me

If you use Amazon Elastic Compute Cloud (Amazon EC2) and plan to migrate your workload to Azure, this guide can help you understand the migration process, feature mappings, and best practices. It's for Amazon Web Services (AWS) professionals who are familiar with Amazon EC2 and plan to move workloads to Azure Virtual Machines. The guide highlights key similarities and differences between the platforms and outlines important architectural considerations. It also provides best practices for performance, cost, and availability to help you plan and complete a successful migration to an Azure infrastructure as a service (IaaS) environment.

What you will accomplish

This guide describes how to do the following tasks:

- Map Amazon EC2 instance families and sizes to the right Azure virtual machine (VM) series and SKUs.
- Translate Amazon machine image (AMI)-based workloads to Microsoft Marketplace or custom images.
- Design Azure storage architectures that meet or exceed existing Amazon Elastic Block Store (Amazon EBS) performance characteristics.
- Recreate Amazon Virtual Private Cloud (Amazon VPC) networking, security, and load-balancing patterns by using Azure-native services.
- Learn about availability, scaling, and placement strategies in Azure that align with AWS designs.
- Build the prerequisite knowledge and skills for a successful migration.
- Validate performance, resiliency, and functionality after migration.

Example scenario: Migrate a production Amazon EC2-based application stack

An organization runs a production web application on Amazon EC2 that uses the following components:

- General-purpose Amazon EC2 instances in multiple availability zones
- Amazon Elastic Load Balancing (Amazon ELB)
- Auto Scaling groups (ASGs) for elasticity
- Amazon EBS volumes for persistent storage

- Custom AMIs as a golden image baseline

The goal is to migrate this workload to Virtual Machines while you maintain availability, performance, and scaling characteristics.

Use Azure Migrate to migrate your Amazon EC2 instances to Azure

Azure Migrate provides a unified platform to assess and migrate on-premises servers, infrastructure, applications, and data. You can use Azure Migrate to discover, assess, and migrate Amazon EC2 instances to Azure.

> **Tip**
>
> Use Azure Migrate when your migration effort grows beyond a few manual builds and requires a repeatable, scalable approach.
>
> Use these guidelines to determine when Azure Migrate suits your VM migration:
>
> - For VMs that use the same operating system, have similar sizes, and have simple dependencies, use Azure Migrate when you migrate five or more VMs.
> - For VMs that use different operating systems or sizes, have multiple disks, or have complex dependencies, use Azure Migrate when you migrate three or more VMs.

For more information, see Migrate Amazon EC2 instances by using Azure Migrate.

Architectural overview

In AWS, the workload uses Amazon EC2 instances distributed across availability zones inside an Amazon VPC, fronted by an Amazon ELB and scaled by using ASGs. Amazon EBS provides storage, and custom AMIs provide images.

In Azure, the following components form the equivalent architecture:

- Virtual Machines or Azure Virtual Machine Scale Sets
- Azure Virtual Network with subnets
- Azure Load Balancer or Azure Application Gateway
- Azure Managed Disks
- Marketplace images or custom images stored in Azure Compute Gallery

Production environment considerations

When you transition from Amazon EC2 to Virtual Machines, plan for the following considerations:

- Differences in VM sizing, disk performance coupling, and networking limits
- Image format and agent compatibility because you can't import AMIs directly
- Availability constructs, like availability zones versus availability sets
- Network security rule evaluation and service integration
- Differences in monitoring, logging, and automation

Step 1: Plan

Inventory the existing Amazon EC2 workload and capture how it behaves in production.

The key assessment activities include the following items:

- Identify Amazon EC2 instance families, sizes, and CPU architecture, like `x86` or `ARM64`.
- Capture baseline and peak CPU, memory, disk input/output operations per second (IOPS), throughput, latency, and network usage.
- Document Amazon EBS volume types and performance settings.
- Review ASG policies and placement strategies.
- Enumerate security group and network access control list (NACL) rules.
- Identify AMI sources, like public, vendor, or custom images.

Formalize your findings by categorizing each capability into one of the following groups:

- Direct matches in Azure
- Matches that have configuration differences
- Capabilities that require alternative designs

Direct capability mapping

Expand table

Amazon EC2 capability	Azure Virtual Machines equivalent	Migration approach
Amazon EC2 instance families like `t`, `m`, `c`, `r`, `i`, and `p`	Azure VM series like B, D, F, E, L, and NC, ND, or NP	Select Azure VM SKUs that have equivalent CPU-to-memory ratios and architecture.

Amazon EC2 capability	Azure Virtual Machines equivalent	Migration approach
ASGs	Virtual Machine Scale Sets	Set up autoscaling in Virtual Machine Scale Sets and distribute instances across zones.
Amazon ELB: Amazon Application Load Balancer (ALB) and Amazon Network Load Balancer (NLB)	Load Balancer and Application Gateway	Map layer-4 or layer-7 behavior and health probes.
Amazon EBS volumes	Azure Managed Disks	Map Amazon EBS volume types to the right disk SKUs and validate limits.
Availability zones	Azure availability zones	Deploy VMs or Virtual Machine Scale Sets instances across zones where supported.

Capability mismatches and alternative strategies

Some Amazon EC2 capabilities don't map directly to Azure:

- **AMI portability:** Azure doesn't support a lift-and-shift approach for AMIs. You must map images to Marketplace images or rebuild them as custom images.
- **Bursting behavior:** AWS supports CPU bursts beyond the `t` family. Azure limits CPU bursts to the B-series.
- **Disk performance scale:** AWS decouples disk performance from instance size. In Azure, both disk SKU and VM size constrain disk performance.
- **Hypervisor access:** Amazon EC2 bare-metal instances expose more control than Azure VMs. For hardware isolation requirements in Azure, use Azure Dedicated Host.

Step 2: Prepare

Prepare the Azure environment before you migrate workloads:

- Design Azure virtual networks and subnets that align with existing Amazon VPC segmentation.
- Establish identity and access control by using Azure role-based access control (Azure RBAC).
- Choose availability zones or availability sets based on your workload requirements and regional support.
- Select base Marketplace images that match the operating system version, architecture, and boot mode.

- Remove AWS-specific agents and ensure Azure VM Agent support.
- Use Azure Migrate for discovery, dependency map creation, and rightsizing recommendations when you migrate multiple VMs.

Step 3: Execute

Each stage of the migration process requires careful planning and execution to ensure a successful transition from Amazon EC2 to Virtual Machines. This section describes how to migrate compute, images, storage, networking, and availability features.

Compute

When you migrate from Amazon EC2 to Virtual Machines, you must know how instance families map to Azure VM series to plan your workloads effectively. Both platforms organize their compute offerings by performance characteristics, but their naming conventions and configuration options differ.

Amazon EC2 instance families

AWS organizes compute resources into instance families based on workload type:

- **General purpose:** Balanced CPU, memory, and networking, like `t3` and `m5`.
- **Compute optimized:** High CPU-to-memory ratio for compute-intensive tasks, like `c5` and `c6g`.
- **Memory optimized:** Large memory footprint for in-memory databases or analytics, like `r5` and `x1e`.
- **Storage optimized:** High disk throughput for large datasets, like `i3` and `d2`.
- **Accelerated computing:** GPU-based workloads for machine learning or graphics, like `p3` and `g4dn`.

AWS instances scale vertically by selecting larger sizes within a family, like moving from `m5.large` to `m5.4xlarge`, and horizontally by using ASGs.

Azure VM series

Azure uses VM series to categorize compute resources:

- **D-series:** General-purpose workloads, similar to the AWS `m` family
- **F-series:** Compute optimized, comparable to the AWS `c` family
- **E-series:** Memory optimized, similar to the AWS `r` family

- **M-series:** Ultra-high memory for SAP HANA and large databases
- **L-series:** Storage optimized with local Non-Volatile Memory Express (NVMe) disks, and comparable to the AWS `i` family
- **NC, ND, and NP-series:** GPU enabled for AI and machine learning workloads, similar to the AWS `p` and `g` families

Azure defines VM sizes by a SKU, like `Standard_D4s_v5`. The SKU specifies the following properties:

- Virtual CPU (vCPU) count
- Memory, expressed in gibibytes (GiB)
- Temporary storage
- Generation

For more information, see VM size naming conventions.

Key differences

- **Naming:** AWS uses family and size, like `c5.xlarge`. Azure uses VM series and VM SKU, like `Standard_F4s_v2`.
- **Performance tiers:** Azure sets disk performance according to VM size and disk SKU. AWS uses Amazon EBS-optimized instances.
- **Regional availability:** Features vary by region on both platforms. On Azure, features like availability zones and spot capacity are available only in specific regions.
- **Burstable options:** The AWS `t` family supports burstable workloads, and AWS provides burst capability on other eligible instance sizes. Azure limits bursting to the B-series.
- **Hypervisor access:** Some AWS sizes allow for more direct control over the hypervisor, like `i3.metal`. Azure provides less control at this layer.

Mapping strategy

Use the Azure VM size documentation and AWS instance type guide to complete the following tasks:

1. Identify your Amazon EC2 instance family and size.
2. Match to an Azure VM series with equivalent CPU-to-memory ratio and the required CPU architecture, like `x86` or `ARM`.
3. Validate storage and network requirements to prevent overprovisioning or underprovisioning:

    - **Establish a baseline in AWS.** Capture typical and peak Amazon EBS IOPS, throughput, and latency, and capture network bandwidth and packets-per-second

(PPS) usage.
    - **Map to Azure limits.** Confirm disk SKU and VM size caps and the VM's network limits, including whether it supports Azure accelerated networking.
    - **Test in Azure.** Run quick storage and network benchmarks before you finalize the VM size.
4. If you use accelerators, like GPU or Field-Programmable Gate Array (FPGA), ensure that the Azure VM series supports them.
5. Consider Azure-specific features like spot VMs for cost savings or virtual machine scale sets for elasticity.

Images

When you migrate workloads that start from an AMI, plan for an *image translation* step.

> **Important**
>
> Azure doesn't support lift and shift of AMIs. You can't import an AMI and run it unchanged on Azure.
>
> Map an AMI to a Marketplace image (catalog to catalog) or map a custom AMI to a custom Azure image (custom to custom) instead. Then validate drivers, agents, and generation compatibility.

Find a matching (catalog) image

Start by identifying what the AMI represents:

- Operating system distribution and version, like Ubuntu 22.04, RHEL 8.9, and Windows Server 2022
- CPU architecture, like `x86_64` versus `ARM64`
- Boot mode and VM generation assumptions, like UEFI with Gen2 VMs or BIOS with Gen1 VMs
- Installed components, like monitoring agents, security tools, and web and app runtimes
- Licensing model, like bring-your-own versus marketplace-provided

Then find an equivalent Azure image:

- Marketplace images best match to public AMIs.
- Images that your organization publishes through Compute Gallery are most similar to private or shared AMIs.

If your AWS workload depends on a vendor AMI, like a firewall, appliance, or hardened image, identify the vendor's equivalent offering in Marketplace and confirm that it meets the following requirements:

- Supported VM sizes and required networking features
- Required disk layout and performance
- Licensing and support terms

Roll your own (custom-to-custom)

If the AMI is a custom image, like a golden image, a hardened baseline, or an image with a preinstalled app, the Azure equivalent is a custom image that you store and version in Compute Gallery.

Use the following steps as a recommended starting point:

1. **Pick a clean base image** from Marketplace that matches the operating system, version, and architecture that you need.
2. **Automate customization** by applying packages, configuration, agents, and hardening through a repeatable pipeline.
3. **Validate the image** when you deploy test VMs and run smoke tests.
4. **Publish and version the image** in Compute Gallery and replicate it to target regions.

Consider the following common tooling patterns:

- **Azure VM Image Builder** is a managed image build service for image creation and distribution.
- **HashiCorp Packer**, including the Azure builders, is for building images in continuous integration and continuous delivery (CI/CD) pipelines.
- **Configuration management tools** like Ansible, Chef, or Puppet help keep customization reproducible.

Consider these operational requirements to plan your *where do you start* checklist:

- **Generalization:**
  - On Windows, run Sysprep before you capture the image.
  - On Linux, install the Azure Linux VM Agent (`waagent`) before you capture the image.
- **Drivers and agents:** Ensure that the image supports the Azure VM Agent and remove any AWS-specific agents or tools that no longer apply.
- **VM generation:** Choose Gen1 or Gen2 early because your base image choice typically determines the generation.
- **Identity and secrets:** Use managed identity and Azure Key Vault instead of embedding secrets in images.

- **Updates:** Patch the image during build and define an image refresh cadence.

Storage

Storage architecture is a critical factor when you migrate from Amazon EC2 to Virtual Machines. Both platforms provide persistent and ephemeral storage options, but implementation and performance models differ.

Amazon EC2 storage options

- **Amazon EBS:** Persistent block storage for Amazon EC2 instances. Supports solid-state drive (SSD) and hard disk drive (HDD) volumes:
  - General-purpose SSD that uses `gp3` and `gp2`
  - Provisioned IOPS SSD that uses `io1` and `io2`
  - Throughput-optimized HDD that uses `st1`
  - Cold HDD that uses `sc1`
- **Network-attached storage (NAS):** Shared file storage for Linux and Windows workloads:
  - **Amazon Elastic File System (Amazon EFS)**
  - **Amazon FSx**, like Windows File Server, NetApp ONTAP, and Lustre
- **Instance store:** Ephemeral storage that's physically attached to the host. Data is deleted when the instance stops or terminates.
- **Amazon Simple Storage Service (Amazon S3):** Object storage for unstructured data, backups, and archival storage.

Key features of Amazon EC2 storage options include the following items:

- Snapshots for Amazon EBS volumes are stored in Amazon S3.
- Performance depends on volume type and size.
- Encryption via AWS Key Management Service (AWS KMS).

Azure VM storage options

- **Managed disks:** Azure manages persistent block storage:
  - **Azure Standard HDD:** Cost effective for infrequent access, nonproduction workloads, and long-term backups.
  - **Azure Standard SSD:** Balanced performance for general workloads.
  - **Azure Premium SSD:** Low latency for production and performance-sensitive apps.
  - **Azure Ultra Disk Storage:** High throughput for data-intensive workloads.
- **Ephemeral OS disks:** Temporary storage for stateless workloads.
- **Externalized storage (NAS and object storage):** Network-attached and object-based storage for shared file access, backups, archival, and large-scale data:

- Azure Blob Storage
- Azure Files
- Azure NetApp Files

Key features of Azure VM storage options include the following items:

- Disk performance tiers depend on VM size and disk SKU.
- Azure VM storage includes built-in snapshot capabilities and integrates with Azure Backup.
- Encryption at rest uses default keys and supports customer-managed keys.

Architectural differences

Expand table

Feature	Amazon EC2 (Amazon EBS)	Azure Virtual Machines (managed disks)
Performance scaling	Based on volume type and size	Based on VM size and disk SKU
Snapshot integration	Stored in Amazon S3	Built-in, integrates with Azure Backup
Encryption	AWS KMS	Azure disk encryption and Key Vault
Resiliency	Availability zone-level replication optional	Zone-redundant storage (ZRS) available

For NAS, Amazon EFS and Amazon FSx map most directly to Azure Files and Azure NetApp Files.

Storage migration considerations

- Map Amazon EBS volumes to Azure Managed Disk tiers:
  - The `gp2` and `gp3` volume types map to Standard SSD for light or moderate use.
  - The `gp2` volume type maps to Premium SSD.
  - The `gp3` volume type maps to Premium SSD v2.
  - The `io2` volume type maps to Ultra Disk Storage.
- Validate IOPS and throughput requirements. Premium SSD and Ultra Disk Storage support high-performance workloads.
- Plan for encryption compliance. Use Azure disk encryption and Key Vault for sensitive data.
- For externalized storage migration, you can use the following approaches:

    - Migrate Amazon S3 to Blob Storage by using AzCopy or Azure Storage migration tools.
    - Migrate Amazon EFS and Amazon FSx to Azure Files for general-purpose file shares, or to Azure NetApp Files for high-performance NAS.

Validate IOPS and throughput requirements carefully because both disk SKU and VM size constrain Azure disk performance.

Networking

Networking architecture is a critical component when you migrate from Amazon EC2 to Virtual Machines. Both platforms provide secure, isolated networks, but terminology, configuration, and feature sets differ.

Amazon EC2 networking

- **Amazon VPC:** Logical isolation of resources within AWS.
- **Subnets:** Network segments within an Amazon VPC for isolation and segmentation.
- **Security groups:** Stateful firewall rules applied at the instance level.
- **Network ACLs:** Stateless rules applied at the subnet level.
- **Elastic IP addresses:** Static public IP addresses for instances.
- **Load balancing:** Amazon ELB supports layer-4 and layer-7 traffic.
- **Hybrid connectivity:** Virtual private network (VPN) and AWS Direct Connect for private links to on-premises.

Azure VM networking components

- **Virtual network:** An isolated and segmented network environment equivalent to an Amazon VPC.
- **Subnets:** Network segments within a virtual network that provide isolation and support network security groups (NSGs) for traffic filtering.
- **NSGs:** Stateful inbound and outbound traffic rules applied at the subnet or network interface card (NIC) level.
- **Azure Firewall:** A managed, cloud-based firewall used for centralized policy enforcement.
- **Azure Private Link:** Private IP address-based access to Azure services.
- **Public IP addresses:** Static or dynamic public IP addresses that you assign to resources.
- **Load balancing:** Load Balancer at layer 4 and Application Gateway at layer 7 with Secure Sockets Layer (SSL) termination and Azure Web Application Firewall.
- **Azure Bastion:** Secure Remote Desktop Protocol (RDP) and Secure Shell (SSH) access without exposing public IP addresses.
- **Azure ExpressRoute:** Private dedicated connectivity to Azure for hybrid scenarios.

Key differences

Expand table

Feature	Amazon EC2 (Amazon VPC)	Azure Virtual Machines (virtual network)
Firewall integration	Security groups and NACLs	NSGs and Azure Firewall
Private service access	Amazon VPC endpoints	Private Link
Hybrid connectivity	VPN and AWS Direct Connect	Azure VPN Gateway and ExpressRoute
Secure remote access	Jump hosts	Azure Bastion

Migration considerations

1. **Plan virtual network architecture:** Align with existing Amazon VPC design for subnet segmentation.
2. **Security rules:** Convert AWS security group rules to NSGs. Review inbound and outbound traffic.
3. **Hybrid connectivity:** Replace AWS Direct Connect with ExpressRoute for private connectivity.
4. **Load balancing:** Map Amazon ELB configurations to Load Balancer or Application Gateway.
5. **Access control:** Use Azure Bastion for secure remote access instead of exposing public IP addresses.

Clustering, availability, and zones

High-availability and resiliency strategies vary between Amazon EC2 and Virtual Machines. These concepts are essential for a fault-tolerant architecture.

Amazon EC2 availability features

- **Availability zones:** Isolated locations within a region for redundancy.
- **ASGs:** Automatic scaling of instance counts based on demand.
- **Amazon ELB:** Traffic distribution across multiple instances.
- **Placement groups:** Instance placement for low-latency or high-throughput workloads.

Azure VM availability features

- **Availability zones:** Physically separate datacenters within a region for zone-level resiliency.
- **Availability sets:** Logical grouping to protect against rack-level failures.
- **Virtual Machine Scale Sets:** Built-in orchestration for horizontal workload scale.
- **Load Balancer and Application Gateway:** Layer-4 and layer-7 traffic distribution.

Mapping translation

Expand table

Feature	Amazon EC2	Azure Virtual Machines	Migration considerations
Zone redundancy	Availability zone-based deployment	Availability zones and ZRS	Map AWS availability zone deployment to Azure availability zones. When you use Virtual Machine Scale Sets, distribute instances across zones for maximum resiliency.
Rack-level protection	Not explicit	Availability sets	Use availability sets for rack-level resiliency.
Automatic scaling	ASGs	Virtual Machine Scale Sets	Map AWS ASGs to Virtual Machine Scale Sets. Consider zone-distributed Virtual Machine Scale Sets where supported.
Load balancing	Amazon ELB that provides layer-4 and layer-7 capabilities	Load Balancer and Application Gateway	Set up load balancing. Replace Amazon ELB with Load Balancer or Application Gateway.

Best practices

- Deploy across multiple zones for disaster recovery.
- Use Virtual Machine Scale Sets with autoscaling policies for elasticity.
- Use ZRS for critical data.
- Integrate Azure Monitor for health checks and alerting.

Scaling and placement

AWS and Azure use different constructs for scaling and placement. These differences affect elasticity and fault isolation during migration.

AWS approach

- **ASGs:** Automatically adjust Amazon EC2 instance count based on demand by using launch templates, which define instance configuration and life cycle, including NICs and disks.
- **Partition placement groups:** Spread Amazon EC2 instances across multiple partitions for resiliency and low-latency workloads.

Azure approach

- **Virtual Machine Scale Sets:** Provides native orchestration for horizontal scaling, integrated with Load Balancer or Application Gateway.
- **VM profiles:** Defines VM configuration and support deep deletion for resource cleanup.
- **Fault domains and availability sets:** Provides rack-level isolation similar to AWS partitioning. For dedicated hardware, use Dedicated Host.

Key differences

Expand table

Feature	Amazon EC2	Azure Virtual Machines
Scaling construct	ASGs	Virtual Machine Scale Sets and autoscaling
Instance configuration	Launch template	VM configuration profile and deep deletion
Placement strategy	Partition placement groups	Fault domains, availability sets, and Dedicated Host

Step 4: Evaluate

After the migration completes, do the following validation steps:

- Validate operating system boot and agent health.
- Confirm that disk throughput, latency, and network performance meet expectations.
- Test scaling behavior and load balancing.
- Validate application functionality and dependencies.
- Set up Azure Monitor alerts and logging.

Successful validation indicates that the workload is ready for production cutover.

Related content

- Virtual machines overview
- Azure for AWS professionals

- Compare AWS and Azure compute services
- Migrate a workload from AWS to Azure

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 02/25/2026

Discover, assess, and migrate Amazon Web Services (AWS) EC2 instances to Azure

This tutorial shows you how to discover, assess, and migrate Amazon Web Services (AWS) EC2 instances to Azure VMs by using Azure Migrate: Server Assessment and the Migration and modernization tool.

> **Note**
>
> You migrate AWS VMs to Azure by treating them as physical servers.

In this tutorial, you learn how to:

- Verify prerequisites for migration.
- Prepare Azure resources with the Migration and modernization tool. Set up permissions for your Azure account and resources to work with Azure Migrate and Modernize.
- Prepare AWS Elastic Compute Cloud (EC2) instances for migration.
- Add the Migration and modernization tool in the Azure Migrate and Modernize hub.
- Set up the replication appliance and deploy the configuration server.
- Install the Mobility service on the AWS VMs you want to migrate.
- Enable replication for VMs.
- Track and monitor the replication status.
- Run a test migration to make sure everything's working as expected.
- Run a full migration to Azure.

If you don't have an Azure subscription, create a free account before you begin.

Discover and assess

Before you migrate to Azure, we recommend that you perform a VM discovery and migration assessment. This assessment helps right-size your AWS VMs for migration to Azure and estimate potential Azure run costs.

To set up an assessment:

1. Follow the tutorial to set up Azure and prepare your AWS VMs for an assessment. Note that:

- Azure Migrate and Modernize uses password authentication to discover AWS instances. AWS instances don't support password authentication by default. Before you can discover an instance, you need to enable password authentication.
  - For Windows machines, allow WinRM port 5985 (HTTP). This port allows remote WMI calls.
  - For Linux machines:
    a. Sign in to each Linux machine.
    b. Open the *sshd_config* file: `vi /etc/ssh/sshd_config`.
    c. In the file, locate the `PasswordAuthentication` line and change the value to `yes`.
    d. Save the file and close it. Restart the ssh service.
- If you're using a root user to discover your Linux VMs, ensure that root login is allowed on the VMs.
  a. Sign in to each Linux machine.
  b. Open the *sshd_config* file: `vi /etc/ssh/sshd_config`.
  c. In the file, locate the `PermitRootLogin` line and change the value to `yes`.
  d. Save the file and close it. Restart the ssh service.

2. Then, follow this tutorial to set up an Azure Migrate project and appliance to discover and assess your AWS VMs.

Although we recommend that you try out an assessment, performing an assessment isn't a mandatory step to be able to migrate VMs.

Prerequisites

- Ensure that the AWS VMs you want to migrate are running a supported operating system (OS) version. AWS VMs are treated like physical machines for the migration. Review the supported operating systems and kernel versions for the physical server migration workflow. You can use standard commands like `hostnamectl` or `uname -a` to check the OS and kernel versions for your Linux VMs. We recommend that you perform a test migration (test failover) to validate if the VM works as expected before you proceed with the migration.
- Make sure your AWS VMs comply with the supported configurations for migration to Azure.
- Verify that the AWS VMs that you replicate to Azure comply with Azure VM requirements.
- Some changes are needed on the VMs before you migrate them to Azure:

- For some operating systems, Azure Migrate and Modernize makes these changes automatically.
- It's important to make these changes before you begin migration. If you migrate the VM before you make the change, the VM might not boot up in Azure. Review the Windows and Linux changes you need to make.

Prepare Azure resources for migration

Prepare Azure for migration with the Migration and modernization tool.

Expand table

Task	Details
Create an Azure Migrate project	Your Azure account needs Contributor or Owner permissions to create a new project.
Verify permissions for your Azure account	Your Azure account needs permissions to create a VM and write to an Azure managed disk.

Assign permissions to create a project

1. In the Azure portal, open the subscription and select **Access control (IAM)**.
2. In **Check access**, find the relevant account and select it to view permissions.
3. You should have **Contributor** or **Owner** permissions.

    - If you just created a free Azure account, you're the owner of your subscription.
    - If you're not the subscription owner, work with the owner to assign the role.

Assign Azure account permissions

Assign the VM Contributor role to the Azure account. This role provides permissions to:

- Create a VM in the selected resource group.
- Create a VM in the selected virtual network.
- Write to an Azure managed disk.

Create an Azure network

Set up an Azure virtual network. When you replicate to Azure, the Azure VMs that are created are joined to the Azure virtual network that you specified when you set up migration.

Prepare AWS instances for migration

To prepare for AWS to Azure migration, you need to prepare and deploy a replication appliance for migration.

Prepare a machine for the replication appliance

The Migration and modernization tool uses a replication appliance to replicate machines to Azure. The replication appliance runs the following components:

- **Configuration server**: The configuration server coordinates communications between the AWS environment and Azure and manages data replication.
- **Process server**: The process server acts as a replication gateway. It receives replication data and optimizes that data with caching, compression, and encryption. Then it sends the data to a cache storage account in Azure.

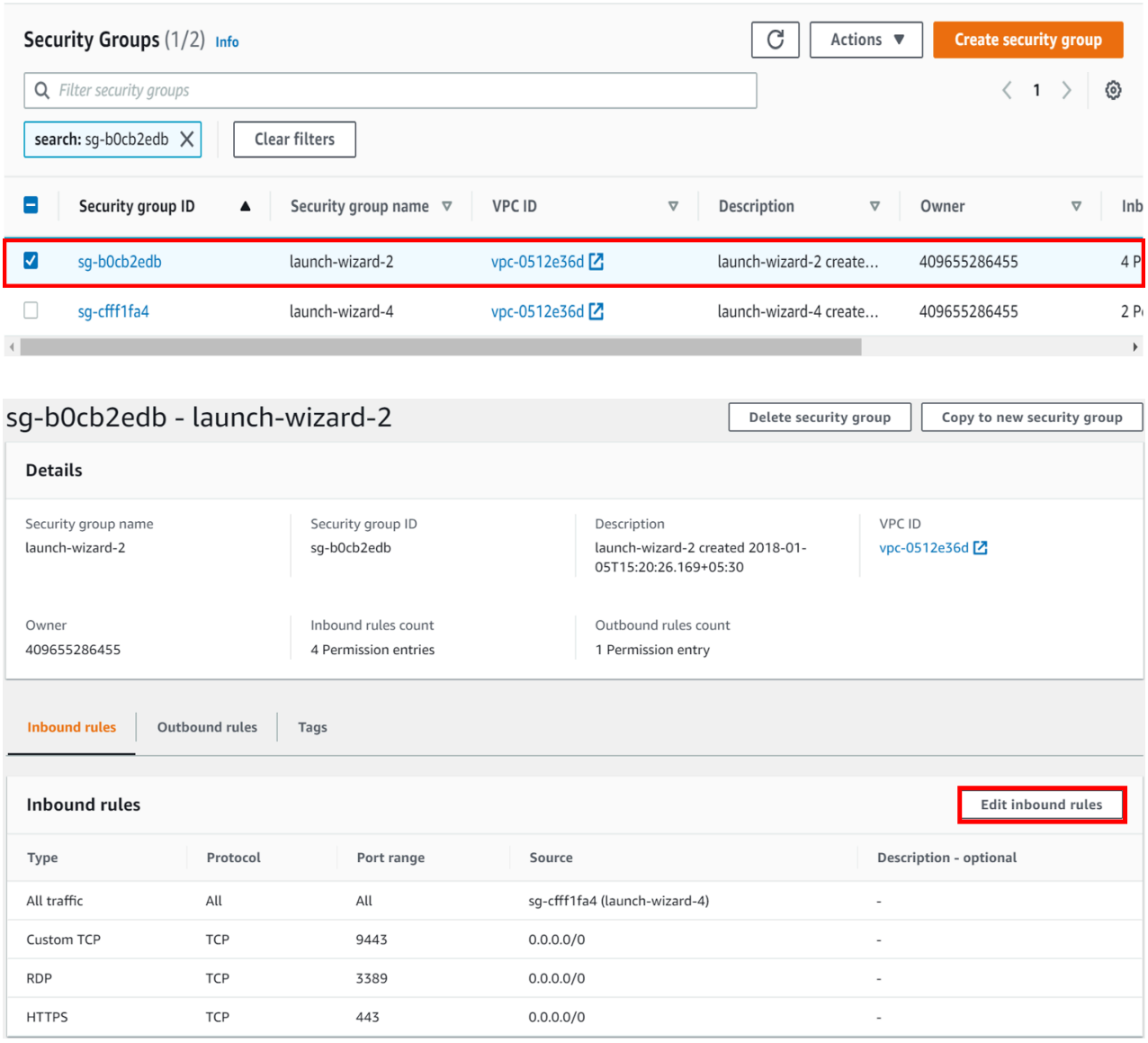
To prepare for appliance deployment:

- Set up a separate EC2 VM to host the replication appliance. This instance must be running Windows Server 2022. Review the hardware, software, and networking requirements for the appliance.

- The appliance shouldn't be installed on a source VM that you want to replicate or on the Azure Migrate: Discovery and assessment appliance you might have installed before. It should be deployed on a different VM.

- The source AWS VMs to be migrated should have a network line of sight to the replication appliance. Configure necessary security group rules to enable this capability. We recommend that you deploy the replication appliance in the same virtual private cloud (VPC) as the source VMs to be migrated. If the replication appliance needs to be in a different VPC, the VPCs must be connected through VPC peering.

- The source AWS VMs communicate with the replication appliance on ports HTTPS 443 (control channel orchestration) and TCP 9443 (data transport) inbound for replication management and replication data transfer. The replication appliance in turn orchestrates and sends replication data to Azure over port HTTPS 443 outbound. To configure these rules, edit the security group inbound/outbound rules with the appropriate ports and source IP information.



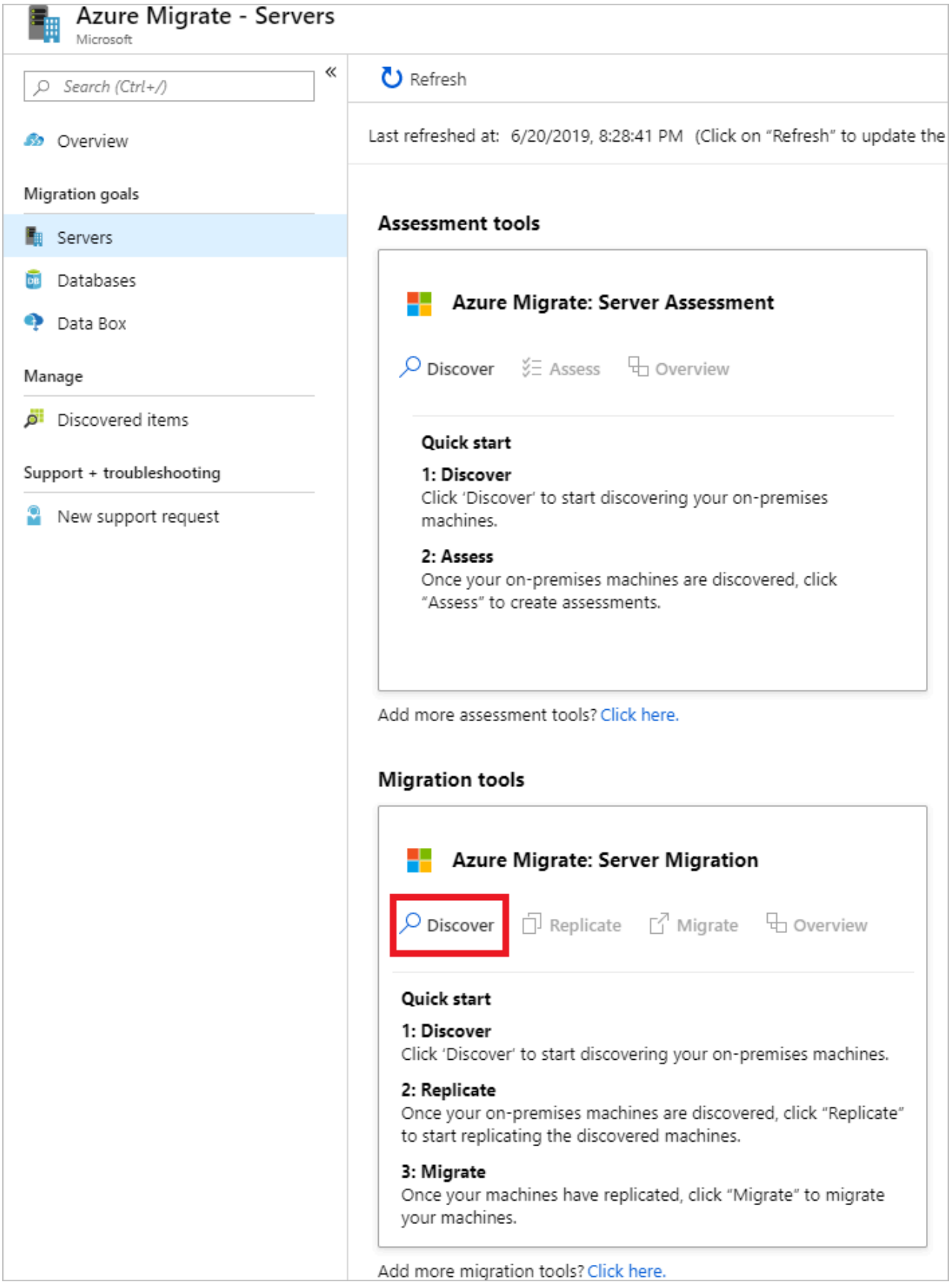
- The replication appliance uses MySQL. Review the options for installing MySQL on the appliance.
- Review the Azure URLs required for the replication appliance to access public and government clouds.

Set up the replication appliance

The first step of migration is to set up the replication appliance. To set up the appliance for AWS VMs migration, you must download the installer file for the appliance and then run it on the VM you prepared.

Download the replication appliance installer

1. In the Azure Migrate project, select **Servers, databases, and web apps** > **Migration and modernization** > **Discover**.



2. In **Discover machines** > **Are your machines virtualized?**, select **Not virtualized/Other**.

3. In **Target region**, select the Azure region to which you want to migrate the machines.

4. Select **Confirm that the target region for migration is <region-name>**.

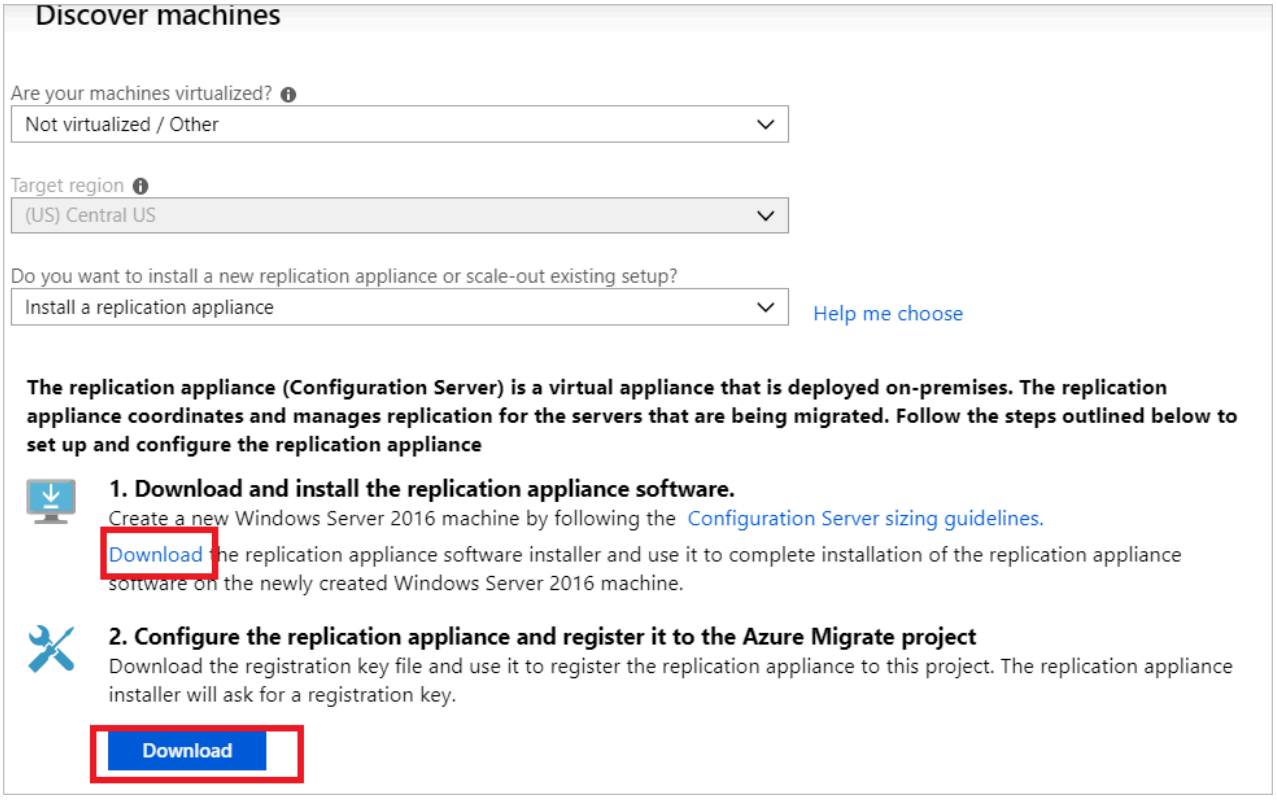
5. Select **Create resources**. This step creates an Azure Site Recovery vault in the background.

- If you already set up migration with the Migration and modernization tool, the target option can't be configured because resources were set up previously.
- You can't change the target region for this project after you select this button.
- To migrate your VMs to a different region, you need to create a new or different Azure Migrate project.

> ⓘ Note
>
> If you selected private endpoint as the connectivity method for the Azure Migrate project when it was created, the Recovery Services vault is also configured for private endpoint connectivity. Ensure that the private endpoints are reachable from the replication appliance. Learn more.

6. In **Do you want to install a new replication appliance?**, select **Install a replication appliance**.

7. In **Download and install the replication appliance software**, download the appliance installer and the registration key. You need the key to register the appliance. The key is valid for five days after download.

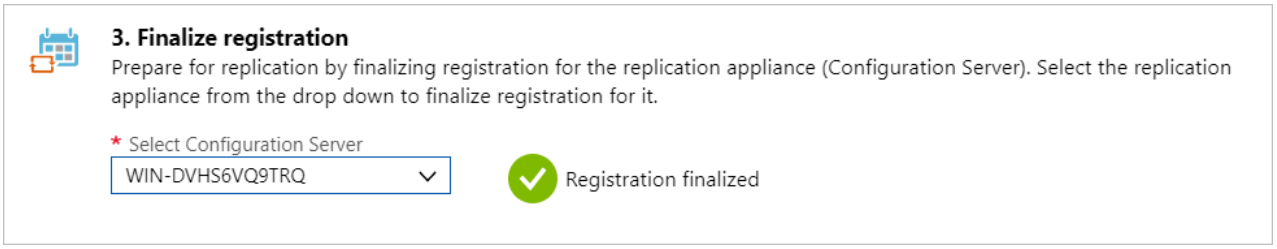


8. Copy the appliance setup file and key file to the Windows Server 2022 AWS VM you created for the replication appliance.

9. Run the replication appliance setup file, as described in the next procedure.
    a. Under **Before You Begin**, select **Install the configuration server and process server**. Then select **Next**.
    b. In **Third-Party Software License**, select **I accept the third-party license agreement**. Then select **Next**.
    c. In **Registration**, select **Browse**, and then go to where you put the vault registration key file. Select **Next**.
    d. In **Internet Settings**, select **Connect to Azure Site Recovery without a proxy server**. Then select **Next**.
    e. The **Prerequisites Check** page runs checks for several items. When it's finished, select **Next**.
    f. In **MySQL Configuration**, provide a password for the MySQL database. Then select **Next**.
    g. In **Environment Details**, select **No**. You don't need to protect your VMs. Then select **Next**.
    h. In **Install Location**, select **Next** to accept the default.
    i. In **Network Selection**, select **Next** to accept the default.
    j. In **Summary**, select **Install**.
    k. **Installation Progress** shows you information about the installation process. When it's finished, select **Finish**. A window displays a message about a reboot. Select **OK**.
    l. Next, a window displays a message about the configuration server connection passphrase. Copy the passphrase to your clipboard and save the passphrase in a temporary text file on the source VMs. You need this passphrase later during the Mobility service installation process.

10. After the installation finishes, the Appliance configuration wizard launches automatically. (You can also launch the wizard manually by using the `cspsconfigtool` shortcut that was created on the appliance desktop.) In this tutorial, we manually install the Mobility service on source VMs to be replicated. You need to create a dummy account in this step to proceed. For your dummy account, use "guest" as the friendly name, "username" as the username, and "password" as the password for the account. You use this dummy account in the Enable Replication stage.

11. After the appliance restarts after setup, in **Discover machines**, select the new appliance in **Select Configuration Server** and select **Finalize registration**. The **Finalize registration** step performs a couple of final tasks to prepare the replication appliance.



Install the Mobility service agent

A Mobility service agent must be preinstalled on the source AWS VMs to be migrated before you can initiate replication. The approach you choose to install the Mobility service agent might depend on your organization's preferences and existing tools. The "push" installation method built into Site Recovery isn't currently supported. Approaches you might want to consider:

- AWS System Manager
- System Center Configuration Manager
- Azure Arc for servers and custom script extensions
- Install Mobility agent for Windows
- Install Mobility agent for Linux

1. Extract the contents of the installer tarball to a local folder (for example, /tmp/MobSvcInstaller) on the AWS VM, as follows:

```
mkdir /tmp/MobSvcInstaller
tar -C /tmp/MobSvcInstaller -xvf <Installer tarball>
cd /tmp/MobSvcInstaller
```

2. Run the installer script:

```
sudo ./install -r MS -v VmWare -q -c CSLegacy
```

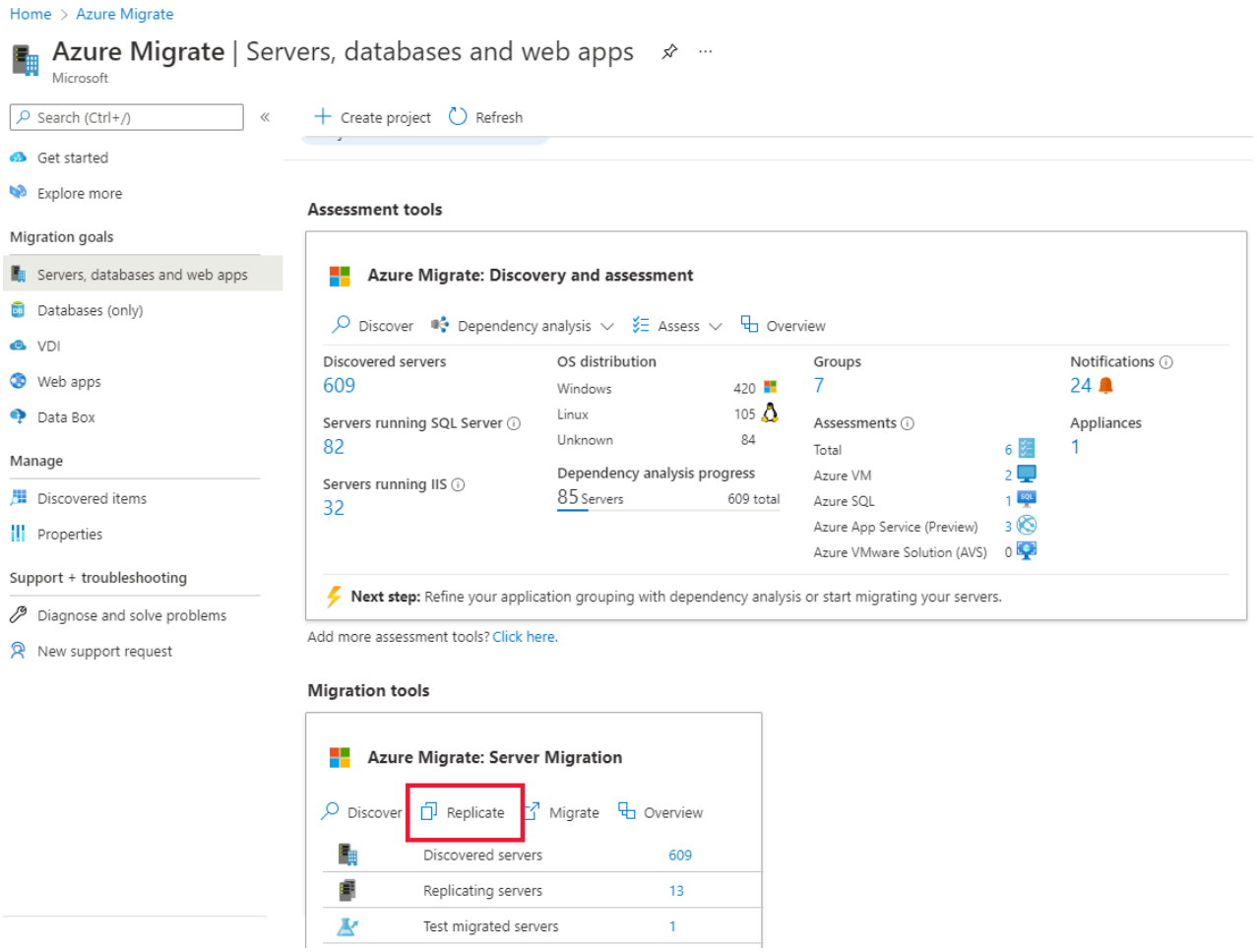
3. Register the agent with the replication appliance:

```
/usr/local/ASR/Vx/bin/UnifiedAgentConfigurator.sh -i <replication appliance IP address> -P <Passphrase File Path>
```

Enable replication for AWS VMs

> **Note**
>
> Through the portal, you can add up to 10 VMs for replication at once. To replicate more VMs simultaneously, you can add them in batches of 10.

1. In the Azure Migrate project, select **Servers, databases, and web apps** > **Migration and modernization** > **Replicate**.



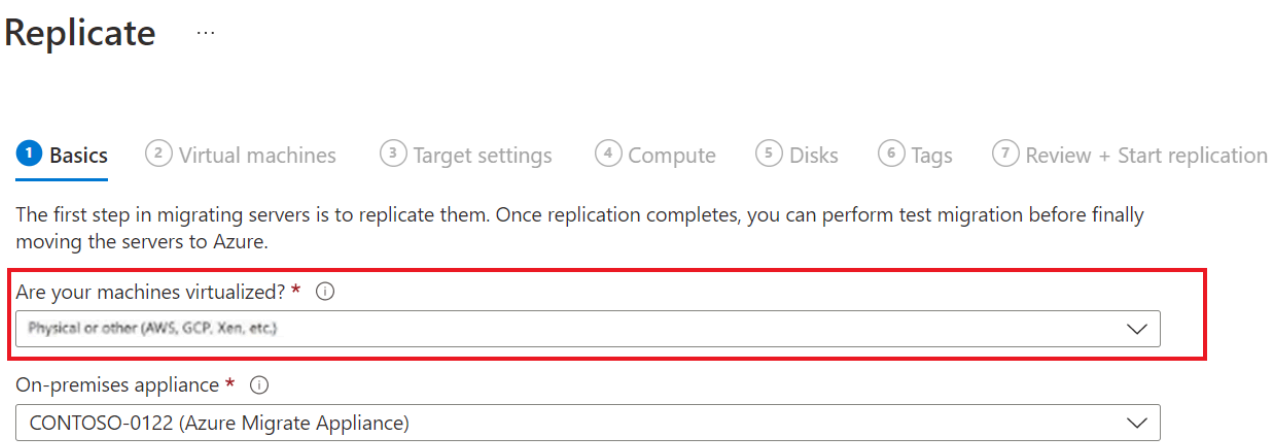
2. In **Replicate**, > **Source settings** > **Are your machines virtualized?**, select **Not virtualized/Other**.

3. In **On-premises appliance**, select the name of the Azure Migrate appliance that you set up.

4. In **Process Server**, select the name of the replication appliance.

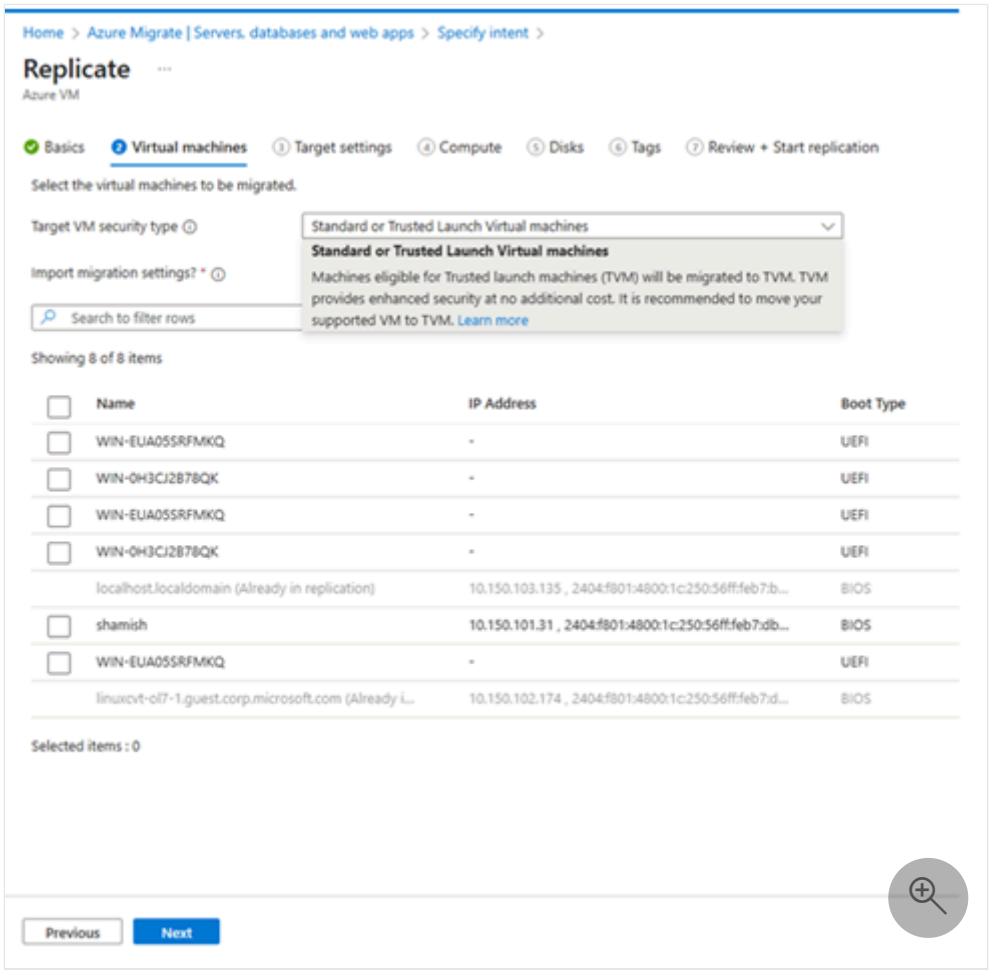
5. In **Guest credentials**, select the dummy account you created previously during the replication installer setup to install the Mobility service manually. (Push installation isn't

supported.) Then select **Next: Virtual machines**.



6. In **Virtual Machines**, in **Import migration settings from an assessment?**, leave the default setting **No, I'll specify the migration settings manually**.

7. Check each VM you want to migrate. Then select **Next: Target settings**.



8. In **Target settings**, select the subscription and target region to which you'll migrate. Specify the resource group in which the Azure VMs will reside after migration.

9. In **Virtual Network**, select the Azure virtual network/subnet to which the Azure VMs will be joined after migration.

10. In **Cache storage account**, keep the default option to use the cache storage account that was automatically created for the project. Use the dropdown list if you want to specify a different storage account to use as the cache storage account for replication.

    > ⓘ **Note**
    >
    > - If you selected private endpoint as the connectivity method for the Azure Migrate project, grant the Recovery Services vault access to the cache storage account. Learn more.
    > - To replicate by using Azure ExpressRoute with private peering, create a private endpoint for the cache storage account. Learn more.

11. In **Availability options**, select:

    - **Availability Zone**: Pins the migrated machine to a specific availability zone in the region. Use this option to distribute servers that form a multinode application tier across availability zones. If you select this option, you need to specify the availability zone to use for each of the selected machines on the **Compute** tab. This option is only available if the target region selected for the migration supports availability zones.
    - **Availability Set**: Places the migrated machine in an availability set. The target resource group that was selected must have one or more availability sets in order to use this option.
    - **No infrastructure redundancy required**: Select this option if you don't need either of these availability configurations for the migrated machines.

12. In **Disk encryption type**, select:

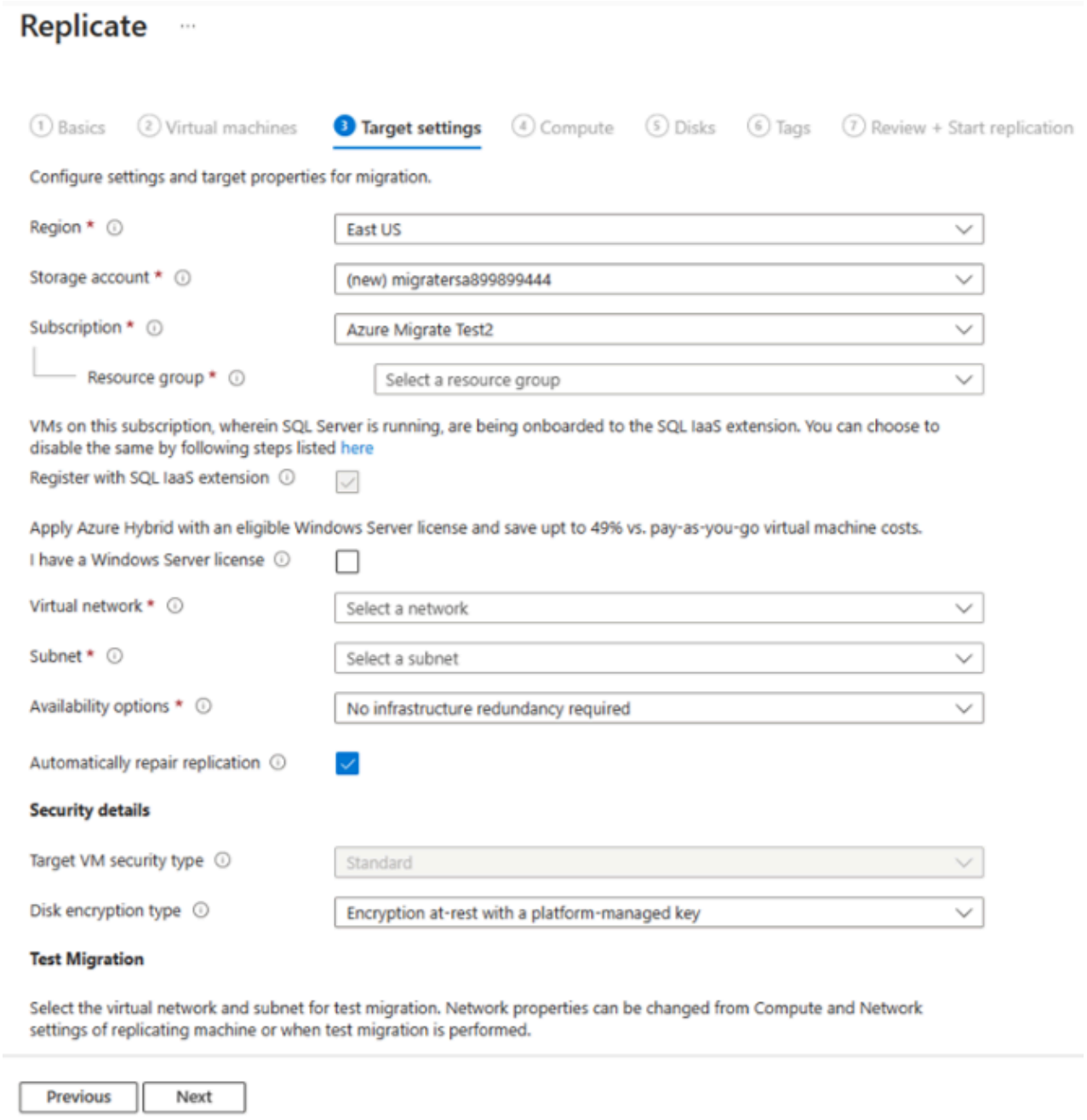
    - Encryption-at-rest with a platform-managed key.
    - Encryption-at-rest with a customer-managed key.
    - Double encryption with platform-managed and customer-managed keys.

    > ⓘ **Note**
    >
    > To replicate VMs with customer-managed keys, you need to create a disk encryption set under the target resource group. A disk encryption set object maps managed

disks to an Azure Key Vault instance that contains the customer-managed key to use for server-side encryption.

13. In **Azure Hybrid Benefit**:

    - Select **No** if you don't want to apply Azure Hybrid Benefit. Then select **Next**.
    - Select **Yes** if you have Windows Server machines that are covered with active Software Assurance or Windows Server subscriptions, and you want to apply the benefit to the machines you're migrating. Then select **Next**.

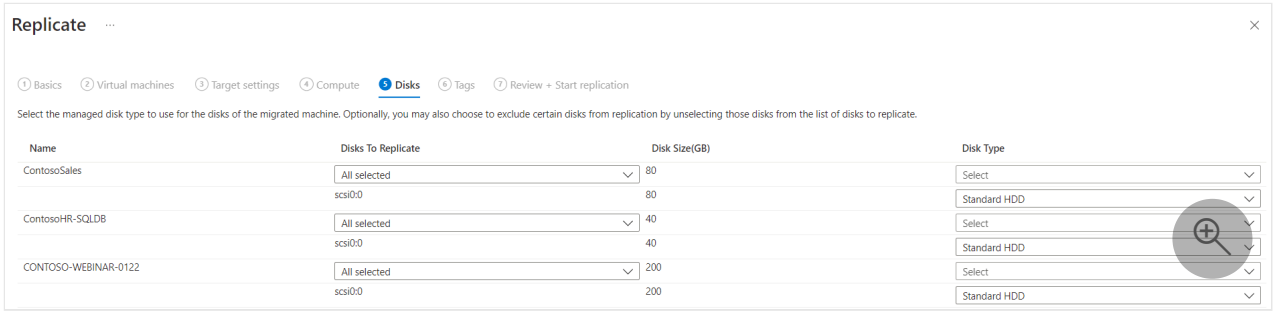


14. In **Compute**, review the VM name, size, OS disk type, and availability configuration (if selected in the previous step). VMs must conform with Azure requirements.

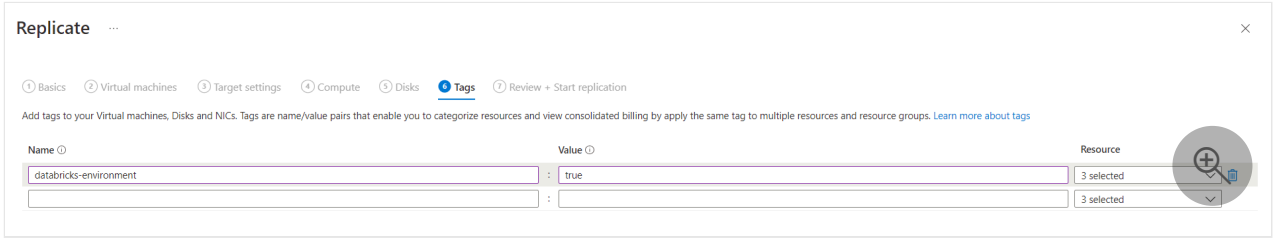
- **VM size**: If you're using assessment recommendations, the VM size dropdown list shows the recommended size. Otherwise, Azure Migrate and Modernize picks a size based on the closest match in the Azure subscription. Alternatively, pick a manual size in **Azure VM size**.
- **OS disk**: Specify the OS (boot) disk for the VM. The OS disk is the disk that has the operating system bootloader and installer.
- **Availability Zone**: Specify the availability zone to use.
- **Availability Set**: Specify the availability set to use.

15. In **Disks**, specify whether the VM disks should be replicated to Azure and select the disk type (Standard SSD/HDD or premium managed disks) in Azure. Then select **Next**.

    - You can exclude disks from replication.
    - If you exclude disks, they won't be present on the Azure VM after migration.



16. In **Tags**, choose to add tags to your virtual machines, disks, and NICs.



17. In **Review and start replication**, review the settings and select **Replicate** to start the initial replication for the servers.

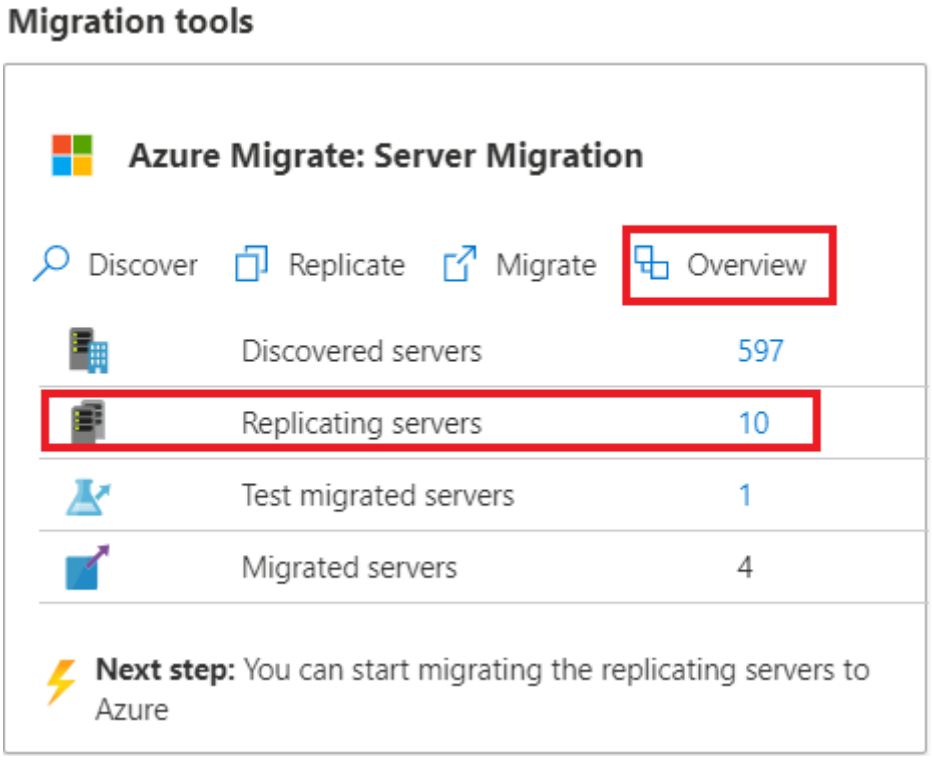
> **Note**
>
> To update replication settings any time before replication starts, select **Manage** > **Replicating machines**. Settings can't be changed after replication starts.

Track and monitor replication status

- When you select **Replicate**, a Start Replication job begins.
- When the Start Replication job finishes successfully, the VMs begin their initial replication to Azure.
- After that initial replication finishes, delta replication begins. Incremental changes to AWS VM disks are periodically replicated to the replica disks in Azure.

You can track job status in the portal notifications.

You can monitor replication status by selecting **Replicating servers** in **Azure Migrate: Server Migration**.



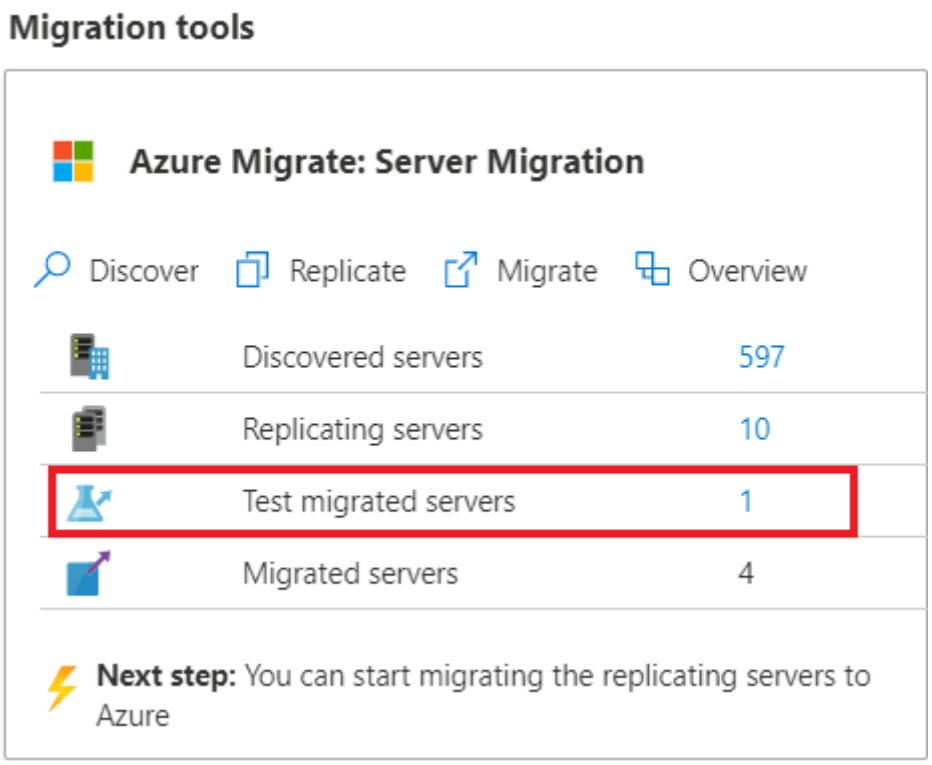
Run a test migration

When delta replication begins, you can run a test migration for the VMs before you run a full migration to Azure. We highly recommend the test migration. It provides an opportunity to discover any potential issues and fix them before you proceed with the actual migration. We recommend that you do this step at least once for each VM before you migrate it.

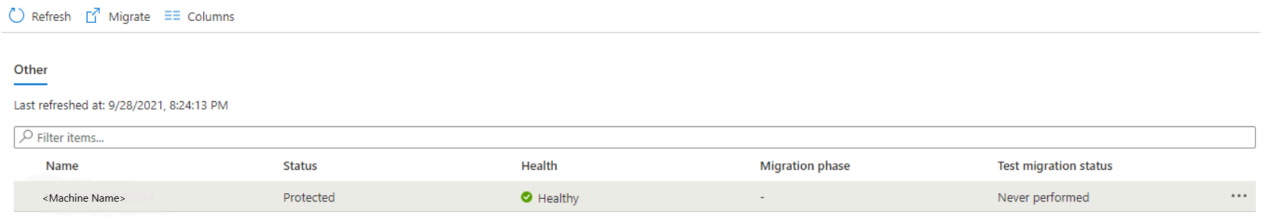
- Running a test migration checks that migration works as expected without affecting the AWS VMs, which remain operational and continue replicating.
- Test migration simulates the migration by creating an Azure VM by using replicated data. (The test usually migrates to a nonproduction virtual network in your Azure subscription.)
- You can use the replicated test Azure VM to validate the migration, perform app testing, and address any issues before full migration.

To do a test migration:

1. In **Migration goals**, select **Servers, databases, and web apps** > **Migration and modernization** > **Test migrated servers**.



2. Right-click the VM you want to test and select **Test migrate**.

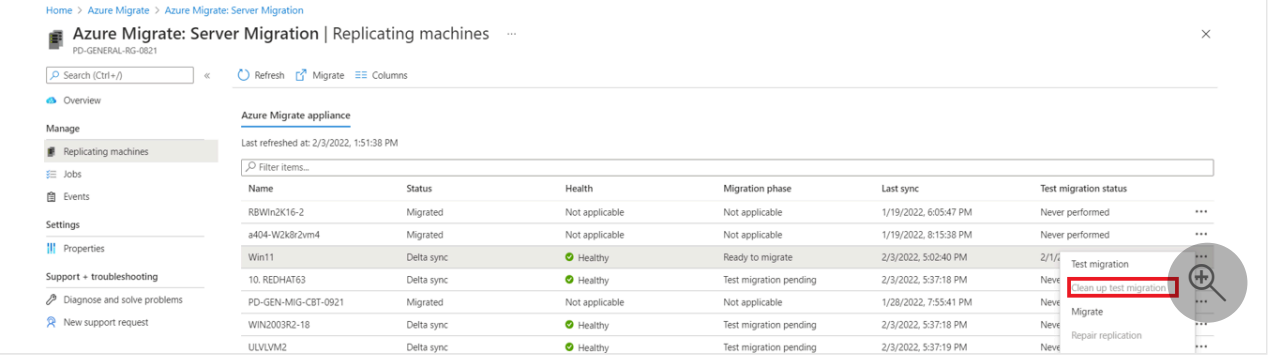


3. In **Test Migration**, select the Azure virtual network in which the Azure VM will be located after the migration. We recommend that you use a nonproduction virtual network.

4. The Test Migration job starts. Monitor the job in the portal notifications.

5. After the migration finishes, view the migrated Azure VM in **Virtual Machines** in the Azure portal. The machine name has the suffix **-Test**.

6. After the test is finished, right-click the Azure VM in **Replicating machines** and select **Clean up test migration**.

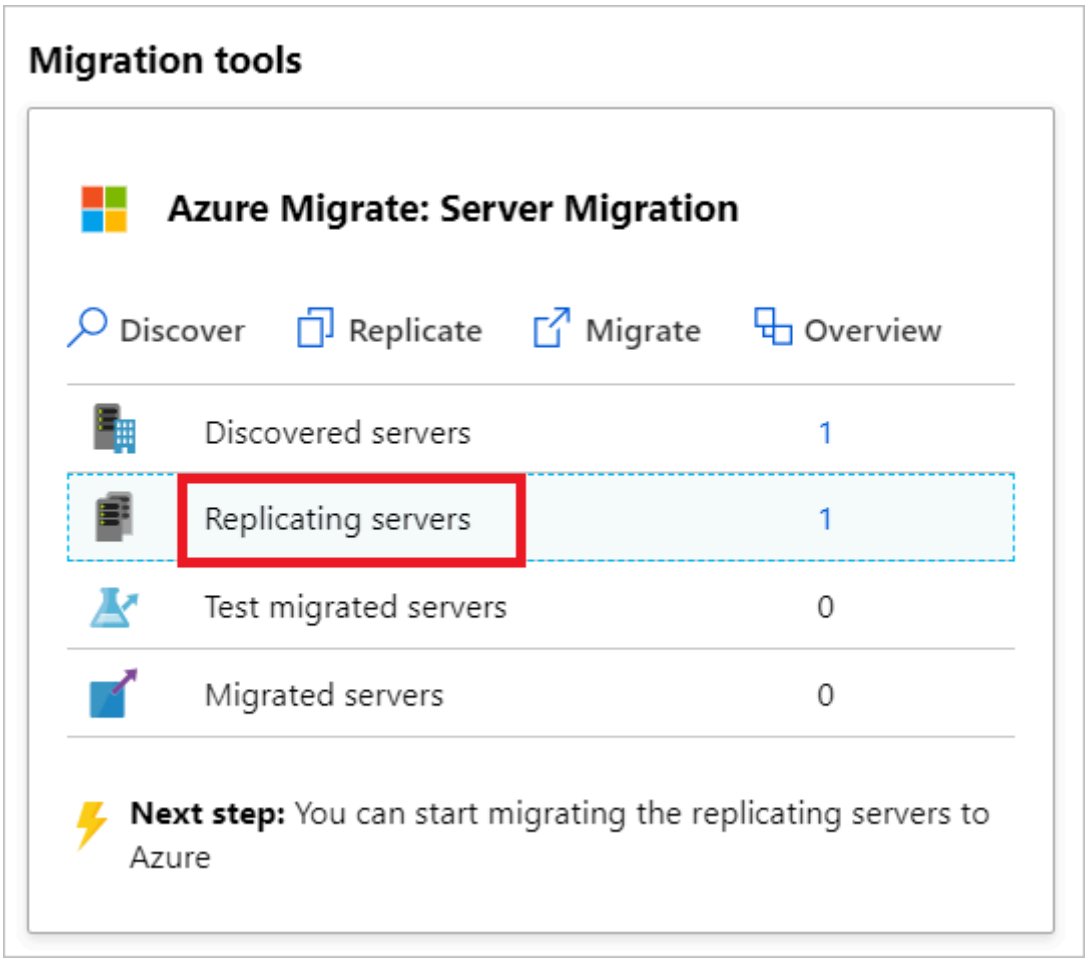


> **Note**
>
> You can now register your servers running SQL Server with SQL VM RP to take advantage of automated patching, automated backup, and simplified license management by using the SQL IaaS Agent Extension.
>
> - Select **Manage** > **Replicating servers** > **Machine containing SQL server** > **Compute and Network** and select **yes** to register with SQL VM RP.
> - Select **Azure Hybrid Benefit for SQL Server** if you have SQL Server instances that are covered with active Software Assurance or SQL Server subscriptions and you want to apply the benefit to the machines you're migrating to.

Migrate AWS VMs

After you verify that the test migration works as expected, you can migrate the AWS VMs.

1. In the Azure Migrate project, select **Servers, databases, and web apps** > **Migration and modernization** > **Replicating servers**.



2. In **Replicating machines**, right-click the VM and select **Migrate**.

3. In **Migrate** > **Shut down virtual machines and perform a planned migration with no data loss**, select **Yes** > **OK**.

   > ⓘ **Note**
   >
   > Automatic shutdown isn't supported while migrating AWS VMs.

4. A migration job starts for the VM. You can view the job status by selecting the notification bell icon on the top right of the portal page. You can also go to the **Jobs** page of the Migration and modernization tool. (Select **Overview** on the tool tile and select **Jobs** from the left menu.)

5. After the job finishes, you can view and manage the VM from the **Virtual Machines** page.

Complete the migration

1. After the migration is finished, right-click the VM and select **Stop migration**. This action:

   - Stops replication for the AWS VM.

    - Removes the AWS VM from the **Replicating servers** count in the Migration and modernization tool.
    - Cleans up replication state information for the VM.
2. Verify and troubleshoot any Windows activation issues on the Azure VM.
3. Perform any post-migration app tweaks, such as updating host names, database connection strings, and web server configurations.
4. Perform final application and migration acceptance testing on the migrated application now running in Azure.
5. Cut over traffic to the migrated Azure VM instance.
6. Update any internal documentation to show the new location and IP address of the Azure VMs.

Post-migration best practices

- For increased resilience:
    - Keep data secure by backing up Azure VMs by using Azure Backup. Learn more.
    - Keep workloads running and continuously available by replicating Azure VMs to a secondary region with Site Recovery. Learn more.
- For increased security:
    - Lock down and limit inbound traffic access with Microsoft Defender for Cloud - Just-in-time administration.
    - Manage and govern updates on Windows and Linux machines with Azure Update Manager.
    - Restrict network traffic to management endpoints with network security groups.
    - Deploy Azure Disk Encryption to help secure disks and keep data safe from theft and unauthorized access.
    - Read more about securing IaaS resources and Microsoft Defender for Cloud.
- For monitoring and management:
    - Consider deploying Microsoft Cost Management to monitor resource usage and spending.

Troubleshooting and tips

Question: I can't see my AWS VM in the discovered list of servers for migration.
Answer: Check if your replication appliance meets the requirements. Make sure Mobility Agent is installed on the source VM to be migrated and is registered to the Configuration Server.

Check the network setting and firewall rules to enable a network path between the replication appliance and source AWS VMs.

Question: How do I know if my VM was successfully migrated?
Answer: Post migration, you can view and manage the VM from the **Virtual Machines** page. Connect to the migrated VM to validate.

Question: I'm unable to import VMs for migration from my previously created Server Assessment results.
Answer: Currently, we don't support the import of assessment for this workflow. As a workaround, you can export the assessment and then manually select the VM recommendation during the Enable Replication step.

Question: I'm getting the error "Failed to fetch BIOS GUID" while trying to discover my AWS VMs.
Answer: Always use root login for authentication and not any pseudo user. Also, review supported operating systems for AWS VMs.

Question: My replication status isn't progressing.
Answer: Check if your replication appliance meets the requirements. Make sure that you enabled the required ports on your replication appliance TCP port 9443 and HTTPS 443 for data transport. Ensure that no stale duplicate versions of the replication appliance are connected to the same project.

Question: I'm unable to discover AWS instances by using Azure Migrate and Modernize because of the HTTP status code of 504 from the remote Windows management service.
Answer: Make sure to review the Azure Migrate appliance requirements and URL access needs. Make sure no proxy settings are blocking the appliance registration.

Question: Do I have to make any changes before I migrate my AWS VMs to Azure?
Answer: You might have to make the following changes before you migrate your EC2 VMs to Azure:

- If you're using cloud-init for your VM provisioning, you might want to disable cloud-init on the VM before you replicate it to Azure. The provisioning steps performed by cloud-init on the VM might be specific to AWS and won't be valid after the migration to Azure.
- If the VM is a paravirtualized (PV) VM and not a hardware VM, you might not be able to run it as is on Azure. PV VMs use a custom boot sequence in AWS. You might be able to

overcome this challenge by uninstalling PV drivers before you perform a migration to Azure.

- We always recommend that you run a test migration before the final migration.

Question: Can I migrate AWS VMs running the Amazon Linux operating system?
Answer: VMs running Amazon Linux can't be migrated as is because the Amazon Linux OS is only supported on AWS. To migrate workloads running on Amazon Linux, you can spin up a RHEL VM in Azure. Then you can migrate the workload running on the AWS Linux machine by using a relevant workload migration approach. For example, depending on the workload, there might be workload-specific tools to aid the migration. These tools might be for databases or deployment tools for web servers.

Next steps

Investigate the cloud migration journey in the Cloud Adoption Framework for Azure.

Last updated on 04/23/2025

Retired VM Sizes Migration Guide

This migration guide is designed for users of Azure virtual machines (VMs) scheduled for retirement. This guide helps you transition to the latest VM series, helping you minimize disruptions while optimizing cost and performance. The guide covers General Purpose, Storage Optimized, and other VM series. It also surfaces critical information for HPC HBv2-series VMs undergoing retirement; specialized workload validation is recommended when migrating HPC workloads.

This guide covers:

- Recommended replacement VM series
- Detailed migration steps
- Common questions and guidance on handling RIs.

By migrating to newer VM series, you gain access to improved price-performance ratios, broader regional availability, and the latest hardware capabilities.

Recommended Replacement VM Series

Expand table

Current VM Series	Target VM Series	Differences in Specification in Target VM*
D Ds Dv2 Dsv2	Dsv5/Ddsv5/Dasv5/Dadsv5 Dasv6/Dadsv6/Dsv6/Ddsv6 Dasv7/Dadsv7 Esv6/Edsv6/Easv6/Eadsv6 Easv7/Eadsv7	D/Ev5 disk controller type: SCSI D/Ev6, D/Ev7 disk controller type: NVMe Local Storage Throughput: 9000 IOPS / 125 MBps Remote Storage Throughput: 3750 IOPS / 82 MBps
Ls	Lsv3/Lasv3 Lsv4/Lasv4	Local Storage: Supported - NVMe Remote Storage Throughput: 12800 IOPS / 200 MBps Disk Controller Type: SCSI and NVMe
Av2 Amv2	Bsv2/Basv2 Dsv5/Ddv5/Dasv5 Esv5/Edv5/Easv5 Dsv6/Ddsv6/Dasv6 Esv6/Edsv6/Easv6	B/Bp/Bav2, D/Ev5 disk controller type: SCSI D/Ev6 disk controller type: NVMe Remote Storage Throughput: 3750 IOPS / 85 MBps

Current VM Series	Target VM Series	Differences in Specification in Target VM*
Bv1	Bsv2/Basv2 Dlsv5/Dldsv5/Dalsv5/Daldsv5 Dlsv6/Dldsv6/Dalsv6/Daldsv6	B/Bp/Bav2, D/Ev5 disk controller type: SCSI D/Ev6 disk controller type: NVMe Remote Storage Throughput: 3750 IOPS / 85 MBps Disk Controller Type: SCSI
F Fs Fsv2	Dlsv6/Dldsv6/Dalsv6/Daldsv6 Falsv6 Dldsv5/Dlsv5/Dsv5/Ddsv5	D/Ev5 disk controller type: SCSI D/Ev6 disk controller type: NVMe Remote Storage Throughput: 4167 IOPS / 124 MBps
G Gs	Lsv3/Lasv3 Lsv4/Lasv4	Lv3/Lv4 controller type: SCSI and NVMe Remote Storage Throughput: 12800 IOPS / 200 MBps
Lsv2	Lsv3/Lasv3 Lasv4/Lasv4	Local Storage: NVMe Remote Storage Throughput: 12800 IOPS / 200 MBps Disk Controller Type: SCSI and NVMe
HBv2	HBv5 HX HBv4 HBv3	Transition to newer AMD EPYC generations (Milan/Genoa/Zen4) and newer InfiniBand fabrics (HDR on HBv3 → NDR on HBv4/HBv5). Validate MPI workload performance, memory bandwidth requirements, and RDMA compatibility on the target series.

*Refers to the smallest VM size in the given target VM series. Full VM specifications are available on each target VM series' product sizes page.

*Refers to the smallest VM size in the given target VM series. Full VM specifications are available on each target VM series' product sizes page.

> **Important**
>
> The following SKUs aren't available in the Sovereign clouds: Bsv2, Bpsv2, Basv2

Recommended Replacement Isolated VM Sizes

Expand table

Current VM Size	Target VM Sizes	Differences in Specification in Target VM*
Standard_E64i_v3 Standard_E64is_v3	Standard_E192is_v6 Standard_E192ids_v6 Standard_E104i_v5 Standard_E104id_v5	Local Storage: Supported - NVMe Local Storage Throughput: 37,500 IOPS / 180 MBps Remote Storage Throughput: 3,750 IOPS / 106 MBps Disk Controller Type: NVMe

Current VM Size	Target VM Sizes	Differences in Specification in Target VM*
	Standard_E104is_v5 Standard_E104ids_v5 Standard_E80is_v4 Standard_E80ids_v4	

*Refers to the smallest VM size in the given target VM series. Full VM specifications are available on each target VM series' product sizes page.

For optimal performance and experience, we recommend using the newer v5 and v6 VM series. This ensures you have access to the latest features such as Premium Storage, Accelerated Networking, and Nested Virtualization. While the v6 VM series is preferred, there are certain scenarios where you might want to consider the v5 or even the v4 VM series. Here are some reasons why:

- v6 VMs require enabling NVMe which means that you must have a supported OS.
- v6 VMs support Generation 2 VMs only.
- v6 VMs require MANA (Microsoft Azure Network Adapter) and a MANA supported operating system.
- v6 VMs may not have available capacity in the regions and zones you need.

Note that Lsv4 and Lasv4 series are the latest generation L-series VMs.

Use the Azure VM size documentation to help identify suitable VM sizes.

Migration Steps

Optional: For Reserved Instance (RI) customers only

- Review your current reservations using the Azure Reservation Management page.
- If applicable, exchange existing reservations for newer VM series or trade in your reservations for an **Azure Savings Plan for compute**.
- **HBv2 customers**: 1-year and 3-year HBv2 Reserved Instance purchases ended on April 2, 2026. If you have active HBv2 RIs, consider exchanging them for supported HPC series RIs (such as HBv3, HBv4, HBv5, or HX) or trading them in for an Azure Savings Plan for compute before the retirement date of May 31, 2027.

Identify the Target VM Size

- Evaluate your current VM's workload and performance requirements.

- Select a comparable size from the above table that meets your CPU, memory, and storage needs.

Check and Request Quota Increases

- Before resizing, verify that your subscription has sufficient quota for the target v6 VM series.
- Request more quota through the Azure portal if needed.

Resize the Virtual Machine

You can resize your VM through the Azure portal, Azure CLI, or PowerShell. Follow these steps:

1. **Stop (deallocate) the VM**.
2. **Resize** the VM to your selected v6 series.
3. **Start the VM** after resizing.

Refer to the full Azure VM resizing guide for more detailed instructions.

FAQ

Q: Which sizes are being retired?

To review retired sizes, see retired Azure VM sizes. View retired isolated sizes at Isolation for VMs in Azure.

> **Note**
>
> HPC HBv2-series VMs are also retiring. See the HBv2 row in the table below for the applicable dates.

Expand table

VM Series	3 YR RI expiration date	1 YR RI expiration date	Retirement Date
D	05/01/2025	05/01/2027	05/01/2028
Ds	05/01/2025	05/01/2027	05/01/2028
Dv2	05/01/2025	05/01/2027	05/01/2028
Dsv2	05/01/2025	05/01/2027	05/01/2028

VM Series	3 YR RI expiration date	1 YR RI expiration date	Retirement Date
Ls	05/01/2025	05/01/2027	05/01/2028
F	11/15/2025	11/15/2027	11/15/2028
Fs	11/15/2025	11/15/2027	11/15/2028
Fsv2	11/15/2025	11/15/2027	11/15/2028
Lsv2	11/15/2025	11/15/2027	11/15/2028
G	11/15/2025	11/15/2027	11/15/2028
Gs	11/15/2025	11/15/2027	11/15/2028
Av2	11/15/2025	11/15/2027	11/15/2028
Amv2	11/15/2025	11/15/2027	11/15/2028
B-series	11/15/2025	11/15/2027	11/15/2028
Bv1	11/15/2025	11/15/2027	11/15/2028
HBv2	04/02/2026	04/02/2026	05/31/2027

Q: Why should I migrate my VM?

Migration is mandatory to avoid unexpected shutdown. Additionally, migration yields the following benefits:

- **Performance**: Newer VM series offer better price-to-performance ratios.
- **Regional Availability**: The v5 and v6 series has broader regional support across Azure data centers.
- **Future-proofing**: Migrate ahead of the retirement schedule to avoid disruption.

Q: What will happen to my VM if I do not resize my VM to a target size within the retirement timeline?

After retirement, VMs using this size will be deallocated and stop incurring charges. The size is no longer supported or covered by an SLA; in-memory and temporary disk data is lost, but managed disk data is preserved. To resume service, you may resize to a supported size and restart the VM.

Specifically for HBv2-series: after May 31, 2027, HBv2-series VMs (Standard_HB120rs_v2 and derived sizes) will be automatically set to a deallocated state. After that date, HBv2 VMs will

stop working, lose SLA and support, and stop incurring billing charges.

Q: Can I recover my VM after it has been deallocated?

Yes, you can resize and restart your deallocated VM following the Azure VM resizing guide.

Q: Will VM migration disrupt pay-as-you-go or Savings Plan Pricing billing?

No. If you're using pay-as-you-go or a savings plan, migrating to a newer VM type won't disrupt your current billing. The migration process remains seamless with no changes required in your subscription or payment plan.

Q: How can I migrate my VM if I am on Reserved Instances (RIs) with a retired VM?

If you have active Reserved Instances for D, Dv2, Dsv2, or L-series VMs, follow these steps:

Step 1: Review Current Reservations

- Check your active RIs in the Azure portal.

Identify which RIs are expiring or will be affected by the VM retirement.

Step 2: Migrate and Manage Your RIs

Depending on your business needs, consider these options:

1. **Exchange Existing Reservations**:
    - Swap current RIs for a new VM series without any penalties.
    - Refer to the RI Exchange Guide
2. **Trade-In for Savings Plan**:
    - Convert your existing RIs into an **Azure Savings Plan for compute**.
    - This offers flexibility across VM families and regions.
    - Follow the Azure RI Trade-In Tutorial.
3. **Purchase New RIs**:
    - Buy new reservations that align with your new v6 VM series.

- Consider shorter terms (1-year) for flexibility.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/14/2026

Retired VM Sizes Migration Guide

This migration guide is designed for users of Azure virtual machines (VMs) scheduled for retirement. This guide helps you transition to the latest VM series, helping you minimize disruptions while optimizing cost and performance. The guide covers General Purpose, Storage Optimized, and other VM series. It also surfaces critical information for HPC HBv2-series VMs undergoing retirement; specialized workload validation is recommended when migrating HPC workloads.

This guide covers:

- Recommended replacement VM series
- Detailed migration steps
- Common questions and guidance on handling RIs.

By migrating to newer VM series, you gain access to improved price-performance ratios, broader regional availability, and the latest hardware capabilities.

Recommended Replacement VM Series

Expand table

Current VM Series	Target VM Series	Differences in Specification in Target VM*
D Ds Dv2 Dsv2	Dsv5/Ddsv5/Dasv5/Dadsv5 Dasv6/Dadsv6/Dsv6/Ddsv6 Dasv7/Dadsv7 Esv6/Edsv6/Easv6/Eadsv6 Easv7/Eadsv7	D/Ev5 disk controller type: SCSI D/Ev6, D/Ev7 disk controller type: NVMe Local Storage Throughput: 9000 IOPS / 125 MBps Remote Storage Throughput: 3750 IOPS / 82 MBps
Ls	Lsv3/Lasv3 Lsv4/Lasv4	Local Storage: Supported - NVMe Remote Storage Throughput: 12800 IOPS / 200 MBps Disk Controller Type: SCSI and NVMe
Av2 Amv2	Bsv2/Basv2 Dsv5/Ddv5/Dasv5 Esv5/Edv5/Easv5 Dsv6/Ddsv6/Dasv6 Esv6/Edsv6/Easv6	B/Bp/Bav2, D/Ev5 disk controller type: SCSI D/Ev6 disk controller type: NVMe Remote Storage Throughput: 3750 IOPS / 85 MBps

Current VM Series	Target VM Series	Differences in Specification in Target VM*
Bv1	Bsv2/Basv2 Dlsv5/Dldsv5/Dalsv5/Daldsv5 Dlsv6/Dldsv6/Dalsv6/Daldsv6	B/Bp/Bav2, D/Ev5 disk controller type: SCSI D/Ev6 disk controller type: NVMe Remote Storage Throughput: 3750 IOPS / 85 MBps Disk Controller Type: SCSI
F Fs Fsv2	Dlsv6/Dldsv6/Dalsv6/Daldsv6 Falsv6 Dldsv5/Dlsv5/Dsv5/Ddsv5	D/Ev5 disk controller type: SCSI D/Ev6 disk controller type: NVMe Remote Storage Throughput: 4167 IOPS / 124 MBps
G Gs	Lsv3/Lasv3 Lsv4/Lasv4	Lv3/Lv4 controller type: SCSI and NVMe Remote Storage Throughput: 12800 IOPS / 200 MBps
Lsv2	Lsv3/Lasv3 Lasv4/Lasv4	Local Storage: NVMe Remote Storage Throughput: 12800 IOPS / 200 MBps Disk Controller Type: SCSI and NVMe
HBv2	HBv5 HX HBv4 HBv3	Transition to newer AMD EPYC generations (Milan/Genoa/Zen4) and newer InfiniBand fabrics (HDR on HBv3 → NDR on HBv4/HBv5). Validate MPI workload performance, memory bandwidth requirements, and RDMA compatibility on the target series.

*Refers to the smallest VM size in the given target VM series. Full VM specifications are available on each target VM series' product sizes page.

*Refers to the smallest VM size in the given target VM series. Full VM specifications are available on each target VM series' product sizes page.

> **Important**
>
> The following SKUs aren't available in the Sovereign clouds: Bsv2, Bpsv2, Basv2

Recommended Replacement Isolated VM Sizes

Expand table

Current VM Size	Target VM Sizes	Differences in Specification in Target VM*
Standard_E64i_v3 Standard_E64is_v3	Standard_E192is_v6 Standard_E192ids_v6 Standard_E104i_v5 Standard_E104id_v5	Local Storage: Supported - NVMe Local Storage Throughput: 37,500 IOPS / 180 MBps Remote Storage Throughput: 3,750 IOPS / 106 MBps Disk Controller Type: NVMe

Current VM Size	Target VM Sizes	Differences in Specification in Target VM*
	Standard_E104is_v5 Standard_E104ids_v5 Standard_E80is_v4 Standard_E80ids_v4	

*Refers to the smallest VM size in the given target VM series. Full VM specifications are available on each target VM series' product sizes page.

For optimal performance and experience, we recommend using the newer v5 and v6 VM series. This ensures you have access to the latest features such as Premium Storage, Accelerated Networking, and Nested Virtualization. While the v6 VM series is preferred, there are certain scenarios where you might want to consider the v5 or even the v4 VM series. Here are some reasons why:

- v6 VMs require enabling NVMe which means that you must have a supported OS.
- v6 VMs support Generation 2 VMs only.
- v6 VMs require MANA (Microsoft Azure Network Adapter) and a MANA supported operating system.
- v6 VMs may not have available capacity in the regions and zones you need.

Note that Lsv4 and Lasv4 series are the latest generation L-series VMs.

Use the Azure VM size documentation to help identify suitable VM sizes.

Migration Steps

Optional: For Reserved Instance (RI) customers only

- Review your current reservations using the Azure Reservation Management page.
- If applicable, exchange existing reservations for newer VM series or trade in your reservations for an **Azure Savings Plan for compute**.
- **HBv2 customers**: 1-year and 3-year HBv2 Reserved Instance purchases ended on April 2, 2026. If you have active HBv2 RIs, consider exchanging them for supported HPC series RIs (such as HBv3, HBv4, HBv5, or HX) or trading them in for an Azure Savings Plan for compute before the retirement date of May 31, 2027.

Identify the Target VM Size

- Evaluate your current VM's workload and performance requirements.

- Select a comparable size from the above table that meets your CPU, memory, and storage needs.

Check and Request Quota Increases

- Before resizing, verify that your subscription has sufficient quota for the target v6 VM series.
- Request more quota through the Azure portal if needed.

Resize the Virtual Machine

You can resize your VM through the Azure portal, Azure CLI, or PowerShell. Follow these steps:

1. **Stop (deallocate) the VM**.
2. **Resize** the VM to your selected v6 series.
3. **Start the VM** after resizing.

Refer to the full Azure VM resizing guide for more detailed instructions.

FAQ

Q: Which sizes are being retired?

To review retired sizes, see retired Azure VM sizes. View retired isolated sizes at Isolation for VMs in Azure.

> **Note**
>
> HPC HBv2-series VMs are also retiring. See the HBv2 row in the table below for the applicable dates.

Expand table

VM Series	3 YR RI expiration date	1 YR RI expiration date	Retirement Date
D	05/01/2025	05/01/2027	05/01/2028
Ds	05/01/2025	05/01/2027	05/01/2028
Dv2	05/01/2025	05/01/2027	05/01/2028
Dsv2	05/01/2025	05/01/2027	05/01/2028

VM Series	3 YR RI expiration date	1 YR RI expiration date	Retirement Date
Ls	05/01/2025	05/01/2027	05/01/2028
F	11/15/2025	11/15/2027	11/15/2028
Fs	11/15/2025	11/15/2027	11/15/2028
Fsv2	11/15/2025	11/15/2027	11/15/2028
Lsv2	11/15/2025	11/15/2027	11/15/2028
G	11/15/2025	11/15/2027	11/15/2028
Gs	11/15/2025	11/15/2027	11/15/2028
Av2	11/15/2025	11/15/2027	11/15/2028
Amv2	11/15/2025	11/15/2027	11/15/2028
B-series	11/15/2025	11/15/2027	11/15/2028
Bv1	11/15/2025	11/15/2027	11/15/2028
HBv2	04/02/2026	04/02/2026	05/31/2027

Q: Why should I migrate my VM?

Migration is mandatory to avoid unexpected shutdown. Additionally, migration yields the following benefits:

- **Performance**: Newer VM series offer better price-to-performance ratios.
- **Regional Availability**: The v5 and v6 series has broader regional support across Azure data centers.
- **Future-proofing**: Migrate ahead of the retirement schedule to avoid disruption.

Q: What will happen to my VM if I do not resize my VM to a target size within the retirement timeline?

After retirement, VMs using this size will be deallocated and stop incurring charges. The size is no longer supported or covered by an SLA; in-memory and temporary disk data is lost, but managed disk data is preserved. To resume service, you may resize to a supported size and restart the VM.

Specifically for HBv2-series: after May 31, 2027, HBv2-series VMs (Standard_HB120rs_v2 and derived sizes) will be automatically set to a deallocated state. After that date, HBv2 VMs will

stop working, lose SLA and support, and stop incurring billing charges.

Q: Can I recover my VM after it has been deallocated?

Yes, you can resize and restart your deallocated VM following the Azure VM resizing guide.

Q: Will VM migration disrupt pay-as-you-go or Savings Plan Pricing billing?

No. If you're using pay-as-you-go or a savings plan, migrating to a newer VM type won't disrupt your current billing. The migration process remains seamless with no changes required in your subscription or payment plan.

Q: How can I migrate my VM if I am on Reserved Instances (RIs) with a retired VM?

If you have active Reserved Instances for D, Dv2, Dsv2, or L-series VMs, follow these steps:

Step 1: Review Current Reservations

- Check your active RIs in the Azure portal.

Identify which RIs are expiring or will be affected by the VM retirement.

Step 2: Migrate and Manage Your RIs

Depending on your business needs, consider these options:

1. **Exchange Existing Reservations**:
    - Swap current RIs for a new VM series without any penalties.
    - Refer to the RI Exchange Guide
2. **Trade-In for Savings Plan**:
    - Convert your existing RIs into an **Azure Savings Plan for compute**.
    - This offers flexibility across VM families and regions.
    - Follow the Azure RI Trade-In Tutorial.
3. **Purchase New RIs**:
    - Buy new reservations that align with your new v6 VM series.

- Consider shorter terms (1-year) for flexibility.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 04/14/2026

Migration Guide for GPU Compute Workloads in Azure

Article • 03/17/2025

As more powerful GPUs become available in the marketplace and in Microsoft Azure datacenters, we recommend re-assessing the performance of your workloads and considering migrating to newer GPUs.

For the same reason, as well as to maintain a high-quality and reliable service offering, Azure periodically retires the hardware that powers older VM sizes. The first group of GPU products to be retired in Azure are the original NC, NC v2 and ND-series VMs, powered by NVIDIA Tesla K80, P100, and P40 datacenter GPU accelerators respectively. These products will be retired on August 31st 2023, and the oldest VMs in this series launched in 2016.

Since then, GPUs have made incredible strides alongside the entire deep learning and HPC industry, typically exceeding a doubling in performance between generations. Since the launch of NVIDIA K80, P40, and P100 GPUs, Azure has shipped multiple newer generations and categories of VM products geared at GPU-accelerated compute and AI, based around NVIDIA's T4, V100, and A100 GPUs, and differentiated by optional features such as InfiniBand-based interconnect fabrics. These are all options we encourage customers to explore as migration paths.

In most cases, the dramatic increase in performance offered by newer generations of GPUs lowers overall TCO by decreasing the duration of job, for burstable jobs- or reducing the quantity of overall GPU-enabled VMs required to cover a fixed-size demand for compute resources, even though costs per GPU-hour may vary. In addition to these benefits, customers may improve Time-to-Solution via higher-performing VMs, and improve the health and supportability of their solution by adopting newer software, CUDA runtime, and driver versions.

Migration vs. Optimization

Azure recognizes that customers have a multitude of requirements that may dictate the selection of a specific GPU VM product, including GPU architectural considerations, interconnects, TCO, Time to Solution, and regional availability based on compliance locality or latency requirements- and some of these even change over time.

At the same time, GPU acceleration is a new and rapidly evolving area.

Thus, there is no true one-size fits-all guidance for this product area, and a migration is a perfect time to re-evaluate potentially dramatic changes to a workload- like moving from a clustered deployment model to a single large 8-GPU VM or vice versa, leveraging reduced precision datatypes, adopting features like Multi-Instance GPU, and much more.

These sorts of considerations- when made the context of already dramatic per-generation GPU performance increases, where a feature such as the addition of TensorCores can increase performance by an order of magnitude, are extremely workload-specific.

Combining migration with application re-architecture can yield immense value and improvement in cost and time-to-solution.

However, these sorts of improvements are beyond the scope of this document, which aims to focus on direct equivalency classes for generalized workloads that may be run by customers today, to identify the most similar VM options in both price *and* performance per GPU to existing VM families undergoing retirement.

Thus, this document assumes that the user may not have any insight or control over workload-specific properties like the number of required VM instances, GPUs, interconnects, and more.

Recommended Upgrade Paths

NC-Series VMs featuring NVIDIA K80 GPUs

The NC (v1)-Series VMs are Azure's oldest GPU-accelerated compute VM type, powered by 1 to 4 NVIDIA Tesla K80 datacenter GPU accelerators paired with Intel Xeon E5-2690 v3 (Haswell) processors. Once a flagship VM type for demanding AI, ML, and HPC applications, they remained a popular choice late into the product lifecycle (particularly via NC-series promotional pricing) for users who valued having a very low absolute cost per GPU-hour over GPUs with higher throughput-per-dollar.

Today, given the relatively low compute performance of the aging NVIDIA K80 GPU platform, in comparison to VM series featuring newer GPUs, a popular use case for the NC-series is real-time inference and analytics workloads, where an accelerated VM must be available in a steady state to serve request from applications as they arrive. In these cases the volume or batch size of requests may be insufficient to benefit from more performant GPUs. NC VMs are also popular for developers and students learning about, developing for, or experimenting with GPU acceleration, who need an inexpensive

cloud-based CUDA deployment target upon which to iterate that doesn't need to perform to production levels.

In general, NC-Series customers should consider moving directly across from NC sizes to NC T4 v3 sizes, Azure's new GPU-accelerated platform for light workloads powered by NVIDIA Tesla T4 GPUs.

Expand table

Current VM Size	Target VM Size	Difference in Specification
Standard_NC6 Standard_NC6_Promo	Standard_NC4as_T4_v3 or Standard_NC8as_T4	CPU: Intel Haswell vs AMD Rome GPU count: 1 (same) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 4 (-2) or 8 (+2) Memory GiB: 16 (-40) or 56 (same) Temp Storage (SSD) GiB: 180 (-160) or 360 (+20) Max data disks: 8 (-4) or 16 (+4) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC12 Standard_NC12_Promo	Standard_NC16as_T4_v3	CPU: Intel Haswell vs AMD Rome GPU count: 1 (-1) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 16 (+4) Memory GiB: 110 (-2) Temp Storage (SSD) GiB: 360 (-320) Max data disks: 48 (+16) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC24 Standard_NC24_Promo	Standard_NC64as_T4_v3*	CPU: Intel Haswell vs AMD Rome GPU count: 4 (same) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 64 (+40) Memory GiB: 440 (+216) Temp Storage (SSD) GiB: 2880 (+1440) Max data disks: 32 (-32) Accelerated Networking: Yes (+) Premium Storage: Yes (+)

Current VM Size	Target VM Size	Difference in Specification
Standard_NC24r Standard_NC24r_Promo	Standard_NC64as_T4_v3*	CPU: Intel Haswell vs AMD Rome GPU count: 4 (same) GPU generation: NVIDIA Keppler vs. Turing (+2 generations, ~2x FP32 FLOPs) GPU memory (GiB per GPU): 16 (+4) vCPU: 64 (+40) Memory GiB: 440 (+216) Temp Storage (SSD) GiB: 2880 (+1440) Max data disks: 32 (-32) Accelerated Networking: Yes (+) Premium Storage: Yes (+) InfiniBand interconnect: No

NC v2-Series VMs featuring NVIDIA Tesla P100 GPUs

The NC v2-series virtual machines are a flagship platform originally designed for AI and Deep Learning workloads. They offered excellent performance for Deep Learning training, with per-GPU performance roughly 2x that of the original NC-Series and are powered by NVIDIA Tesla P100 GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. Like the NC and ND -Series, the NC v2-Series offers a configuration with a secondary low-latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

In general, NCv2-Series customers should consider moving directly across to NC A100 v4 sizes, Azure's new GPU-accelerated platform powered by NVIDIA Ampere A100 PCIe GPUs.

Expand table

Current VM Size	Target VM Size	Difference in Specification
Standard_NC6s_v2	Standard_NC24ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 1 (same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generation) GPU memory (GiB per GPU): 80 (+64) vCPU: 24 (+18) Memory GiB: 220 (+108) Temp Storage (SSD) GiB: 1123 (+387) Max data disks: 12 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)

Current VM Size	Target VM Size	Difference in Specification
Standard_NC12s_v2	Standard_NC48ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 2 (same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generations) GPU memory (GiB per GPU): 80 (+64) vCPU: 48 (+36) Memory GiB: 440 (+216) Temp Storage (SSD) GiB: 2246 (+772) Max data disks: 24 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC24s_v2	Standard_NC96ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 4 (same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generations) GPU memory (GiB per GPU): 80 (+64) vCPU: 96 (+72) Memory GiB: 880 (+432) Temp Storage (SSD) GiB: 4492 (+1544) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_NC24rs_v2	Standard_NC96ads_A100_v4	CPU: Intel Broadwell vs AMD Milan GPU count: 4 (Same) GPU generation: NVIDIA Pascal vs. Ampere (+2 generations) GPU memory (GiB per GPU): 80 (+64) vCPU: 96 (+72) Memory GiB: 880 (+432) Temp Storage (SSD) GiB: 4492 (+1544) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+) InfiniBand interconnect: No (-)

ND-Series VMs featuring NVIDIA Tesla P40 GPUs

The ND-series virtual machines are a midrange platform originally designed for AI and Deep Learning workloads. They offered excellent performance for batch inferencing via improved single-precision floating point operations over their predecessors and are powered by NVIDIA Tesla P40 GPUs and Intel Xeon E5-2690 v4 (Broadwell) CPUs. Like the NC and NC v2-Series, the ND-Series offers a configuration with a secondary low-

latency, high-throughput network through RDMA, and InfiniBand connectivity so you can run large-scale training jobs spanning many GPUs.

Expand table

Current VM Size	Target VM Size	Difference in Specification
Standard_ND6	Standard_NC4as_T4_v3 or Standard_NC8as_T4_v3	CPU: Intel Broadwell vs AMD Rome GPU count: 1 (same) GPU generation: NVIDIA Pascal vs. Turing (+1 generation) GPU memory (GiB per GPU): 16 (-8) vCPU: 4 (-2) or 8 (+2) Memory GiB: 16 (-40) or 56 (-56) Temp Storage (SSD) GiB: 180 (-552) or 360 (-372) Max data disks: 8 (-4) or 16 (+4) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_ND12	Standard_NC16as_T4_v3	CPU: Intel Broadwell vs AMD Rome GPU count: 1 (-1) GPU generation: NVIDIA Pascal vs. Turing (+1 generations) GPU memory (GiB per GPU): 16 (-8) vCPU: 16 (+4) Memory GiB: 110 (-114) Temp Storage (SSD) GiB: 360 (-1,114) Max data disks: 48 (+16) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_ND24	Standard_NC64as_T4_v3*	CPU: Intel Broadwell vs AMD Rome GPU count: 4 (same) GPU generation: NVIDIA Pascal vs. Turing (+1 generations) GPU memory (GiB per GPU): 16 (-8) vCPU: 64 (+40) Memory GiB: 440 (same) Temp Storage (SSD) GiB: 2880 (same) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+)
Standard_ND24r	Standard_ND96amsr_A100_v4	CPU: Intel Broadwell vs AMD Rome GPU count: 8 (+4) GPU generation: NVIDIA Pascal vs. Ampere (+2 generation)

Current VM Size	Target VM Size	Difference in Specification
		GPU memory (GiB per GPU): 80 (+56) vCPU: 96 (+72) Memory GiB: 1900 (+1452) Temp Storage (SSD) GiB: 6400 (+3452) Max data disks: 32 (same) Accelerated Networking: Yes (+) Premium Storage: Yes (+) InfiniBand interconnect: Yes (Same)

Migration Steps

General Changes

1. Choose a series and size for migration. Leverage the pricing calculator for further insights.
2. Get quota for the target VM series
3. Resize the current N* series VM size to the target size. This may also be a good time to update the operating system used by your Virtual Machine image, or adopt one of the HPC images with drivers pre-installed as your starting point.

> **Important**
>
> Your VM image may have been produced with an older version of the CUDA runtime, NVIDIA driver, and (if applicable, for RDMA-enabled sizes only) Mellanox OFED drivers than your new GPU VM series requires, which can be updated by following the instructions in the Azure Documentation.

Breaking Changes

Select target size for migration

After assessing your current usage, decide what type of GPU VM you need. Depending on the workload requirements you have few different choices.

> **Note**

A best practice is to select a VM size based on both cost and performance. The recommendations in this guide are based on a general-purpose, one-to-one comparison of performance metrics and the nearest match in another VM series. Before deciding on the right size, get a cost comparison using the Azure Pricing Calculator.

Important

All legacy NC, NC v2 and ND-Series sizes are available in multi-GPU sizes, including 4-GPU sizes with and without InfiniBand interconnect for scale-out, tightly-coupled workloads that demand more compute power than a single 4-GPU VM, or a single K80, P40, or P100 GPU can supply respectively. Although the recommendations above offer a straightforward path forward, users of these sizes should consider achieving their performance goals with more powerful NVIDIA V100 GPU-based VM series like the NC v3-Series and ND v2-series, which typically enable the same level of workload performance at lower costs and with improved manageability by providing considerably greater performance per GPU and per VM before multi-GPU and multi-node configurations are required, respectively.

Get quota for the target VM family

Follow the guide to request an increase in vCPU quota by VM family. Select the target VM size you have selected for migration.

Resize the current virtual machine

You can resize the virtual machine.

Next steps

For a full list of GPU enabled virtual machine sizes, see GPU - accelerated compute overview

Feedback

Was this page helpful? Yes No

NV series migration guide

> **Important**
>
> The NV-series is retired as of September 6th, 2023, and is no longer available. Please refer to the NV-series retirement page for more information.

As more powerful GPU VM sizes become available in Azure datacenters, assess your workloads and migrate virtual machines (VMs) in the NV and NV_Promo series. These legacy VMs can be migrated into new VM series, such as NVsv3 and NVasv4, for better performance with reduced cost. The NVsv3 VM series is powered by Nvidia M60 GPUs. The NVasv4 series is powered by AMD Radeon Instinct MI25 GPUs.

The main differences between the NV and NV_Promo series and the newer NVsv3 and NVasv4 series are:

- Improved performance.
- Support for Premium storage.
- The option to choose from a fractional GPU size to multi-GPU configurations.

Both the NVsv3 and NVasv4 series have more modern cores and greater capacity.

The following section summarizes the differences between the legacy NV series and the NVsv3 and NVv4 series.

NVsv3 series

The NVv3-series VMs are powered by NVIDIA Tesla M60 GPUs and NVIDIA GRID technology with Intel E5-2690 v4 (Broadwell) CPUs and Intel Hyper-Threading Technology. These VMs are targeted for GPU accelerated graphics applications and virtual desktops where customers want to:

- Visualize their data.
- Simulate results to view.
- Work on CAD.
- Render and stream content.

These VMs can also run single precision workloads, such as encoding and rendering.

NVv3 VMs support Premium storage and come with twice the system memory (RAM) when compared with the NV series. For the most up-to-date specifications, see GPU accelerated compute VM sizes: NVsv3 series.

Expand table

Current VM size	Target VM size	Difference in specification
Standard_NV6 Standard_NV6_Promo	Standard_NV12s_v3	vCPU: 12 (+6) Memory: GiB 112 (+56) Temp storage (SSD) GiB: 320 (-20) Max data disks: 12 (-12) Accelerated networking: Yes Premium storage: Yes
Standard_NV12 Standard_NV12_Promo	Standard_NV24s_v3	vCPU: 24 (+12) Memory: GiB 224 (+112) Temp storage (SSD) GiB: 640 (-40) Max data disks: 24 (-24) Accelerated networking: Yes Premium storage: Yes
Standard_NV24 Standard_NV24_Promo	Standard_NV48s_v3	vCPU: 48 (+24) Memory: GiB 448 (+224) Temp storage (SSD) GiB: 1280 (-160) Max data disks: 32 (-32) Accelerated networking: Yes Premium storage: Yes

NVv4 series

NVv4-series VMs are powered by AMD Radeon Instinct MI25 GPUs and AMD EPYC 7V12 (Rome) CPUs. With the NVv4 series, Azure introduces VMs with partial GPUs. Choose the right size VM for GPU accelerated graphics applications and virtual desktops that start at one-eighth of a GPU with a 2-GiB frame buffer to a full GPU with a 16-GiB frame buffer.

NVv4 VMs currently support only the Windows guest operating system. For the most up-to-date specifications, see GPU accelerated compute VM sizes: NVv4 series.

Expand table

Current VM size	Target VM size	Difference in specification
Standard_NV6 Standard_NV6_Promo	Standard_NV16as_v4	vCPU: 16 (+10) Memory: GiB 56

Current VM size	Target VM size	Difference in specification
		Temp storage (SSD) GiB: 352 (+12) Max data disks: 16 (-8) Accelerated networking: Yes Premium storage: Yes
Standard_NV12 Standard_NV12_Promo	Standard_NV32as_v4	vCPU: 32 (+20) Memory: GiB 112 Temp storage (SSD) GiB: 704 (+24) Max data disks: 32 (+16) Accelerated networking: Yes Premium storage: Yes
Standard_NV24 Standard_NV24_Promo	N/A	N/A

Migration steps for general changes

To deal with general changes:

1. Choose a series and size for migration.
2. Get a quota for the target VM series.
3. Resize the current NV-series VM size to the target size.

If the target size is NVv4, make sure to remove the Nvidia GPU driver and install the AMD GPU driver.

Migration steps for breaking changes

To deal with breaking changes, follow the steps in the next sections.

Select a target size for migration

After you assess your current usage, decide what type of GPU VM you need. Depending on the workload requirements, you have a few different choices. Here's how to choose:

- If the workload is graphics or visualizations and has a hard dependency on using the Nvidia GPU, migrate to the NVsv3 series.
- If the workload is graphics or visualizations and has no hard dependency on a specific type of GPU, migrate to the NVsv3 or NVVasv4 series.

> **Note**
>
> A best practice is to select a VM size based on both cost and performance. The recommendations in this article are based on a one-to-one comparison of performance metrics for the NV and NV_Promo sizes and the nearest match in another VM series. Before you decide on the right size, get a cost comparison by using the Azure Pricing Calculator.

Get a quota for the target VM family

Follow the guide to request an increase in vCPU quota by VM family. Select the NVSv3 series or NVv4 series as the VM family name depending on the target VM size you selected for migration.

Resize the current VM

You can resize the VM.

FAQ

Q: Which GPU driver should I use for the target VM size?

A: For the NVsv3 series, use the Nvidia GRID driver. For NVv4, use the AMD GPU drivers.

Q: I use the Nvidia GPU driver extension today. Will it work for the target VM size?

A: The current Nvidia driver extension will work for NVsv3. Use the AMD GPU driver extensions if the target VM size is NVv4.

Q: Which target VM series should I use if I have dependency on CUDA?

A: NVv3 supports CUDA. The NVv4 VM series with the AMD GPUs doesn't support CUDA.

Last updated on 07/11/2025

Azure Dedicated Host SKU Retirement Migration Guide

As hardware ages, it must be retired and workloads must be migrated to newer, faster, and more efficient Azure Dedicated Host SKUs. The legacy Dedicated Host SKUs should be migrated to newer Dedicated Host SKUs. The main differences between the retiring Dedicated Host SKUs and the newly recommended Dedicated Host SKUs are:

- Newer, more efficient processors
- Increased RAM
- Increased available vCPUs
- Greater regional capacity compared to the retiring Dedicated Host SKUs

Review the FAQs before you get started on migration. The next section will go over which Dedicated Host SKUs to migrate to help aid in migration planning and execution.

Host SKUs being retired

Some Azure Dedicated Host SKUs will be retired soon. Refer to the Azure Dedicated Host SKU Retirement documentation to learn more.

Dsv3-Type1 and Dsv3-Type2

The Dsv3-Type1 and Dsv3-Type2 run Dsv3-series VMs, which offer a combination of vCPU, memory, and temporary storage best suited for most general-purpose workloads. We recommend migrating your existing VMs to one of the following Dedicated Host SKUs:

- Dsv3-Type3
- Dsv3-Type4

Note that both the Dsv3-Type3 and Dsv3-Type4 won't be impacted by the 31 March 2023 retirement date. We recommend moving to either the Dsv3-Type3 or Dsv3-Type4 based on regional availability, pricing, and your organization’s needs.

Esv3-Type1 and Esv3-Type2

The Esv3-Type1 and Esv3-Type2 run Esv3-series VMs, which offer a combination of vCPU, memory, and temporary storage best suited for most memory-intensive workloads. We recommend migrating your existing VMs to one of the following Dedicated Host SKUs:

- Esv3-Type3

- Esv3-Type4

Note that both the Esv3-Type3 and Esv3-Type4 won't be impacted by the 31 March 2023 retirement date. We recommend moving to either the Esv3-Type3 or Esv3-Type4 based on regional availability, pricing, and your organization's needs.

Migrating to supported hosts

To migrate your workloads and avoid Dedicated Host SKU retirement, follow the directions for your migration method of choice.

Automatic migration (Resize)

Moving a host and all associated VMs to newer generation hardware can be done through the host resize feature. Resize simplifies the migration process and avoids having to manually create new hosts and move all VMs individually.

Resize limitations:

- Host can only be resized to an ADH within the same VM family. A Dsv3-Type3 host can be resized to Dsv3-Type4 but **not** to an **E**sv3-Type4.
- You can only resize to newer generation of hardware. A Dsv3-Type3 host can be resized to Dsv3-Type4 but **not** Dsv3-Type2.
- Resizing changes the 'Host Asset ID'. The 'Host ID' remains the same.
- The host and all associated VMs become unavailable during the resize operation.

> ⚠ **Warning**
>
> The resize operation causes the loss of any non-persistent data such as temp disk data. Save all your work to persistent data storage before triggering resize.

> ⓘ **Note**
>
> If the source host is already running on the latest hardware, 'Size' page would display an empty list. If you're looking for enhanced performance, consider switching to a different VM family.

Portal

1. Search for and select the host.

2. In the left menu under **Settings**, select **Size**.
3. Once on the size page from the list of SKUs, select the desired SKU to resize to.
4. Selecting a target size from the list would enable **Resize** button on the bottom on the page.
5. Click **Resize**, host's 'Provisioning State' changes from 'Provisioning Succeeded' to 'Updating'
6. Once the resizing is complete, the host's 'Provisioning State' reverts to 'Provisioning Succeeded'

Manual migration

This includes steps for manually placed VMs, automatically placed VMs, and virtual machine scale sets on your Dedicated Hosts:

Manually Placed VMs

1. Choose a target Dedicated Host SKU to migrate to.
2. Ensure you have quota for the VM family associated with the target Dedicated Host SKU in your given region.
3. Provision a new Dedicated Host of the target Dedicated Host SKU in the same Host Group.
4. Stop and deallocate the VM(s) on your old Dedicated Host.
5. Reassign the VM(s) to the target Dedicated Host.
6. Start the VM(s).
7. Delete the old host.

More detailed instructions can be found in the following sections.

> ⓘ **Note**
>
> **Certain sections are different for automatically placed VMs or virtual machine scale set**. These differences will explicitly be called out in the respective steps.

Ensure quota for the target VM family

Be sure that you have enough vCPU quota for the VM family of the Dedicated Host SKU that you'll be using. If you need quota, follow this guide to request an increase in vCPU quota for

your target VM family in your target region. Select the Dsv3-series or Esv3-series as the VM family, depending on the target Dedicated Host SKU.

Create a new Dedicated Host

Within the same Host Group as the existing Dedicated Host, create a Dedicated Host of the target Dedicated Host SKU.

Stop the VM(s) or virtual machine scale set

PowerShell

Refer to the PowerShell documentation to stop a VM through PowerShell or stop a virtual machine scale set through PowerShell.

Reassign the VM(s) to the target Dedicated Host

> ⓘ Note
>
> **Skip this step for automatically placed VMs and virtual machine scale set.**

Once the target Dedicated Host has been created and the VM has been stopped, reassign the VM to the target Dedicated Host.

Start the VM(s) or virtual machine scale set

> ⓘ Note
>
> **Automatically placed VM(s) and virtual machine scale set require that you delete the old host *before* starting the autoplaced VM(s) or virtual machine scale set.**

PowerShell

Refer to the PowerShell documentation to start a VM through PowerShell or start a virtual machine scale set through PowerShell.

Delete the old Dedicated Host

Once all VMs have been migrated from your old Dedicated Host to the target Dedicated Host, delete the old Dedicated Host.

Help and support

If you have questions, ask community experts in Microsoft Q&A.

Last updated on 03/17/2025

Use infrastructure automation tools with virtual machines in Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets ✔ Uniform scale sets

To create and manage Azure virtual machines (VMs) in a consistent manner at scale, some form of automation is typically desired. There are many tools and solutions that allow you to automate the complete Azure infrastructure deployment and management lifecycle. This article introduces some of the infrastructure automation tools that you can use in Azure. These tools commonly fit in to one of the following approaches:

- Automate the configuration of VMs
  - Tools include Ansible, Chef, Puppet, Bicep, and Azure Resource Manager template.
  - Tools specific to VM customization include cloud-init for Linux VMs, PowerShell Desired State Configuration (DSC), and the Azure Custom Script Extension for all Azure VMs.
- Automate infrastructure management
  - Tools include Packer to automate custom VM image builds, and Terraform to automate the infrastructure build process.
  - Azure Automation can perform actions across your Azure and on-premises infrastructure.
- Automate application deployment and delivery
  - Examples include Azure DevOps Services and Jenkins.

Terraform

Terraform is an automation tool that allows you to define and create an entire Azure infrastructure with a single template format language - the HashiCorp Configuration Language (HCL). With Terraform, you define templates that automate the process to create network, storage, and VM resources for a given application solution. You can use your existing Terraform templates for other platforms with Azure to ensure consistency and simplify the infrastructure deployment without needing to convert to an Azure Resource Manager template.

Learn how to:

- Install and configure Terraform with Azure.
- Create an Azure infrastructure with Terraform.

Azure Automation

Azure Automation uses runbooks to process a set of tasks on the VMs you target. Azure Automation is used to manage existing VMs rather than to create an infrastructure. Azure Automation can run across both Linux and Windows VMs, and on-premises virtual or physical machines with a hybrid runbook worker. Runbooks can be stored in a source control repository, such as GitHub. These runbooks can then run manually or on a defined schedule.

Azure Automation also provides a Desired State Configuration (DSC) service that allows you to create definitions for how a given set of VMs should be configured. DSC then ensures that the required configuration is applied and the VM stays consistent. Azure Automation DSC runs on both Windows and Linux machines.

Learn how to:

- Create a PowerShell runbook.
- Use Hybrid Runbook Worker to manage on-premises resources.
- Use Azure Automation DSC.

Azure DevOps Services

Azure DevOps Services is a suite of tools that help you share and track code, use automated builds, and create a complete continuous integration and development (CI/CD) pipeline. Azure DevOps Services integrates with Visual Studio and other editors to simplify usage. Azure DevOps Services can also create and configure Azure VMs and then deploy code to them.

Learn more about:

- Azure DevOps Services.

Azure Resource Manager template

Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure subscription. You use management features, like access control, locks, and tags, to secure and organize your resources after deployment.

Learn how to:

- Deploy Spot VMs using a Resource Manager template.
- Create a Windows virtual machine from a Resource Manager template.
- Download the template for a VM.
- Create an Azure Image Builder template.

Bicep

Bicep is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. In a Bicep file, you define the infrastructure you want to deploy to Azure, and then use that file throughout the development lifecycle to repeatedly deploy your infrastructure. Your resources are deployed in a consistent manner.

Get started with the Quickstart.

Cloud-init

Cloud-init is a widely used approach to customize a Linux VM as it boots for the first time. You can use cloud-init to install packages and write files, or to configure users and security. Because cloud-init is called during the initial boot process, there are no extra steps or required agents to apply your configuration. For more information on how to properly format your `#cloud-config` files, see the cloud-init documentation site. `#cloud-config` files are text files encoded in base64.

Cloud-init also works across distributions. For example, you don't use **apt-get install** or **yum install** to install a package. Instead you can define a list of packages to install. Cloud-init automatically uses the native package management tool for the distro you select.

We're actively working with our endorsed Linux distro partners in order to have cloud-init enabled images available in the Azure Marketplace. These images make your cloud-init deployments and configurations work seamlessly with VMs and Virtual Machine Scale Sets. Learn more details about cloud-init on Azure:

- Cloud-init support for Linux virtual machines in Azure
- Try a tutorial on automated VM configuration using cloud-init.

PowerShell DSC

PowerShell Desired State Configuration (DSC) is a management platform to define the configuration of target machines. DSC can also be used on Linux through the Open Management Infrastructure (OMI) server.

DSC configurations define what to install on a machine and how to configure the host. A Local Configuration Manager (LCM) engine runs on each target node that processes requested actions based on pushed configurations. A pull server is a web service that runs on a central host to store the DSC configurations and associated resources. The pull server communicates with the LCM engine on each target host to provide the required configurations and report on compliance.

Learn how to:

- Create a basic DSC configuration.
- Configure a DSC pull server.
- Use DSC for Linux.

Azure Custom Script Extension

The Azure Custom Script Extension for Linux or Windows downloads and executes scripts on Azure VMs. You can use the extension when you create a VM, or anytime after the VM is in use.

Scripts can be downloaded from Azure storage or any public location such as a GitHub repository. With the Custom Script Extension, you can write scripts in any language that runs on the source VM. These scripts can be used to install applications or configure the VM as desired. To secure credentials, sensitive information such as passwords can be stored in a protected configuration. These credentials are only decrypted inside the VM.

Learn how to:

- Create a Linux VM with the Azure CLI and use the Custom Script Extension.
- Create a Windows VM with Azure PowerShell and use the Custom Script Extension.

Ansible

Ansible is an automation engine for configuration management, VM creation, or application deployment. Ansible uses an agent-less model, typically with SSH keys, to authenticate and manage target machines. Configuration tasks are defined in playbooks, with several Ansible modules available to carry out specific tasks. For more information, see How Ansible works.

Learn how to:

- Install and configure Ansible on Linux for use with Azure.
- Create a Linux virtual machine.
- Manage a Linux virtual machine.

Chef

Chef is an automation platform that helps define how your infrastructure is configured, deployed, and managed. Some components include Chef Habitat for application lifecycle automation rather than the infrastructure, and Chef InSpec that helps automate compliance with security and policy requirements. Chef Clients are installed on target machines, with one or more central Chef Servers that store and manage the configurations. For more information, see An Overview of Chef.

Learn how to:

- Deploy Chef Automate from the Azure Marketplace.
- Install Chef on Windows and create Azure VMs.

Puppet

Puppet is an enterprise-ready automation platform that handles the application delivery and deployment process. Agents are installed on target machines to allow Puppet Master to run manifests that define the desired configuration of the Azure infrastructure and VMs. Puppet can integrate with other solutions such as Jenkins and GitHub for an improved devops workflow. For more information, see How Puppet works.

Learn how to:

- Deploy Puppet.

Packer

Packer automates the build process when you create a custom VM image in Azure. You use Packer to define the OS and run post-configuration scripts that customize the VM for your specific needs. Once configured, the VM is then captured as a Managed Disk image. Packer automates the process to create the source VM, network and storage resources, run configuration scripts, and then create the VM image.

Learn how to:

- Use Packer to create a Linux VM image in Azure.
- Use Packer to create a Windows VM image in Azure.

Jenkins

Jenkins is a continuous integration server that helps deploy and test applications, and create automated pipelines for code delivery. There are hundreds of plugins to extend the core Jenkins platform, and you can also integrate with many other products and solutions through webhooks. You can manually install Jenkins on an Azure VM, run Jenkins from within a Docker container, or use a pre-built Azure Marketplace image.

Learn how to:

- Create a development infrastructure on a Linux VM in Azure with Jenkins, GitHub, and Docker.

Next steps

There are many different options to use infrastructure automation tools in Azure. You have the freedom to use the solution that best fits your needs and environment. To get started and try some of the tools built-in to Azure, see how to automate the customization of a Linux or Windows VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

User Data for Azure Virtual Machine

Article • 08/22/2024

User data allows you to pass your own scripts or metadata to your virtual machine.

What is "user data"

User data is a set of scripts or other metadata that's inserted to an Azure virtual machine at provision time. Any application on the virtual machine can access the user data from the Azure Instance Metadata Service (IMDS) after provision.

User data is a new version of custom data and it offers added benefits:

- User data can be retrieved from Azure Instance Metadata Service(IMDS) after provision.
- User data is persistent. It will be available during the lifetime of the VM.
- User data can be updated from outside the VM, without stopping or rebooting the VM.
- User data can be queried via GET VM/VMSS API with $expand option.

In addition, if user data isn't added at provision time, you can still add it after provision.

Security warning

> ⚠ **Warning**
>
> User data will not be encrypted, and any process on the VM can query this data. You should not store confidential information in user data.

Make sure you get the latest Azure Resource Manager API to use the new user data features. The contents should be base64 encoded before passed to the API. The size can't exceed 64 KB.

Create user data for Azure VM/VMSS

Adding user data when creating new VM

Use this Azure Resource Manager template to create a new VM with user data. If you're using rest API, for single VMs, add 'UserData' to the "properties" section with the

PUT request to create the VM.

JSON

```
{
  "name": "testVM",
  "location": "West US",
  "properties": {
    "hardwareProfile": {
      "vmSize": "Standard_A1"
    },
    "storageProfile": {
      "osDisk": {
        "osType": "Windows",
        "name": "osDisk",
        "createOption": "Attach",
        "vhd": {
          "uri":
"http://myaccount.blob.core.windows.net/container/directory/blob.vhd"
        }
      }
    },
    "userData": "c2FtcGxlIHVzZXJEYXRh",
    "networkProfile": { "networkInterfaces" : [ { "name" : "nic1" } ] },
  }
}
```

Adding user data when you create new Virtual Machine Scale Set

Using rest API, add 'UserData' to the "virtualMachineProfile" section with the PUT request when creating the Virtual Machine Scale Set.

JSON

```
{
  "location": "West US",
  "sku": {
    "name": "Standard_A1",
    "capacity": 1
  },
  "properties": {
    "upgradePolicy": {
      "mode": "Automatic"
    },
    "virtualMachineProfile": {
      "userData": "VXNlckRhdGE=",
      "osProfile": {
        "computerNamePrefix": "TestVM",
        "adminUsername": "TestUserName",
        "windowsConfiguration": {
          "provisionVMAgent": true,
          "timeZone": "Dateline Standard Time"
```

```
        }
      },
      "storageProfile": {
        "osDisk": {
          "createOption": "FromImage",
          "caching": "ReadOnly"
        },
        "imageReference": {
          "publisher": "publisher",
          "offer": "offer",
          "sku": "sku",
          "version": "1.2.3"
        }
      },
      "networkProfile": {"networkInterfaceConfigurations":
[{"name":"nicconfig1","properties":{"ipConfigurations":
[{"name":"ip1","properties":{"subnet":{"id":"vmssSubnet0"}}}]}}]},
      "diagnosticsProfile": {
        "bootDiagnostics": {
          "enabled": true,
          "storageUri": "https://crputest.blob.core.windows.net"
        }
      }
    },
    "provisioningState": 0,
    "overprovision": false,
    "uniqueId": "00000000-0000-0000-0000-000000000000"
  }
}
```

Retrieving user data

Applications running inside the VM can retrieve user data from IMDS endpoint. For details, see IMDS sample code here.

Customers can retrieve existing value of user data via rest API using $expand=userData endpoint (request body can be left empty).

Single VMs:

`GET "/subscriptions/{guid}/resourceGroups/{RGName}/providers/Microsoft.Compute/virtualMachines/{VMName}?$expand=userData"`

Virtual Machine Scale Set:

`GET "/subscriptions/{guid}/resourceGroups/{RGName}/providers/Microsoft.Compute/virtualM`

`achineScaleSets/{VMSSName}?$expand=userData"`

Virtual Machine Scale Set VM:

`GET "/subscriptions/{guid}/resourceGroups/{RGName}/providers/Microsoft.Compute/virtualMachineScaleSets/{VMSSName}/virtualmachines/{vmss instance id}?$expand=userData"`

Updating user data

With REST API, you can use a normal PUT or PATCH request to update the user data. The user data is updated without the need to stop or reboot the VM.

`PUT "/subscriptions/{guid}/resourceGroups/{RGName}/providers/Microsoft.Compute/virtualMachines/{VMName}`

`PATCH "/subscriptions/{guid}/resourceGroups/{RGName}/providers/Microsoft.Compute/virtualMachines/{VMName}`

The VM.Properties in these requests should contain your desired UserData field, like this:

JSON

```
"properties": {
        "hardwareProfile": {
          "vmSize": "Standard_D1_v2"
        },
        "storageProfile": {
          "imageReference": {
            "sku": "2016-Datacenter",
            "publisher": "MicrosoftWindowsServer",
            "version": "latest",
            "offer": "WindowsServer"
          },
          "osDisk": {
            "caching": "ReadWrite",
            "managedDisk": {
              "storageAccountType": "StandardSSD_LRS"
            },
            "name": "vmOSdisk",
            "createOption": "FromImage"
          }
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "/subscriptions/{subscription-
id}/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkInterf
aces/{existing-nic-name}",
```

```
              "properties": {
                "primary": true
              }
            }
          ]
        },
        "osProfile": {
          "adminUsername": "{your-username}",
          "computerName": "{vm-name}",
          "adminPassword": "{your-password}"
        },
        "diagnosticsProfile": {
          "bootDiagnostics": {
            "storageUri": "http://{existing-storage-account-
name}.blob.core.windows.net",
            "enabled": true
          }
        },
        "userData": "U29tZSBDdXN0b20gRGF0YQ=="
      }
```

> ⓘ **Note**
>
> If you pass in an empty string for "userData" in this case, the user data is deleted.

User data and custom data

Custom data continues to work the same way as today. Note you can't retrieve custom data from IMDS.

Adding user data to an existing VM

If you have an existing VM/VMSS without user data, you can still add user data to this VM by using the updating commands, as described in the "Updating the User data" section. Make sure you upgrade to the latest version of Azure Resource Manger API.

Next steps

Try out Azure Instance Metadata Service, learn how to get the VM instance metadata and user data from its endpoint.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Custom data and cloud-init on Azure Virtual Machines

Article • 09/09/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

You might need to inject a script or other metadata into a Microsoft Azure virtual machine (VM) at provisioning time. In other clouds, this concept is often called *user data*. Microsoft Azure has a similar feature called *custom data*.

Custom data is made available to the VM during first startup or setup, which is called *provisioning*. Provisioning is the process where VM creation parameters (for example, host name, username, password, certificates, custom data, and keys) are made available to the VM. A provisioning agent, such as the Linux Agent or cloud-init, processes those parameters.

Pass custom data to the VM

To use custom data, you must Base64-encode the contents before passing the data to the API--unless you're using a CLI tool that does the conversion for you, such as the Azure CLI. The size can't exceed 64 KB.

In the CLI, you can pass your custom data as a file, as the following example shows. The file is converted to Base64.

Azure CLI

```
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --image Ubuntu2204 \
  --custom-data cloud-init.txt \
  --generate-ssh-keys
```

In Azure Resource Manager, there's a base64 function:

JSON

```
"name": "[parameters('virtualMachineName')]",
"type": "Microsoft.Compute/virtualMachines",
"apiVersion": "2019-07-01",
"location": "[parameters('location')]",
"dependsOn": [
```

```
..],
"variables": {
        "customDataBase64": "[base64(parameters('stringData'))]"
    },
"properties": {
..
    "osProfile": {
        "computerName": "[parameters('virtualMachineName')]",
        "adminUsername": "[parameters('adminUsername')]",
        "adminPassword": "[parameters('adminPassword')]",
        "customData": "[variables('customDataBase64')]"
        },
```

Process custom data

The provisioning agents installed on the VMs handle communication with the platform and placing data on the file system.

Windows

Custom data is placed in *%SYSTEMDRIVE%\AzureData\CustomData.bin* as a binary file, but it isn't processed. If you want to process this file, you need to build a custom image and write code to process *CustomData.bin*.

Linux

On Linux operating systems, custom data is passed to the VM via the *ovf-env.xml* file. That file is copied to the */var/lib/waagent* directory during provisioning. Newer versions of the Linux Agent copy the Base64-encoded data to */var/lib/waagent/CustomData* for convenience.

Azure currently supports two provisioning agents:

- **Linux Agent**. By default, the agent doesn't process custom data. You need to build a custom image with the data enabled. The relevant settings are:
  - `Provisioning.DecodeCustomData`
  - `Provisioning.ExecuteCustomData`

  When you enable custom data and run a script, the virtual machine will not report a successful VM provision until the script has finished executing. If the script exceeds the total VM provisioning time limit of 40 minutes, VM creation fails.

  If the script fails to run, or errors happen during execution, that's not a fatal provisioning failure. You need to create a notification path to alert you for the

completion state of the script.

To troubleshoot custom data execution, review */var/log/waagent.log*.

- **cloud-init**. By default, this agent processes custom data. It accepts multiple formats of custom data, such as cloud-init configuration and scripts.

  Similar to the Linux Agent, if errors happen during execution of the configuration processing or scripts when cloud-init is processing the custom data, that's not a fatal provisioning failure. You need to create a notification path to alert you for the completion state of the script.

  However, unlike the Linux Agent, cloud-init doesn't wait for custom data configurations from the user to finish before reporting to the platform that the VM is ready. For more information on cloud-init on Azure, including troubleshooting, see cloud-init support for virtual machines in Azure.

FAQ

Can I update custom data after the VM has been created?

For single VMs, you can't update custom data in the VM model. But for Virtual Machine Scale Sets, you can update custom data. For more information, see Modify a Scale Set. When you update custom data in the model for a Virtual Machine Scale Set:

- Existing instances in the scale set don't get the updated custom data until they're updated to the latest model and reimaged.
- New instances receive the new custom data.

Can I place sensitive values in custom data?

We advise *not* to store sensitive data in custom data. For more information, see Azure data security and encryption best practices.

Is custom data made available in IMDS?

Custom data isn't available in Azure Instance Metadata Service (IMDS). We suggest using user data in IMDS instead. For more information, see User data through Azure Instance Metadata Service.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Ansible on Azure documentation

Learn how to use Ansible to automate cloud provisioning, configuration management, and application deployments.

About Ansible on Azure

OVERVIEW

About Ansible on Azure

REFERENCE

Releases and features

All modules for Azure

Install and Configure Ansible

QUICKSTART

Configure Ansible using Azure Cloud Shell

Configure Ansible using Azure CLI

TUTORIAL

Configure a dynamic inventory

Manage Virtual Machines

TUTORIAL

Create a Linux VM

Manage a Linux VM

Create a VM from a Shared Image Gallery

Configure VM security using Azure Key Vault

Manage Virtual Machine Scale Sets

TUTORIAL

Configure virtual machine scale sets

Deploy applications to virtual machine scale sets

Automatically scale virtual machine scale sets

Update custom image

Create a from Shared Image Gallery

Manage Virtual Networks

TUTORIAL

Configure peering

Configure route tables

Manage AKS

TUTORIAL

Configure AKS clusters

Configure Azure CNI networking

Configure RBAC roles in AKS cluster

Terraform on Azure documentation

Learn how to use Terraform to reliably provision virtual machines and other infrastructure on Azure.

Getting started

OVERVIEW

About Terraform on Azure

Terraform AzAPI provider

REFERENCE

AzureRM provider versions

Terraform Microsoft Graph provider

ARCHITECTURE

Terraform AzureRM Provider

CONCEPT

Comparing features of Terraform and Bicep

GET STARTED

Install and configure Terraform

Install the Azure Terraform Visual Studio Code extension

Generate Terraform configurations using Azure Copilot

Authenticate Terraform to Azure

Store Terraform state in Azure Storage

Azure Export for Terraform

OVERVIEW

Azure Export for Terraform overview

CONCEPT

Azure Export for Terraform concepts

QUICKSTART

Export your first resources

Export resources to HCL code

HOW-TO GUIDE

Select custom resources

Advanced scenarios

Terraform Export in the Azure portal

OVERVIEW

Resource Provider Overview

QUICKSTART

Export your first Terraform resources in the Azure portal

Popular Terraform Azure service articles

QUICKSTART

Create an Azure resource group using Terraform

Create an AKS cluster

Create a Linux VM

Create a Windows VM

Create an Azure key vault and key using Terraform

Direct web traffic with Azure Application Gateway - Terraform

Create a single database in Azure SQL Database using Terraform

Create an Azure API Management service using Terraform

Create an Azure Front Door Standard/Premium profile using Terraform

Create an Azure Container Instance with a public IP address using Terraform

Networking

QUICKSTART

Deploy with IP Groups - Terraform

Deploy with multiple addresses - Terraform

Deploy with Availability Zones - Terraform

Create a NAT Gateway

Create a private endpoint

Manage infrastructure

QUICKSTART

Create a Linux VM

Create a Linux VM cluster

Create a Windows VM

Create a Windows VM cluster

HOW-TO GUIDE

Provision VM scale set with infrastructure

Provision VM scale set from a Packer custom image

Work with Containers (AKS, Application Gateway, ...)

QUICKSTART

Create an AKS cluster

Create an ACI instance with a public IP address

Deploy Azure Application Gateway v2

HOW-TO GUIDE

Create an Application Gateway ingress controller in AKS

Jenkins on Azure documentation

Learn how to use Jenkins to automate continuous integration and continuous delivery on Azure.

About Jenkins on Azure

OVERVIEW

About Jenkins on Azure

REFERENCE

Jenkins plug-ins for Azure

ARCHITECTURE

Jenkins architecture

Install and configure Jenkins

QUICKSTART

Configure Jenkins using Azure CLI

DOWNLOAD

Sample jobs and scripts

Scale using Jenkins and Azure

TUTORIAL

Scale with Azure Container Instances

Scale with Azure VM agents

CI/CD to Azure Functions

TUTORIAL

Deploy to Azure Functions

CI/CD to AKS

TUTORIAL

Deploy from GitHub to AKS

CI/CD to App Service

TUTORIAL

Create Azure resources in a pipeline job

CI/CD to Azure DevOps

TUTORIAL

Deploy to a Linux VM using Azure DevOps Services

CI/CD to Azure Storage

TUTORIAL

Using Azure Storage with a Jenkins C/I solution

CI/CD to Linux VMs

TUTORIAL

Create a dev infrastructure on a Linux VM

CI/CD to Service Fabric

TUTORIAL

Build and deploy an Azure Service Fabric app

CI/CD to Azure Spring Apps

TUTORIAL

Deploy to Azure Spring Apps using the Azure CLI

Plan to manage costs for virtual machines

Article • 08/22/2024

This article describes how you plan for and manage costs for virtual machines. Before you deploy the service, you can use the Azure pricing calculator to estimate costs for virtual machines. Later, as you deploy Azure resources, review the estimated costs.

If you'd like to learn how to monitor costs for virtual machines, see Monitor costs for virtual machines. If you'd like to review the best practices for virtual machine cost optimization, see Best practices for virtual machine cost optimization.

In this article, you'll:

- Learn how to estimate costs before using virtual machines
- Gain an understanding of the billing model for virtual machines
- Learn how to review costs of virtual machine deployments in the Azure portal

Prerequisites

Cost analysis in Cost Management supports most Azure account types but not all of them. To view the full list of supported account types, see Understand Cost Management data. To view cost data, you need at least read access for an Azure account. For information about assigning access to Microsoft Cost Management data, see Assign access to data.

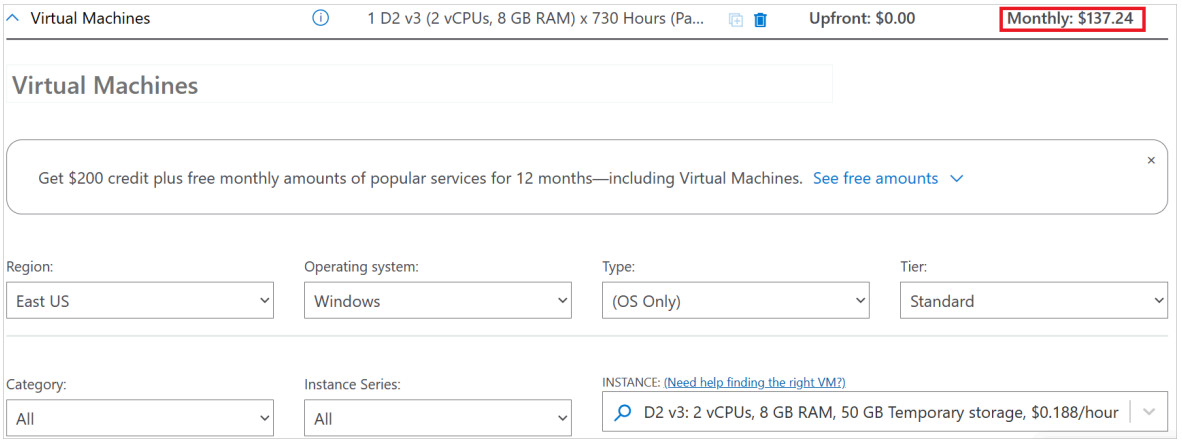
Estimate costs before using virtual machines

Use the Azure pricing calculator to estimate costs before you add virtual machines.

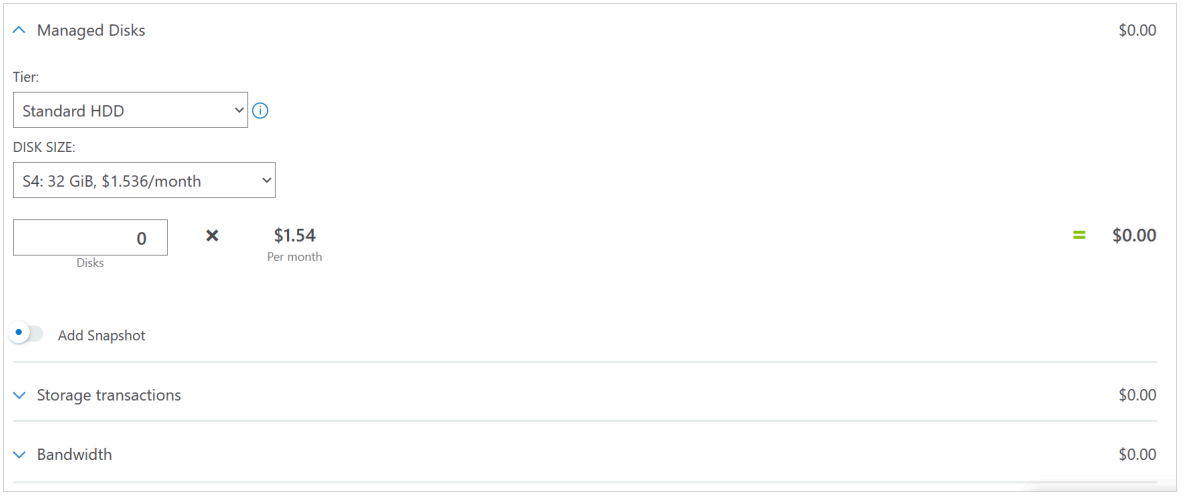
1. On the **Virtual Machines** tile under the **Products** tab, select **Add to estimate** and scroll down to the **Your Estimate** section.

2. Select options from the drop-downs. There are various options available to choose from. The options that have the largest impact on your estimate total are your virtual machine's operating system, the operating system license if applicable, the VM size you select under **INSTANCE**, the number of instances you choose, and the amount of time your month your instances to run.

Notice that the total estimate changes as you select different options. The estimate appears in the upper corner and the bottom of the **Your Estimate** section.



3. Below the **Savings Options** section, there are choices for optional, additional resources you can deploy with your virtual machine, including **Managed Disks**, **Storage transactions**, and **Bandwidth**. For optimal performance, we recommend pairing your virtual machines with managed disks, but make sure to review the additional cost incurred by these resources.



As you add new resources to your workspace, return to this calculator and add the same resource here to update your cost estimates.

For more information, see Azure Virtual Machines pricing for Windows or Linux.

Understand the full billing model for virtual machines

Virtual machines run on Azure infrastructure that accrues costs when you deploy new resources. It's important to understand that there could be other infrastructure costs that might accrue.

How you're charged for virtual machines

When you create or use virtual machines resources, you might get charged for the following meters:

- **Virtual machines** - You're charged for it based on the number of hours per VM.
  - The price also changes based on your VM size.
  - The price also changes based on the region where your virtual machine is located.
  - As virtual machine instances go through different states, they're billed differently.
- **Storage** - Pricing changes based on multiple factors, including disk types, transactions, and configurations.
  - For more information about Disk Storage billing, see Understand Azure Disk Storage billing.
- **Virtual network** - You're charged for it based on the number of GBs of data transferred.
- **Bandwidth** - You're charged for it based on the number of GBs of data transferred.
- **Azure Monitor** - You're charged for it based on the number of GBs of data ingested.
- **Azure Bastion** - You're charged for it based on the number of GBs of data transferred.
- **Azure DNS** - You're charged for it based on the number of DNS zones hosted in Azure and the number of DNS queries received.
- **Load balancer, if used** - You're charged for it based on the number of rulesets, hours used, and GB of data processed.

Any premium software from the Azure Marketplace comes with its own billing meters.

At the end of your billing cycle, the charges for each meter are summed. Your bill or invoice shows a section for all virtual machines costs. There's a separate line item for each meter.

Other costs that might accrue with virtual machines

When you create resources for virtual machines, resources for other Azure services are also created. They include:

- Virtual Network
- Virtual Network Interface Card (NIC)
  - NICs don't incur any costs by themselves. However, your VM size limits how many NICs you can deploy, so play accordingly.

- A private IP and sometimes a public IP
- Network Security Group (NSG)
  - NSGs don't incur any costs.
- OS disk and, optionally, additional disks
- In some cases, a load balancer

For more information, see the Parts of a VM and how they're billed section of the virtual machines documentation overview.

Costs might accrue after resource deletion

After you delete virtual machines resources, the following resources might continue to exist. They continue to accrue costs until you delete them.

- Any disks deployed other than the OS and local disks
  - By default, the OS disk is deleted with the VM, but it can be set not to during the VM's creation
- Virtual network
  - Your virtual NIC and public IP, if applicable, can be set to delete along with your virtual machine
- Bandwidth
- Load balancer

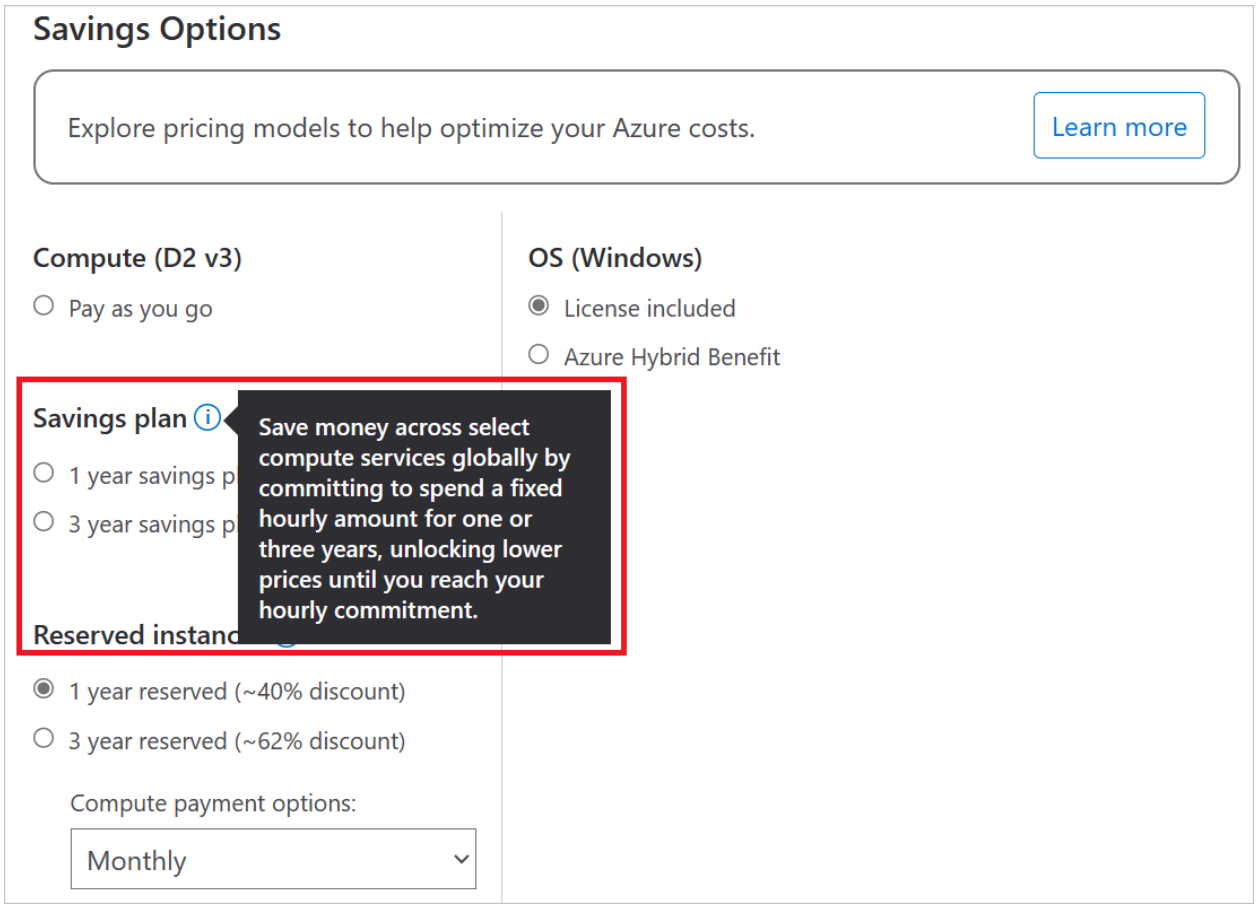
If your OS disk isn't deleted with your VM, it likely incurs P10 disk costs even in a stopped state. The OS disk size is smaller by default for some images and incurs lower costs accordingly.

For virtual networks, one virtual network is billed per subscription and per region. Virtual networks can't span regions or subscriptions. Setting up private endpoints in vNet setups may also incur charges.

Bandwidth is charged by usage; the more data transferred, the more you're charged.

Using a savings plan with virtual machines

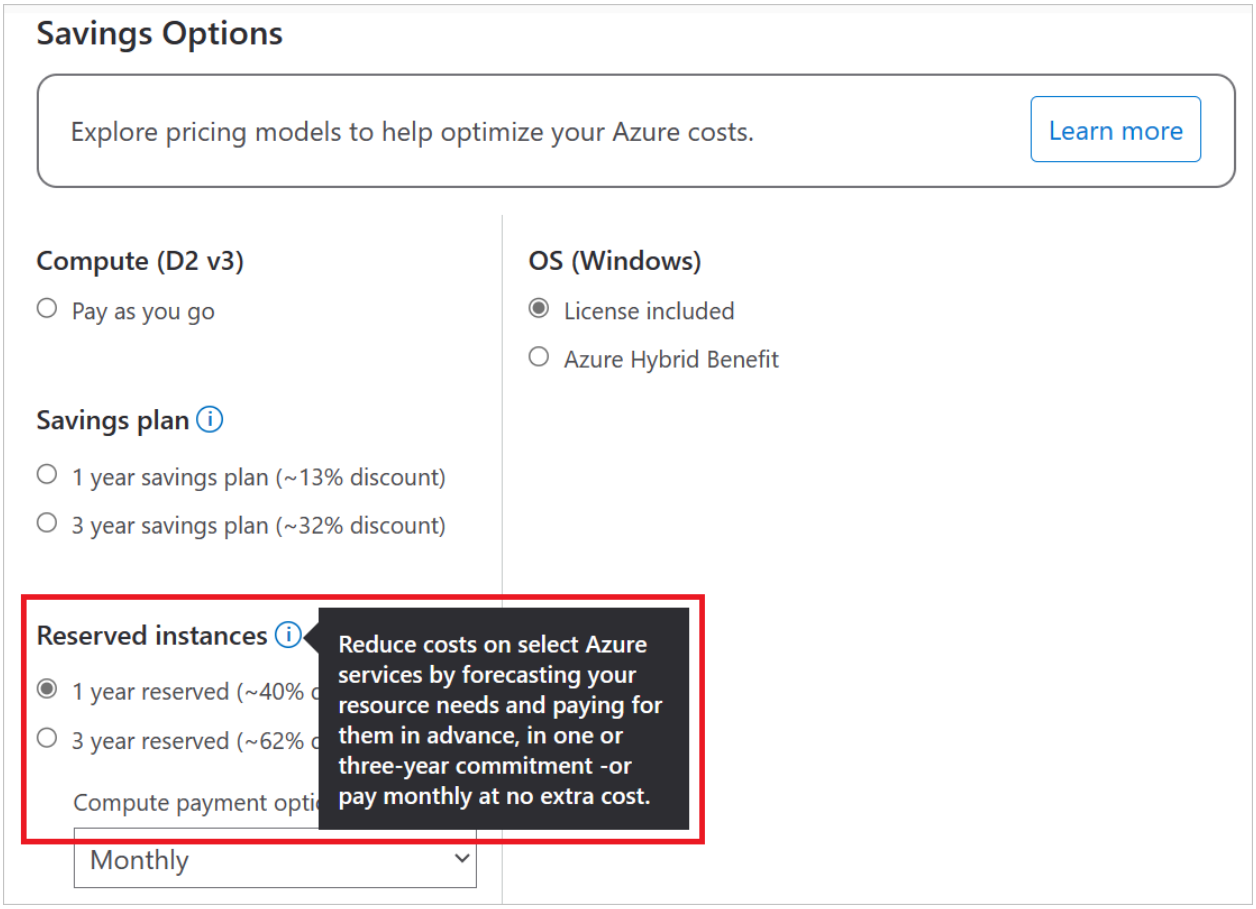
You can choose to spend a fixed hourly amount for your virtual machines, unlocking lower prices until you reach your hourly commitment. These savings plans are available in one- and three-year options.



Using Azure Prepayment with virtual machines

You can pay for virtual machines charges with your Azure Prepayment credit. However, you can't use Azure Prepayment credit to pay for charges for third party products and services including those from the Azure Marketplace.

When you prepay for virtual machines, you're purchasing reserved instances. Committing to a reserved VM instance can save you money. The reservation discount is applied automatically to the number of running virtual machines that match the reservation scope and attributes. Reserved instances are available in one- and three-year plans.



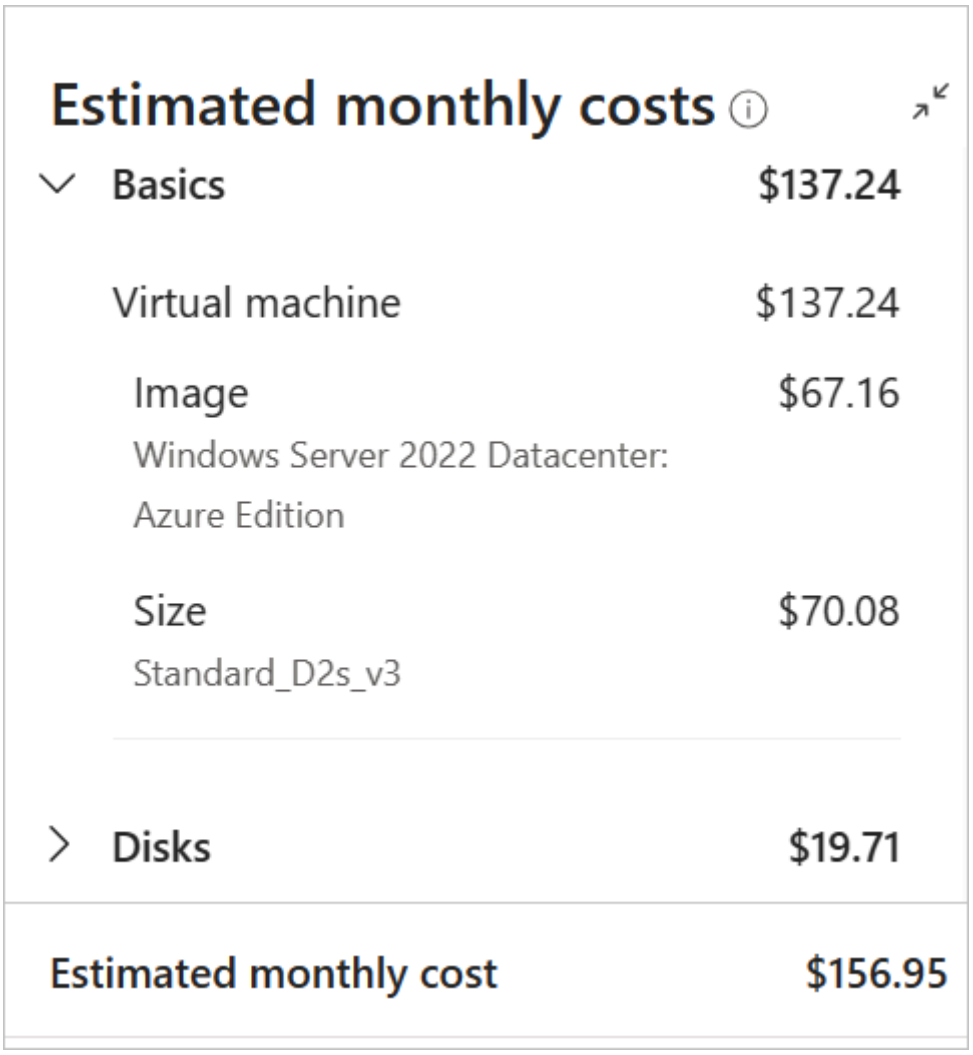
For more information, see Save costs with Azure Reserved VM Instances.

Review estimated costs in the Azure portal

As you create resources for virtual machines, you see estimated costs.

To create a virtual machine and view the estimated price:

1. Sign in to the Azure portal.
2. Enter *virtual machines* in the search.
3. Under **Services**, select **Virtual machines**.
4. In the **Virtual machines** page, select **Create** and then **Azure virtual machine**. The **Create a virtual machine** page opens.
5. On the right side, you see a summary of the estimated costs. Adjust the options in the creation settings to see how the price changes and review the estimated costs.



6. Finish creating the resource.

If your Azure subscription has a spending limit, Azure prevents you from spending over your credit amount. As you create and use Azure resources, your credits are used. If you reach your credit limit, the resources that you deployed are disabled for the rest of that billing period. You can't change your credit limit, but you can remove it. For more information about spending limits, see Azure spending limit.

Next steps

In this article, you learned about estimating costs of virtual machines, the virtual machine billing model, and reviewing virtual machine costs in the Azure portal.

For more information on virtual machine cost optimization, see the following articles:

- Learn how to monitor costs for virtual machines.
- Review the virtual machine cost optimization best practices.
- Learn how to optimize your cloud investment with Microsoft Cost Management.
- Learn more about managing costs with cost analysis.
- Learn about how to prevent unexpected costs.

- Take the Cost Management guided learning course.
- Learn how to create Linux and Windows virtual machines.
- Take the Microsoft Azure Well-Architected Framework - Cost Optimization training.
- Review the Well-Architected Framework cost optimization design principles and how they apply to virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Monitor costs for virtual machines

Article • 08/22/2024

This article describes how you monitor costs for virtual machines.

If you'd like to see how the billing model works for virtual machines and how to plan for costs ahead of resource deployment, see Plan to manage costs. If you'd like to review the best practices for virtual machine cost optimization, see Best practices for virtual machine cost optimization.

After you start using virtual machine resources, use Cost Management features to set budgets and monitor costs. You can also review forecasted costs and identify spending trends to identify areas where you might want to act. Costs for virtual machines are only a portion of the monthly costs in your Azure bill. Although this article explains how to plan for and manage costs for virtual machines, your bill includes the costs of all Azure services and resources used in your Azure subscription, including the third-party services. You can learn more about the billing model in Plan to manage costs.

In this article, you'll learn how to:

- Monitor virtual machine costs
- Create budgets for virtual machines
- Export virtual machine cost data

Prerequisites

Cost analysis in Cost Management supports most Azure account types but not all of them. To view the full list of supported account types, see Understand Cost Management data. To view cost data, you need at least read access for an Azure account. For information about assigning access to Microsoft Cost Management data, see Assign access to data.

Monitor costs

As you use Azure resources with virtual machines, you incur costs. Azure resource usage unit costs vary by time intervals (seconds, minutes, hours, and days) or by unit usage (bytes, megabytes, and so on.) As soon as virtual machine use starts, costs are incurred, and you can see the costs in cost analysis.

When you use cost analysis, you view virtual machine costs in graphs and tables for different time intervals. Some examples are by day, current and prior month, and year.

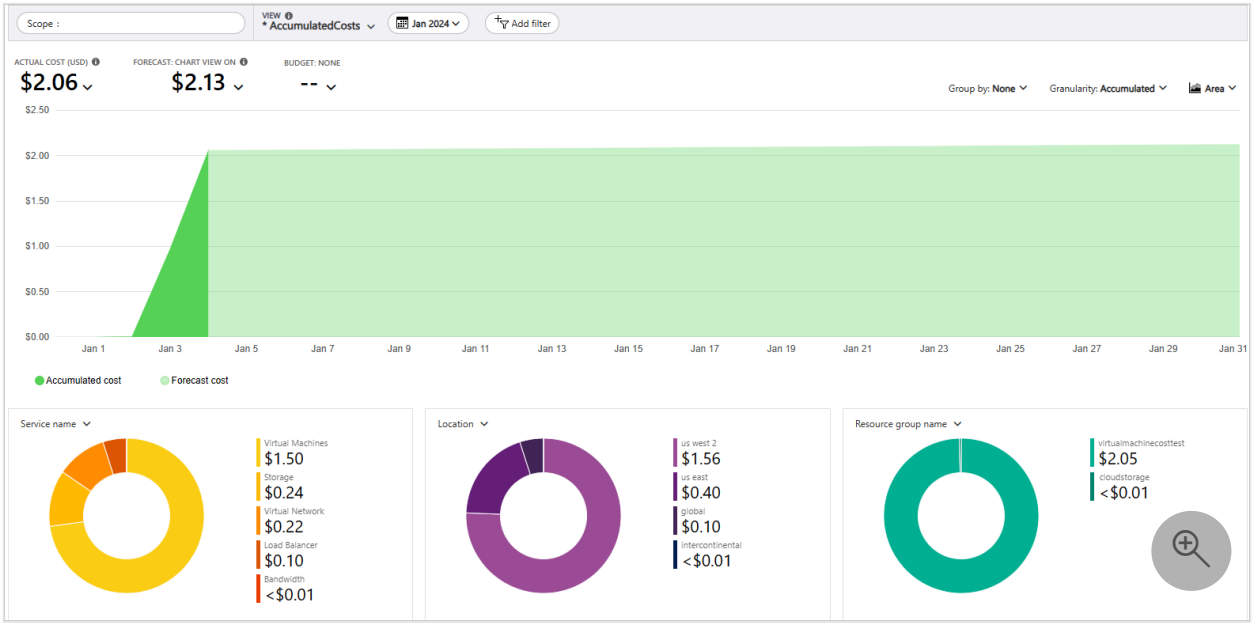
You also view costs against budgets and forecasted costs. Switching to longer views over time can help you identify spending trends, and you can see where you might be overspending. If you create budgets, you can also easily see where they're exceeded.

To view virtual machine costs in cost analysis:

1. Sign in to the Azure portal.
2. Open the scope in the Azure portal and select **Cost analysis** in the menu. For example, go to **Subscriptions**, select a subscription from the list, and then select **Cost analysis** in the menu. Select **Scope** to switch to a different scope in cost analysis.
3. By default, cost for services are shown in the first donut chart. Select the area in the chart labeled virtual machines.

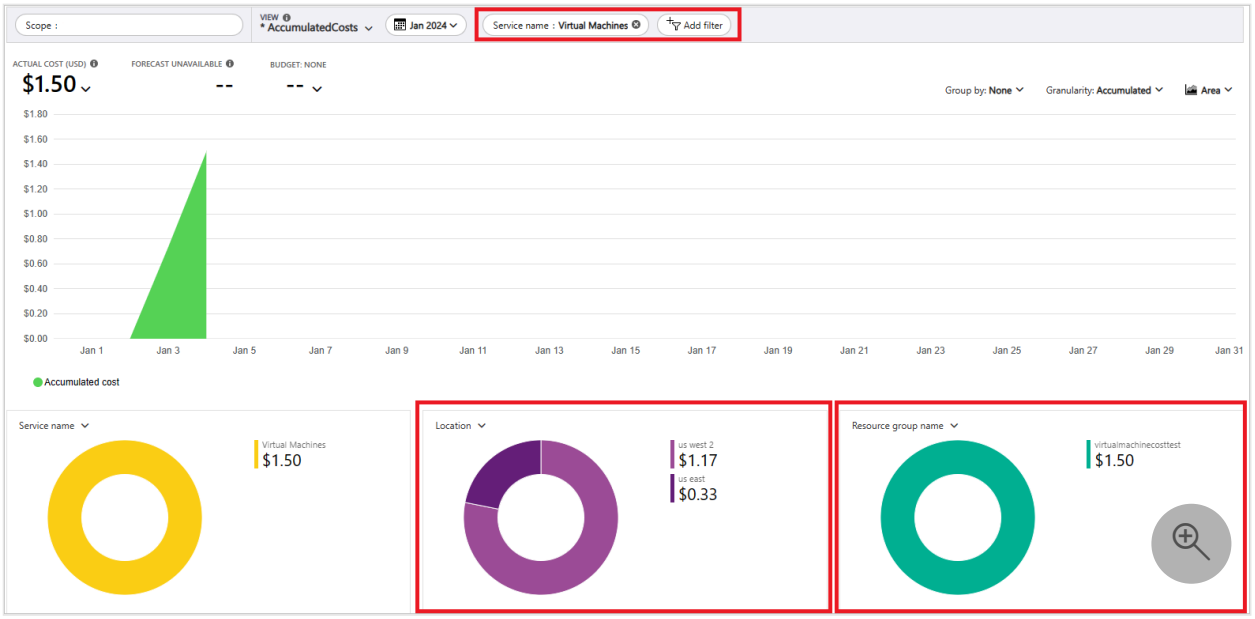
> **Note**
>
> If you just created your virtual machine, cost and usage data is typically only available within 8-24 hours.

Actual monthly costs are shown when you initially open cost analysis. Here's an example showing all monthly usage costs.



To narrow costs for a single service, like virtual machines, select **Add filter** and then select **Service name**. Then, select **Virtual Machines**.

Here's an example showing costs for just virtual machines.



In the preceding example, you see the current cost for the service. Costs by Azure regions (locations) and virtual machines costs by resource group are also shown. From here, you can explore costs on your own.

Create budgets

You can create budgets to manage costs and create alerts that automatically notify stakeholders of spending anomalies and overspending risks. Alerts are based on spending compared to budget and cost thresholds. Budgets and alerts are created for Azure subscriptions and resource groups, so they're useful as part of an overall cost monitoring strategy.

Budgets can be created with filters for specific resources or services in Azure if you want more granularity present in your monitoring. Filters help ensure that you don't accidentally create new resources that cost you more money. For more information about the filter options available when you create a budget, see Group and filter options.

Export cost data

You can also export your cost data to a storage account. This is helpful when you need or others to do additional data analysis for costs. For example, a finance team can analyze the data using Excel or Power BI. You can export your costs on a daily, weekly, or monthly schedule and set a custom date range. Exporting cost data is the recommended way to retrieve cost datasets.

Next steps

In this article, you learned how to monitor virtual machine costs, create budgets, and export cost data.

For more information on virtual machine cost optimization, see the following articles:

- Learn how to plan to manage costs for virtual machines.
- Review the virtual machine cost optimization best practices.
- Learn how to optimize your cloud investment with Microsoft Cost Management.
- Learn more about managing costs with cost analysis.
- Learn about how to prevent unexpected costs.
- Take the Cost Management guided learning course.
- Learn how to create Linux and Windows virtual machines.
- Take the Microsoft Azure Well-Architected Framework - Cost Optimization training.
- Review the Well-Architected Framework cost optimization design principles and how they apply to virtual machines.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Best practices for virtual machine cost optimization

Article • 08/22/2024

This article describes the best practices for managing costs for virtual machines.

If you'd like to see how the billing model works for virtual machines and how to plan for costs ahead of resource deployment, see Plan to manage costs. If you'd like to learn how to monitor costs for virtual machines, see Monitor costs for virtual machines.

In this article, you'll learn:

- Best practices for managing and reducing costs for virtual machines
- How to use Azure policies to manage and reduce costs

Best practices to manage and reduce costs for virtual machines

The following are some best practices you can use to reduce the cost of your virtual machines:

- Use the virtual machines selector to identify the best VMs for your needs
  - For development and test environments:
    - Use B-Series virtual machines
      - Use at least B2 for Windows machines
    - Use HDDs instead of SSDs when you can
    - Use locally redundant storage (LRS) accounts instead of geo- or zone-redundant storage accounts
    - Use Logic Apps or Azure Automation to implement an automatic start and stop schedule for your VMs
  - For production environments:
    - Use the dedicated Standard pricing tier or higher
    - Use a Premium SSD v2 disk and programmatically adjust its performance to account for either higher or lower demand based on your workload patterns
    - For other disk types, size your disks to achieve your desired performance without the need for over-provisioning. Account for fluctuating workload patterns, and minimizing unused provisioned capacity
- Use role-based-access-control (RBAC) to control who can create resources
- Use Azure Spot virtual machines where you can

- For Windows virtual machines, consider Azure Hybrid Benefit for Windows Server to save cost on licensing
- Use cost alerts to monitor usage and spending
- Automate shutdown of underutilized VMs, for example CPU utilization below 10% for 1 hour, by using alerts, action groups, and Automation runbooks
- Minimize idle instances by configuring autoscaling
- Configure Azure Bastion for operational access

Use policies to help manage and reduce costs for virtual machines

You can use Azure Policy to help govern and optimize the costs of your resources.

There are built-in policies for virtual machines and networking services that can help with cost savings:

- **Allowed virtual machine SKUs** - This policy enables you to specify a set of virtual machine size SKUs that your organization can deploy. You could use this policy to restrict any virtual machine sizes that exceed your desired budget. This policy would require updates to maintain as new virtual machine SKUs are added.
  - https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/Compute/VMSkusAllowed_Deny.json
  - You can review the available VM sizes and cross-reference their associated costs on the pricing pages for Windows and Linux.
- **Network interfaces should not have public IPs** - This policy restricts the creation of public IP addresses, except in cases where they are explicitly allowed. Restricting unnecessary exposure to the internet can help reduce bandwidth and virtual network data costs.

You can also make custom policies using Azure Policy. Some examples include:

- Implement policies to restrict what resources can be created:
  - https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/General/AllowedResourceTypes_Deny.json
- Implement policies to not allow certain resources to be created:
  - https://github.com/Azure/azure-policy/blob/master/built-in-policies/policyDefinitions/General/InvalidResourceTypes_Deny.json
- Using the resource policy to limit the allowed locations where virtual machines can be deployed.
- Auditing resources that are incurring costs even after virtual machine deletion.
- Auditing resources to enforce the use of the Azure Hybrid Benefit.

Next steps

In this article, you learned the best practices for managing and reducing costs for virtual machines and how to use Azure policies to manage and reduce costs.

For more information on virtual machine cost optimization, see the following articles:

- Learn how to plan to manage costs for virtual machines.
- Learn how to monitor costs for virtual machines.
- Learn how to optimize your cloud investment with Microsoft Cost Management.
- Learn more about managing costs with cost analysis.
- Learn about how to prevent unexpected costs.
- Take the Cost Management guided learning course.
- Learn how to create Linux and Windows virtual machines.
- Take the Microsoft Azure Well-Architected Framework - Cost Optimization training.
- Review the Well-Architected Framework cost optimization design principles and how they apply to virtual machines.

Feedback

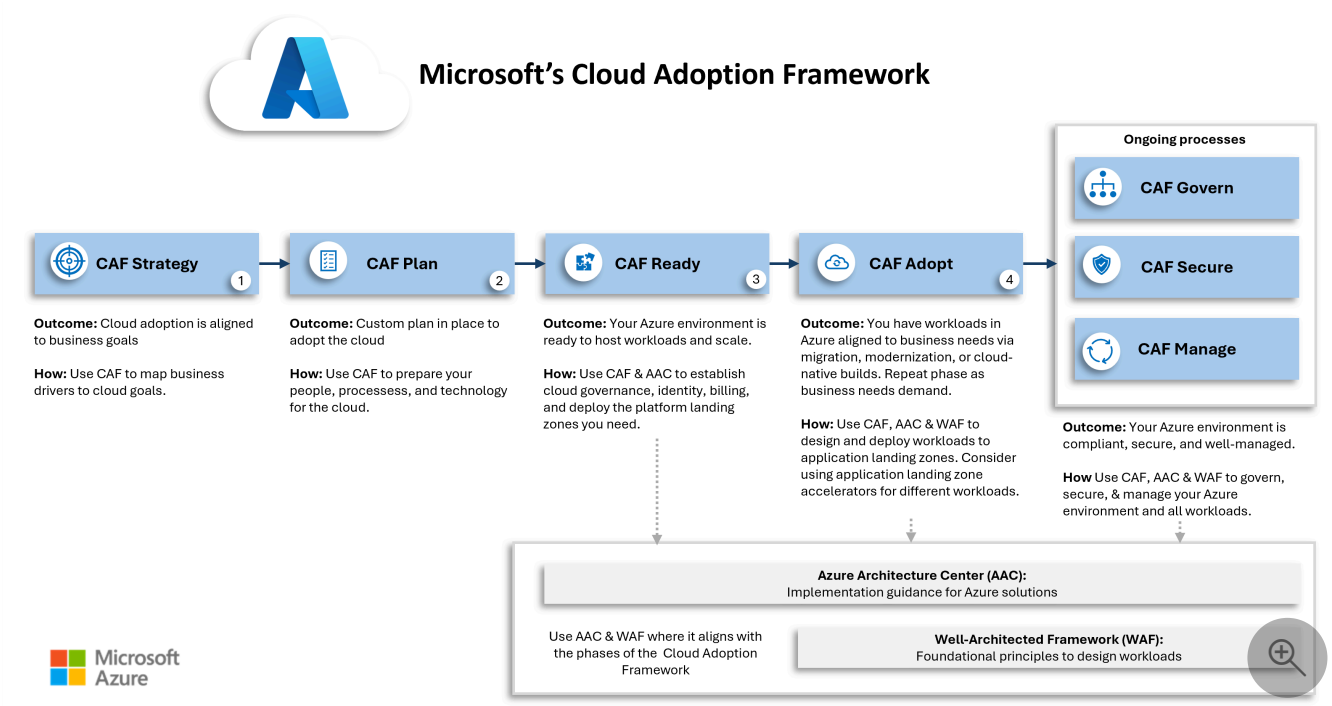
Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What is the Microsoft Cloud Adoption Framework?

Summarize this article for me

The Microsoft Cloud Adoption Framework (CAF) is a structured roadmap that helps organizations successfully adopt Azure and integrate it into their existing IT environments. It provides best practices and guidance to help organizations make the right decisions for their business throughout the cloud journey.



How does the Cloud Adoption Framework work?

The Cloud Adoption Framework organizes Azure guidance into seven core methodologies that together guide an organization's cloud journey. The foundational methodologies (Strategy, Plan, Ready, and Adopt) are sequential. They help you define business outcomes, prepare your organization and environment, and deploy workloads to Azure. These phases are where most organizations start and are revisited as cloud adoption matures. Once workloads are running in Azure, the operational methodologies (Govern, Secure, and Manage) define different aspects of your cloud operations that help ensure your Azure environment remains compliant, protected, and optimized over time.

Expand table

CAF methodology (phases)	Outcome	How
1. Strategy	Cloud adoption aligned to business goals	Map business drivers to cloud outcomes.
2. Plan	A plan for cloud adoption	Operating model, cloud skills, migration plan, and cloud cost estimation.
3. Ready	An Azure environment ready for workloads	Azure purchasing, tenant setup, platform landing zone, application landing zones.
4. Adopt	Workloads in Azure meeting business needs	Migrate, modernize, or build-cloud native workloads.
5. Govern	Control workloads	Assess cloud risks and mitigate with Azure and Microsoft tools.
6. Secure	Protect workloads	Apply security control with with Azure and Microsoft tools.
7. Manage	Optimize workloads	Administer workloads with Azure and Microsoft tools.

Why use the Cloud Adoption Framework?

The Cloud Adoption Framework is about adopting cloud technologies and integrating them into your business. It provides proven practices to govern, secure, and manage all workloads. It provides structured processes for migrating, modernizing, and building cloud-native solutions. Throughout each stage of adoption, the framework guides you to the right tools, architectures, and other resources to help you achieve your organizational goals. The framework integrates with the Well-Architected Framework and Azure Architecture Center.

Who is the Cloud Adoption Framework for?

The Microsoft Cloud Adoption Framework serves organizations across all industries and scales pursuing strategic Azure cloud adoption. This framework adapts to your specific cloud maturity level, from initial Azure evaluation through advanced multicloud optimization strategies. It's designed for anyone in the organization responsible for technology adoption, whether that responsibility spans business strategy, technology governance, cloud integration, or day-to-day operations.

Expand table

Organization Type	How the framework accelerates success
Startups	Build cloud-native applications using best practices. Access Microsoft for Startups for Azure credits and technical resources.
Enterprise	Establish a governance foundation to control workloads at enterprise scale.

Cloud Adoption Framework scenarios

The Cloud Adoption Framework has specific guidance aligned to common scenarios across our customer base.

Expand table

CAF Scenario	Purpose
Data	How to unify your data platform for AI and analytics
AI agents	How and when to adopt AI agents
AI adoption	How to adopt AI (generative and nongenerative)
Azure Virtual Desktop	How to adopt Azure Virtual Desktop
Azure VMware Solution	How to adopt Azure VMware Solution
Cloud-Scale Analytics	How to do analytics in the cloud
Oracle on Azure	Run Oracle workloads on Azure.
SAP on Azure	How to run SAP workloads on Azure

Last updated on 01/26/2026

Azure Architecture Center

Design solutions on Azure by using established patterns and practices. Azure Architecture Center is a catalog of solution ideas, example workloads, reference architectures, technology decision guides, and architecture guides for Azure workloads.



ARCHITECTU...
Browse Azure architectures



CONCEPT
Build by using application architecture...



REFERENCE
Make technology choices



CONCEPT
Learn cloud design patterns



WHAT'S NEW
See what's new and updated

Architect workloads on Azure

Explore best practices and patterns for building workloads on Microsoft Azure.



Design for the cloud

- Learn fundamental architecture styles
- Use best practices in cloud applications
- Avoid performance anti-patterns
- Use responsible engineering practices
- Use cloud design patterns



Choose the right technology

- Choose a compute service
- Choose a container service
- Choose a data store
- Choose AI services
- Choose a messaging service





Implement an Azure landing zone

- Deploy a platform landing zone
- Automate through subscription vending
- Application landing zone solutions

Specialized scenarios

- Architecture for startups
- Architecture for SaaS and multitenant applications
- Build mission-critical workloads
- Azure for AWS professionals
- Azure for Google Cloud professionals

Technology areas

Explore architectures and guides for different technologies.

Popular articles

- Microsoft Foundry chat baseline architecture
- Develop and optimize a RAG implementation
- AI agent orchestration patterns
- Cloud design patterns
- Hub-spoke network topology
- AKS production baseline
- Choose your Azure compute service

AI & Machine Learning

- AI architecture design
- Microsoft Foundry chat baseline architecture
- Build a generative AI gateway
- AI agent orchestration patterns
- Develop and optimize a RAG implementation
- Use MLOps
- Use GenAIOps

Analytics

- Analytics architecture design
- Choose a data analytics technology
- Choose a Microsoft Fabric deployment pattern
- Analytics end-to-end with Microsoft Fabric
- Enterprise BI with Microsoft Fabric
- Near real-time lakehouse data processing
- Stream processing with Azure Databricks

Compute

- Choose an Azure compute service
- Run a Linux VM on Azure
- Run a Windows VM on Azure
- Azure VM baseline architecture
- Understand multi-region compute balancing
- Build workloads on spot virtual machines

Containers

- Choose a container service

Databases

- Databases architecture design

- Design for Azure Kubernetes Service
- Azure Kubernetes Service baseline cluster
- Plan Day-2 operations on AKS
- Deploy microservices with Azure Container Apps
- Deploy microservices with Azure Container Apps and Dapr

- Migrate an Oracle database to Azure
- DataOps for a modern data warehouse
- Design a medallion lakehouse
- Use change feed to replicate data

Hybrid + multicloud

- Hybrid architecture design
- Azure hybrid options
- Connect a cross-premises network to Azure
- Azure Local baseline architecture
- Hybrid Kubernetes clusters
- Hybrid file services

Identity

- Identity architecture design
- Identity in multitenant architectures
- Integrate on-premises AD with Microsoft Entra ID
- Extend AD DS to Azure
- Create an AD DS forest in Azure

Networking

- Choose a hybrid network architecture
- Hub-spoke topology
- ExpressRoute with VPN failover
- Implement a secure hybrid network
- Highly available network virtual appliances

Security

- Security architecture design
- Azure security in AWS
- Zero-trust network with Azure Firewall and Application Gateway
- Securely managed web applications

Web apps

- Enterprise web app patterns
- Basic web application
- Baseline zone-redundant web application
- Protect APIs with Application Gateway and API Management
- Serverless web application

Cloud adoption and workload design

Build a strong cloud adoption strategy and a consistent approach to workload design.



Cloud Adoption Framework for Azure



Azure Well-Architected Framework pillars



Well-architected workloads

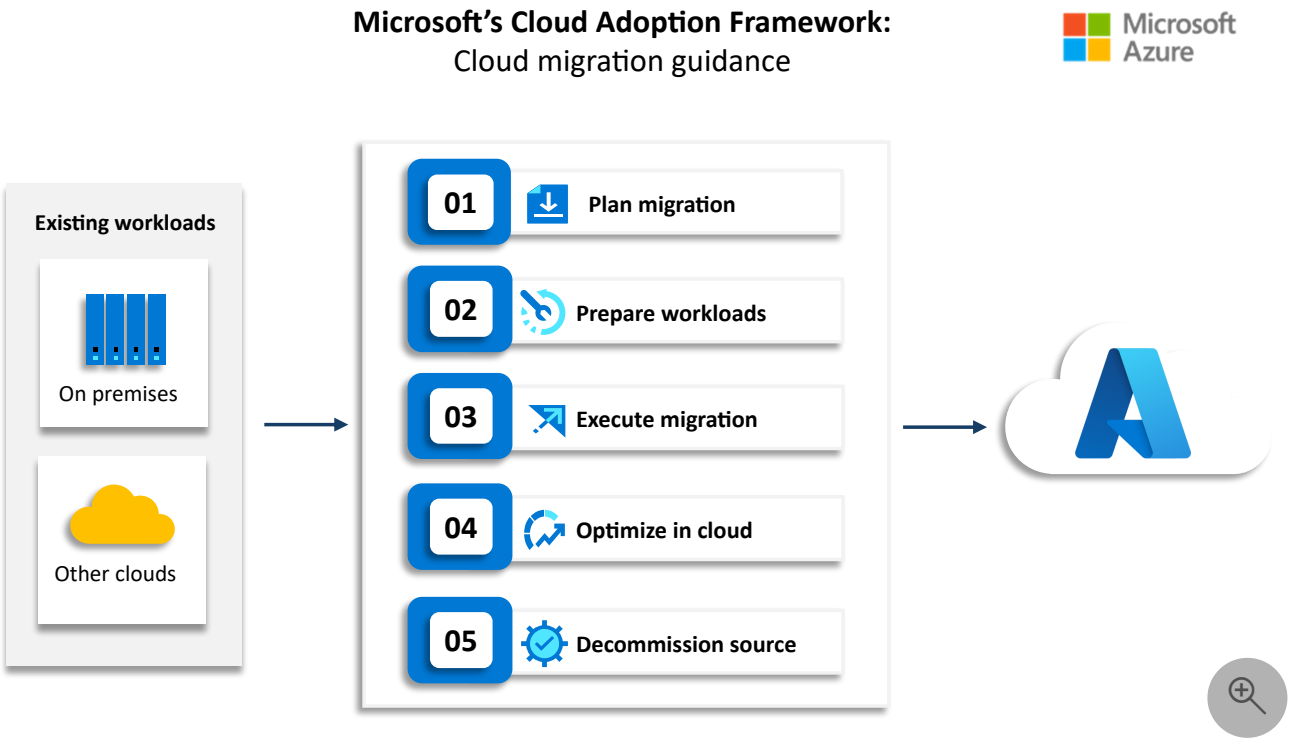


Well-architected service guides

Plan your migration

A migration plan defines the specific order, timing, and approach for migrating workloads to Azure. This plan translates high-level migration strategies into actionable deployment sequences. It builds on the cloud adoption plan by addressing tactical decisions such as workload prioritization, migration sequencing, and data transfer methods.

Prerequisites: Migration adoption plan, Azure landing zone



Assess migration readiness and skills

A readiness assessment ensures that your team has the skills and support needed to execute the migration plan. This step identifies capability gaps and accelerates progress through targeted training or external support.

1. **Evaluate your team's Azure skills.** Review your team's experience with Azure services, migration tools, and cutover. This evaluation helps you identify specific knowledge gaps and determine what training your team needs to succeed.

2. **Engage external expertise when needed.** If your team lacks experience with cloud migration, bring in Microsoft or a Microsoft partner. External experts can validate your migration strategy, recommend appropriate tools, and help establish realistic timelines. This support reduces risk and speeds up your migration, especially for complex or large-scale projects.

Choose your data migration path

A data migration path is how you move data from your current location to Azure. The right path ensures your data transfers securely, quickly, and cost-effectively. First, check what network connections you have available, ExpressRoute, VPN, or public internet, to understand your options.

1. **Use ExpressRoute when you have it.** ExpressRoute gives you a private, dedicated connection to Azure that's faster and more secure than internet connections. If you already have ExpressRoute or plan to get it, use this method for all your workloads. Keep in mind that ExpressRoute requires setup time and has data transfer costs.

2. **Use VPN if ExpressRoute isn't available.** Choose a VPN when you need secure data transfer but don't have ExpressRoute. A VPN creates an encrypted tunnel over the internet to Azure, though it's typically slower than ExpressRoute. Make sure you have a VPN Gateway configured in Azure before starting.

3. **Use Azure Data Box for large amounts of data.** Data Box is best for offline migrations with lots of data. Microsoft ships you a physical device to copy your data onto, then you ship it back. This option avoids using your network but takes the longest due to shipping time.

4. **Use public internet for less sensitive data.** This option works when your data doesn't need encryption and you can't use ExpressRoute or Data Box. While this method is available everywhere, it's the least secure and can slow down your other internet activities.

Expand table

Data Migration Path	When to use	Pros	Cons
ExpressRoute	Any workload when available	Secure and fast	Setup required, costs money
VPN	Secure transfers when no ExpressRoute	More secure than public internet	Requires setup, slower than ExpressRoute
Azure Data Box	Offline migration with large amounts of data	Moves data without using your network	Slowest method due to shipping
Public internet	Nonsensitive data and can't use Data Box	Works everywhere	Least secure, uses your bandwidth

Determine the migration sequence

Migration sequencing reduces risk and builds team confidence by establishing a logical order for workload migration. The sequence determines which workloads move first and how dependent components migrate together to prevent service disruptions. Organize large portfolios into migration waves. For detailed guidance on wave planning, see Migration wave planning.

Find dependencies

1. **Discover all dependencies first.** Dependencies between workloads cause service disruptions if not migrated together. Map internal and external dependencies to discover these connections before creating migration groups.

2. **Analyze dependency types and criticality.** Different dependency types require different migration approaches. Distinguish between these categories:

Expand table

Dependency type	Description	Migration approach
Direct dependencies	Require immediate communication and low latency between components.	Move all directly connected components together to maintain performance and avoid disruptions.
Indirect dependencies	Involve occasional or non-critical interactions between systems.	Migrate together or in separate waves if the connection tolerates latency or supports hybrid use.
Business dependencies	Depend on organizational or management relationships.	Group and migrate related workloads and reporting systems together to align with business priorities.

3. **Group workloads by dependency relationships.** Create groups based on shared databases, APIs, authentication services, or network connections. These groups form the foundation of your migration waves and ensure all components required for functionality move together. When uncertainty exists about dependency criticality, group the components together. This conservative approach provides flexibility for future separation.

4. **Document each dependency group systematically.** Tag assets based on their dependency groups using consistent naming conventions. Document each group with:

    - **Group name and ID** - Unique identifier and descriptive name

- **Component inventory** - All infrastructure elements, applications, and services
- **Critical dependencies** - Essential connections requiring special handling
- **Migration constraints** - Business, technical, or timing requirements

5. **Validate group completeness.** Confirm each group includes all necessary components for applications to operate, including supporting infrastructure such as load balancers, DNS records, or caching layers.

Address split-environment operations

1. **Plan for unmovable dependencies.** Identify components that must stay in your source environment due to technical or regulatory reasons. Document why they can't move, how they connect to other systems, and what data they share. This documentation helps you create strategies for these components to work smoothly with your cloud systems.

2. **Minimize split-environment operation time.** When components can move to the cloud later but not immediately, document their connections and data flows with cloud systems. Create a clear plan with timelines and risk management approaches to reduce the time your workload operates across both environments. Consider delaying the migration until more components can move together.

3. **Connect environments effectively.** Use integration methods like API gateways, message queues, and data synchronization to create reliable connections between your cloud workloads and source environment components. These approaches reduce delays, improve security, and prepare the way for eventually moving remaining components to the cloud.

Prioritize workloads to migrate

1. **Review workload details.** Work with stakeholders to review business and technical details for each workload. Ensure that downtime or failure impacts are well understood and align with current business priorities. Use the migration adoption plan to verify details like business unit, workload owner, technical dependencies, and criticality classification. These details help prioritize and sequence workloads effectively.

Expand table

Priority	Business value	Effort	Description
High	High	Low	Quick wins - migrate first for immediate impact

Priority	Business value	Effort	Description
Medium-High	High	High	Strategic investments - plan carefully with adequate resources
Medium-Low	Low	Low	Easy candidates - fill gaps between major migrations
Low	Low	High	Avoid or defer - focus resources on higher-value opportunities

2. **Start with simpler workloads to reduce risk.** Begin migrating workloads that are less complex and have lower risk. This approach helps your team gain confidence and refine migration processes before tackling more challenging workloads. Target internal tools, development environments, or low-usage applications with standalone architectures and minimal integration points.

3. **Move non-production environments before production.** Nonproduction environments provide a safe space to test the full migration process. Migrate development, staging, and QA environments before production to validate readiness. This order allows teams to test configurations, performance, and recovery procedures without affecting users. Use nonproduction migrations to train operations teams.

4. **Schedule critical systems after you demonstrate initial success.** Critical applications require proven migration capabilities before you move them to Azure. Plan these migrations for later waves when your team demonstrates competency with Azure services. Business deadlines such as hardware refresh cycles might require you to prioritize critical applications earlier with more safeguards and extended testing periods.

5. **Include representative complex workloads to test scenarios.** Add one or two complex workloads to each early wave to expose challenges you face with mission-critical applications. Choose workloads that represent common patterns such as multi-tier applications or database-dependent systems.

Create a detailed migration schedule

1. **Set start and end dates for each migration.** Include buffer time for testing and issue resolution to ensure smooth execution. This detailed scheduling reduces the risk of delays and supports effective resource planning.

2. **Align timelines with business events.** Avoid scheduling migrations during critical business periods such as financial close, product launches, or holiday seasons. This alignment reduces the risk of business disruption and ensures stakeholder confidence.

3. **Use project management tools to track progress.** Use tools like Azure DevOps to manage dependencies, track milestones, and communicate changes effectively. These tools provide visibility into migration progress and support proactive issue resolution.

Choose the migration method for each workload

Migration methods fall into two categories: migration with downtime and migration with near-zero downtime. Choose the best migration method for each workload based on its downtime tolerance and business criticality.

1. **Choose downtime migration for workloads that tolerate planned outages.** Downtime migration is simpler and faster because it doesn't require real-time synchronization between source and target environments. This method works well for noncritical workloads such as development environments, test systems, or applications with scheduled maintenance windows. Document the acceptable downtime duration for each workload and schedule migrations during low-usage periods to minimize business effects.

2. **Choose near-zero downtime migration for critical workloads.** Near-zero downtime migration ensures that critical workloads remain operational during the transition through continuous data replication and cutover techniques. This method is essential for customer-facing applications, real-time transaction systems, or workloads with strict service level agreements. Validate that the workload architecture supports continuous replication and that network bandwidth can handle real-time data transfer. Test connectivity and replication processes in a nonproduction environment to confirm readiness for this migration method.

Expand table

Migration Method	When to use	Pros	Cons
Downtime migration	Noncritical workloads, development environments	Simpler process, faster execution	Service interruption required
Near-zero downtime migration	Critical workloads, strict SLAs	Minimal service disruption	Complex setup and requires testing

Define rollback plan

A rollback plan enables teams to quickly reverse changes when a deployment fails or introduces risk. A well-defined plan minimizes downtime, limits business impact, and maintains

system reliability. Always establish rollback criteria and procedures before initiating any migration or deployment.

1. **Define failed deployment.** Collaborate with business stakeholders, workload owners, and operations teams to decide what counts as a failed deployment. Examples include failed health checks, poor performance, security issues, or unmet success metrics. This definition ensures rollback decisions align with your organization's risk tolerance. Include specific conditions that trigger a rollback in your deployment plan, such as CPU usage limits, response time thresholds, or error rates. This evaluation makes rollback decisions clear and consistent during incidents.

2. **Automate rollback steps in CI/CD pipelines.** Use tools like Azure Pipelines or GitHub Actions to automate rollback tasks. For example, set up pipelines to redeploy a prior version if health checks fail.

3. **Create workload-specific rollback instructions.** Write rollback steps that match your workload type, environment, and deployment method. For example, IaC deployments require reapplying past templates. App rollbacks involve redeploying a prior container image. Attach rollback scripts, configuration snapshots, and IaC templates to your rollback plan. These assets speed up the process and reduce the need for manual steps.

4. **Test rollback procedures.** Simulate deployment failures in a staging environment to confirm your rollback works. Find and fix gaps in automation, permissions, or dependencies. Verify that the rollback procedure restores the system to a stable, known-good state.

5. **Improve rollback strategies.** After each deployment or rollback, review what worked and what didn't. Update rollback rules, steps, and scripts based on lessons learned. Factor in design changes or new tools. Keep your guides updated so rollback plans stay current and useful.

Engage stakeholders on migration plan

Stakeholder approval validates that your migration plan meets business requirements and risk tolerance. You should secure formal approval before executing migrations.

1. **Document the migration plan with business justification.** Create a structured plan showing workload name, owner, criticality, migration method, downtime window, and business effects. Include rationale for each approach and explain how it minimizes risk.

2. **Present tested rollback procedures.** Show specific rollback plans with steps, timeframes, and success criteria. Include automated and manual capabilities. Document preproduction test results to prove quick service restoration.

3. **Validate schedules against business constraints.** Review timelines with stakeholders to avoid critical business periods, maintenance freezes, and seasonal peaks. Provide alternative options with trade-offs if conflicts exist.

4. **Obtain formal approval and rollback authority.** Secure written approval from stakeholders for the migration plan and rollback procedures. Define decision-making authority and establish emergency communication channels.

5. **Define success criteria and review checkpoints.** Set measurable metrics including performance benchmarks, functionality validation, and user acceptance criteria. Schedule formal review points for go/no-go decisions.

Next step

Prepare workloads for the cloud

Last updated on 08/01/2025

Common Azure CLI commands for managing Azure resources

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

The Azure CLI allows you to create and manage your Azure resources on macOS, Linux, and Windows. This article details some of the most common commands to create and manage virtual machines (VMs).

This article requires the Azure CLI version 2.0.4 or later. Run `az --version` to find the version. If you need to upgrade, see Install Azure CLI. You can also use Cloud Shell from your browser.

Basic Azure Resource Manager commands in Azure CLI

For more detailed help with specific command line switches and options, you can use the online command help and options by typing `az <command> <subcommand> --help`.

Create VMs

Expand table

Task	Azure CLI commands
Create a resource group	`az group create --name myResourceGroup --location eastus`
Create a Linux VM	`az vm create --resource-group myResourceGroup --name myVM --image LinuxImageName`
Create a Windows VM	`az vm create --resource-group myResourceGroup --name myVM --image win2016datacenter`

Manage VM state

Expand table

Task	Azure CLI commands
Start a VM	`az vm start --resource-group myResourceGroup --name myVM`
Stop a VM	`az vm stop --resource-group myResourceGroup --name myVM`
Deallocate a VM	`az vm deallocate --resource-group myResourceGroup --name myVM`
Restart a VM	`az vm restart --resource-group myResourceGroup --name myVM`
Redeploy a VM	`az vm redeploy --resource-group myResourceGroup --name myVM`
Delete a VM	`az vm delete --resource-group myResourceGroup --name myVM`

Get VM info

Expand table

Task	Azure CLI commands
List VMs	`az vm list`
Get information about a VM	`az vm show --resource-group myResourceGroup --name myVM`
Get usage of VM resources	`az vm list-usage --location eastus`
Get all available VM sizes	`az vm list-sizes --location eastus`

Disks and images

Expand table

Task	Azure CLI commands
Add a data disk to a VM	`az vm disk attach --resource-group myResourceGroup --vm-name myVM --disk myDataDisk --size-gb 128 --new`
List attached disks to a VM	`az vm show --resource-group groupName --name vmName --query "storageProfile"`
Remove a data disk from a VM	`az vm disk detach --resource-group myResourceGroup --vm-name myVM --disk myDataDisk`
Resize a disk	`az disk update --resource-group myResourceGroup --name myDataDisk --size-gb 256`

Task	Azure CLI commands
Snapshot a disk	`az snapshot create --resource-group myResourceGroup --name mySnapshot --source myDataDisk`
Create image of a VM	`az image create --resource-group myResourceGroup --source myVM --name myImage`
Create VM from image	`az vm create --resource-group myResourceGroup --name myNewVM --image myImage`

Next steps

For additional examples of the CLI commands, see the Create and Manage Linux VMs with the Azure CLI tutorial.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Common PowerShell commands for creating and managing Azure Virtual Machines

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs ✔ Flexible scale sets

This article covers some of the basic Azure PowerShell commands that you can use to create and manage virtual machines in your Azure subscription. For more detailed help with specific command-line switches and options, you can use the **Get-Help** *command*.

These variables might be useful if running more than one of the commands in this article:

- $location - The location of the virtual machine. You can use Get-AzLocation to find a geographical region that works for you.
- $myResourceGroup - The name of the resource group that contains the virtual machine.
- $myVM - The name of the virtual machine.

Create a VM - simplified

Expand table

Task	Command
Create a simple VM	New-AzVM -Name $myVM New-AzVM has a set of *simplified* parameters, where all that is required is a single name. The value for -Name will be used as the name for all of the resources required for creating a new VM. You can specify more, but this is all that is required.
Create a VM from a custom image	New-AzVm -ResourceGroupName $myResourceGroup -Name $myVM ImageName "myImage" -Location $location You need to have already created your own managed image. You can use an image to make multiple, identical VMs.

Create a VM - advanced

Expand table

Task	Command
Create a VM configuration	$vm = New-AzVMConfig -VMName $myVM -VMSize "Standard_D1_v1" The VM configuration is used to define or update settings for the VM. The configuration is initialized with the name of the VM and its size.
Add configuration settings	$vm = Set-AzVMOperatingSystem -VM $vm -Windows -ComputerName $myVM -Credential $cred -ProvisionVMAgent -EnableAutoUpdate Operating system settings including credentials are added to the configuration object that you previously created using New-AzVMConfig.
Add a network interface	$vm = Add-AzVMNetworkInterface -VM $vm -Id $nic.Id A VM must have a network interface to communicate in a virtual network. You can also use Get-AzNetworkInterface to retrieve an existing network interface object.
Specify a platform image	$vm = Set-AzVMSourceImage -VM $vm -PublisherName "publisher_name" -Offer "publisher_offer" -Skus "product_sku" -Version "latest" Image information is added to the configuration object that you previously created using New-AzVMConfig. The object returned from this command is only used when you set the OS disk to use a platform image.
Create a VM	New-AzVM -ResourceGroupName $myResourceGroup -Location $location -VM $vm All resources are created in a resource group. Before you run this command, run New-AzVMConfig, Set-AzVMOperatingSystem, Set-AzVMSourceImage, Add-AzVMNetworkInterface, and Set-AzVMOSDisk.
Update a VM	Update-AzVM -ResourceGroupName $myResourceGroup -VM $vm

Task	Command
	Get the current VM configuration using Get-AzVM, change configuration settings on the VM object, and then run this command.

Get information about your VMs

Expand table

Task	Command
List VMs in a subscription	Get-AzVM
List VMs in a resource group	Get-AzVM -ResourceGroupName $myResourceGroup To get a list of resource groups in your subscription, use Get-AzResourceGroup.
Get information about a VM	Get-AzVM -ResourceGroupName $myResourceGroup -Name $myVM

Manage your VMs

Expand table

Task	Command
Start a VM	Start-AzVM -ResourceGroupName $myResourceGroup -Name $myVM
Stop a VM	Stop-AzVM -ResourceGroupName $myResourceGroup -Name $myVM
Restart a running VM	Restart-AzVM -ResourceGroupName $myResourceGroup -Name $myVM
Delete a VM	Remove-AzVM -ResourceGroupName $myResourceGroup -Name $myVM

Next steps

- See the basic steps for creating a virtual machine in Create a Windows VM using Resource Manager and PowerShell.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Common PowerShell commands for Azure Virtual Networks

Article • 08/22/2024

Applies to: ✔ Linux VMs ✔ Windows VMs

If you want to create a virtual machine, you need to create a virtual network or know about an existing virtual network in which the VM can be added. Typically, when you create a VM, you also need to consider creating the resources described in this article.

See How to install and configure Azure PowerShell for information about installing the latest version of Azure PowerShell, selecting your subscription, and signing in to your account.

Some variables might be useful for you if running more than one of the commands in this article:

- $location - The location of the network resources. You can use Get-AzLocation to find a geographical region that works for you.
- $myResourceGroup - The name of the resource group where the network resources are located.

Create network resources

Expand table

Task	Command
Create subnet configurations	$subnet1 = New-AzVirtualNetworkSubnetConfig -Name "mySubnet1" -AddressPrefix XX.X.X.X/XX $subnet2 = New-AzVirtualNetworkSubnetConfig -Name "mySubnet2" -AddressPrefix XX.X.X.X/XX A typical network might have a subnet for an internet facing load balancer and a separate subnet for an internal load balancer.
Create a virtual network	$vnet = New-AzVirtualNetwork -Name "myVNet" -ResourceGroupName $myResourceGroup -Location $location -AddressPrefix XX.X.X.X/XX -Subnet $subnet1, $subnet2
Test for a unique domain name	Test-AzDnsAvailability -DomainNameLabel "myDNS" -Location $location You can specify a DNS domain name for a public IP resource, which creates a

Task	Command
	mapping for domainname.location.cloudapp.azure.com to the public IP address in the Azure-managed DNS servers. The name can contain only letters, numbers, and hyphens. The first and last character must be a letter or number and the domain name must be unique within its Azure location. If **True** is returned, your proposed name is globally unique.
Create a public IP address	$pip = New-AzPublicIpAddress -Name "myPublicIp" -ResourceGroupName $myResourceGroup -DomainNameLabel "myDNS" -Location $location -AllocationMethod Dynamic The public IP address uses the domain name that you previously tested and is used by the frontend configuration of the load balancer.
Create a frontend IP configuration	$frontendIP = New-AzLoadBalancerFrontendIpConfig -Name "myFrontendIP" -PublicIpAddress $pip The frontend configuration includes the public IP address that you previously created for incoming network traffic.
Create a backend address pool	$beAddressPool = New-AzLoadBalancerBackendAddressPoolConfig -Name "myBackendAddressPool" Provides internal addresses for the backend of the load balancer that are accessed through a network interface.
Create a probe	$healthProbe = New-AzLoadBalancerProbeConfig -Name "myProbe" -RequestPath 'HealthProbe.aspx' -Protocol http -Port 80 -IntervalInSeconds 15 -ProbeCount 2 Contains health probes used to check availability of virtual machines instances in the backend address pool.
Create a load balancing rule	$lbRule = New-AzLoadBalancerRuleConfig -Name HTTP -FrontendIpConfiguration $frontendIP -BackendAddressPool $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 80 Contains rules that assign a public port on the load balancer to a port in the backend address pool.
Create an inbound NAT rule	$inboundNATRule = New-AzLoadBalancerInboundNatRuleConfig -Name "myInboundRule1" -FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3441 -BackendPort 3389 Contains rules mapping a public port on the load balancer to a port for a specific virtual machine in the backend address pool.
Create a load balancer	$loadBalancer = New-AzLoadBalancer -ResourceGroupName $myResourceGroup -Name "myLoadBalancer" -Location $location -FrontendIpConfiguration $frontendIP -InboundNatRule $inboundNATRule -

Task	Command
	LoadBalancingRule $lbRule -BackendAddressPool $beAddressPool -Probe $healthProbe
Create a network interface	$nic1= New-AzNetworkInterface -ResourceGroupName $myResourceGroup -Name "myNIC" -Location $location -PrivateIpAddress XX.X.X.X -Subnet $subnet2 -LoadBalancerBackendAddressPool $loadBalancer.BackendAddressPools[0] -LoadBalancerInboundNatRule $loadBalancer.InboundNatRules[0] Create a network interface using the public IP address and virtual network subnet that you previously created.

Get information about network resources

Expand table

Task	Command
List virtual networks	Get-AzVirtualNetwork -ResourceGroupName $myResourceGroup Lists all the virtual networks in the resource group.
Get information about a virtual network	Get-AzVirtualNetwork -Name "myVNet" -ResourceGroupName $myResourceGroup
List subnets in a virtual network	Get-AzVirtualNetwork -Name "myVNet" -ResourceGroupName $myResourceGroup | Select Subnets
Get information about a subnet	Get-AzVirtualNetworkSubnetConfig -Name "mySubnet1" -VirtualNetwork $vnet Gets information about the subnet in the specified virtual network. The $vnet value represents the object returned by Get-AzVirtualNetwork.
List IP addresses	Get-AzPublicIpAddress -ResourceGroupName $myResourceGroup Lists the public IP addresses in the resource group.
List load balancers	Get-AzLoadBalancer -ResourceGroupName $myResourceGroup Lists all the load balancers in the resource group.
List network interfaces	Get-AzNetworkInterface -ResourceGroupName $myResourceGroup Lists all the network interfaces in the resource group.

Task	Command
Get information about a network interface	Get-AzNetworkInterface -Name "myNIC" -ResourceGroupName $myResourceGroup Gets information about a specific network interface.
Get the IP configuration of a network interface	Get-AzNetworkInterfaceIPConfig -Name "myNICIP" -NetworkInterface $nic Gets information about the IP configuration of the specified network interface. The $nic value represents the object returned by Get-AzNetworkInterface.

Manage network resources

Expand table

Task	Command
Add a subnet to a virtual network	Add-AzVirtualNetworkSubnetConfig -AddressPrefix XX.X.X.X/XX -Name "mySubnet1" -VirtualNetwork $vnet Adds a subnet to an existing virtual network. The $vnet value represents the object returned by Get-AzVirtualNetwork.
Delete a virtual network	Remove-AzVirtualNetwork -Name "myVNet" -ResourceGroupName $myResourceGroup Removes the specified virtual network from the resource group.
Delete a network interface	Remove-AzNetworkInterface -Name "myNIC" -ResourceGroupName $myResourceGroup Removes the specified network interface from the resource group.
Delete a load balancer	Remove-AzLoadBalancer -Name "myLoadBalancer" -ResourceGroupName $myResourceGroup Removes the specified load balancer from the resource group.
Delete a public IP address	Remove-AzPublicIpAddress-Name "myIPAddress" -ResourceGroupName $myResourceGroup Removes the specified public IP address from the resource group.

Next Steps

Use the network interface that you just created when you create a VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Virtual machines in an Azure Resource Manager template

Article • 08/22/2024

Applies to: ✔ Windows VMs

This article describes aspects of an Azure Resource Manager template that apply to virtual machines. This article doesn't describe a complete template for creating a virtual machine; for that you need resource definitions for storage accounts, network interfaces, public IP addresses, and virtual networks. For more information about how these resources can be defined together, see the Resource Manager template walkthrough.

There are many templates in the gallery that include the VM resource. Not all elements that can be included in a template are described here.

This example shows a typical resource section of a template for creating a specified number of VMs:

JSON

```
"resources": [
  {
    "apiVersion": "2016-04-30-preview",
    "type": "Microsoft.Compute/virtualMachines",
    "name": "[concat('myVM', copyindex())]",
    "location": "[resourceGroup().location]",
    "copy": {
      "name": "virtualMachineLoop",
      "count": "[parameters('numberOfInstances')]"
    },
    "dependsOn": [
      "[concat('Microsoft.Network/networkInterfaces/myNIC', copyindex())]"
    ],
    "properties": {
      "hardwareProfile": {
        "vmSize": "Standard_DS1"
      },
      "osProfile": {
        "computername": "[concat('myVM', copyindex())]",
        "adminUsername": "[parameters('adminUsername')]",
        "adminPassword": "[parameters('adminPassword')]"
      },
      "storageProfile": {
        "imageReference": {
          "publisher": "MicrosoftWindowsServer",
          "offer": "WindowsServer",
          "sku": "2012-R2-Datacenter",
          "version": "latest"
```

```
        },
        "osDisk": {
          "name": "[concat('myOSDisk', copyindex())]",
          "caching": "ReadWrite",
          "createOption": "FromImage"
        },
        "dataDisks": [
          {
            "name": "[concat('myDataDisk', copyindex())]",
            "diskSizeGB": "100",
            "lun": 0,
            "createOption": "Empty"
          }
        ]
      },
      "networkProfile": {
        "networkInterfaces": [
          {
            "id": "[resourceId('Microsoft.Network/networkInterfaces',
              concat('myNIC', copyindex()))]"
          }
        ]
      },
      "diagnosticsProfile": {
        "bootDiagnostics": {
          "enabled": "true",
          "storageUri": "[concat('https://', variables('storageName'),
'.blob.core.windows.net')]"
        }
      }
    },
    "resources": [
      {
        "name": "Microsoft.Insights.VMDiagnosticsSettings",
        "type": "extensions",
        "location": "[resourceGroup().location]",
        "apiVersion": "2016-03-30",
        "dependsOn": [
          "[concat('Microsoft.Compute/virtualMachines/myVM', copyindex())]"
        ],
        "properties": {
          "publisher": "Microsoft.Azure.Diagnostics",
          "type": "IaaSDiagnostics",
          "typeHandlerVersion": "1.5",
          "autoUpgradeMinorVersion": true,
          "settings": {
            "xmlCfg": "[base64(concat(variables('wadcfgxstart'),
            variables('wadmetricsresourceid'),
            concat('myVM', copyindex()),
            variables('wadcfgxend')))]",
            "storageAccount": "[variables('storageName')]"
          },
          "protectedSettings": {
            "storageAccountName": "[variables('storageName')]",
            "storageAccountKey": "[listkeys(variables('accountid'),
```

```
              '2015-06-15').key1]",
            "storageAccountEndPoint": "https://core.windows.net"
          }
        }
      },
      {
        "name": "MyCustomScriptExtension",
        "type": "extensions",
        "apiVersion": "2016-03-30",
        "location": "[resourceGroup().location]",
        "dependsOn": [
          "[concat('Microsoft.Compute/virtualMachines/myVM', copyindex())]"
        ],
        "properties": {
          "publisher": "Microsoft.Compute",
          "type": "CustomScriptExtension",
          "typeHandlerVersion": "1.7",
          "autoUpgradeMinorVersion": true,
          "settings": {
            "fileUris": [
              "[concat('https://', variables('storageName'),
                '.blob.core.windows.net/customscripts/start.ps1')]"
            ],
            "commandToExecute": "powershell.exe -ExecutionPolicy
Unrestricted -File start.ps1"
          }
        }
      }
    ]
  }
]
```

> **Note**
>
> This example relies on a storage account that was previously created. You could create the storage account by deploying it from the template. The example also relies on a network interface and its dependent resources that would be defined in the template. These resources are not shown in the example.

API Version

When you deploy resources using a template, you have to specify a version of the API to use. The example shows the virtual machine resource using this apiVersion element:

JSON

```
"apiVersion": "2016-04-30-preview",
```

The version of the API you specify in your template affects which properties you can define in the template. In general, you should select the most recent API version when creating templates. For existing templates, you can decide whether you want to continue using an earlier API version, or update your template for the latest version to take advantage of new features.

Use these opportunities for getting the latest API versions:

- REST API - List all resource providers
- PowerShell - Get-AzResourceProvider
- Azure CLI - az provider show

Parameters and variables

Parameters make it easy for you to specify values for the template when you run it. This parameters section is used in the example:

JSON

```
"parameters": {
  "adminUsername": { "type": "string" },
  "adminPassword": { "type": "securestring" },
  "numberOfInstances": { "type": "int" }
},
```

When you deploy the example template, you enter values for the name and password of the administrator account on each VM and the number of VMs to create. You have the option of specifying parameter values in a separate file that's managed with the template, or providing values when prompted.

Variables make it easy for you to set up values in the template that are used repeatedly throughout it or that can change over time. This variables section is used in the example:

JSON

```
"variables": {
  "storageName": "mystore1",
  "accountid": "[concat('/subscriptions/', subscription().subscriptionId,
    '/resourceGroups/', resourceGroup().name,
  '/providers/','Microsoft.Storage/storageAccounts/',
variables('storageName'))]",
  "wadlogs": "<WadCfg>
    <DiagnosticMonitorConfiguration overallQuotaInMB=\"4096\"
xmlns=\"http://schemas.microsoft.com/ServiceHosting/2010/10/DiagnosticsConfi
guration\">
      <DiagnosticInfrastructureLogs
```

```
scheduledTransferLogLevelFilter=\"Error\"/>
      <WindowsEventLog scheduledTransferPeriod=\"PT1M\" >
        <DataSource name=\"Application!*[System[(Level = 1 or Level = 2)]]\"
/>
        <DataSource name=\"Security!*[System[(Level = 1 or Level = 2)]]\" />
        <DataSource name=\"System!*[System[(Level = 1 or Level = 2)]]\" />
      </WindowsEventLog>",
  "wadperfcounters": "<PerformanceCounters scheduledTransferPeriod=\"PT1M\">
      <PerformanceCounterConfiguration
counterSpecifier=\"\\Process(_Total)\\Thread Count\" sampleRate=\"PT15S\"
unit=\"Count\">
        <annotation displayName=\"Threads\" locale=\"en-us\"/>
      </PerformanceCounterConfiguration>
    </PerformanceCounters>",
  "wadcfgxstart": "[concat(variables('wadlogs'),
variables('wadperfcounters'),
    '<Metrics resourceId=\"')]",
  "wadmetricsresourceid": "[concat('/subscriptions/',
subscription().subscriptionId,
    '/resourceGroups/', resourceGroup().name ,
    '/providers/', 'Microsoft.Compute/virtualMachines/')]",
  "wadcfgxend": "\"><MetricAggregation scheduledTransferPeriod=\"PT1H\"/>
    <MetricAggregation scheduledTransferPeriod=\"PT1M\"/>
    </Metrics></DiagnosticMonitorConfiguration>
    </WadCfg>"
},
```

When you deploy the example template, variable values are used for the name and identifier of the previously created storage account. Variables are also used to provide the settings for the diagnostic extension. Use the best practices for creating Azure Resource Manager templates to help you decide how you want to structure the parameters and variables in your template.

Resource loops

When you need more than one virtual machine for your application, you can use a copy element in a template. This optional element loops through creating the number of VMs that you specified as a parameter:

JSON

```
"copy": {
  "name": "virtualMachineLoop",
  "count": "[parameters('numberOfInstances')]"
},
```

Also, notice in the example that the loop index is used when specifying some of the values for the resource. For example, if you entered an instance count of three, the

names of the operating system disks are myOSDisk1, myOSDisk2, and myOSDisk3:

JSON

```
"osDisk": {
  "name": "[concat('myOSDisk', copyindex())]",
  "caching": "ReadWrite",
  "createOption": "FromImage"
}
```

> **Note**
>
> This example uses managed disks for the virtual machines.

Keep in mind that creating a loop for one resource in the template may require you to use the loop when creating or accessing other resources. For example, multiple VMs can't use the same network interface, so if your template loops through creating three VMs it must also loop through creating three network interfaces. When assigning a network interface to a VM, the loop index is used to identify it:

JSON

```
"networkInterfaces": [ {
  "id": "[resourceId('Microsoft.Network/networkInterfaces',
    concat('myNIC', copyindex()))]"
} ]
```

Dependencies

Most resources depend on other resources to work correctly. Virtual machines must be associated with a virtual network and to do that it needs a network interface. The dependsOn element is used to make sure that the network interface is ready to be used before the VMs are created:

JSON

```
"dependsOn": [
  "[concat('Microsoft.Network/networkInterfaces/', 'myNIC', copyindex())]"
],
```

Resource Manager deploys in parallel any resources that aren't dependent on another resource being deployed. Be careful when setting dependencies because you can inadvertently slow your deployment by specifying unnecessary dependencies.

Dependencies can chain through multiple resources. For example, the network interface depends on the public IP address and virtual network resources.

How do you know if a dependency is required? Look at the values you set in the template. If an element in the virtual machine resource definition points to another resource that is deployed in the same template, you need a dependency. For example, your example virtual machine defines a network profile:

JSON

```
"networkProfile": {
  "networkInterfaces": [ {
    "id": "[resourceId('Microsoft.Network/networkInterfaces',
      concat('myNIC', copyindex())]"
  } ]
},
```

To set this property, the network interface must exist. Therefore, you need a dependency. You also need to set a dependency when one resource (a child) is defined within another resource (a parent). For example, the diagnostic settings and custom script extensions are both defined as child resources of the virtual machine. They can't be created until the virtual machine exists. Therefore, both resources are marked as dependent on the virtual machine.

Profiles

Several profile elements are used when defining a virtual machine resource. Some are required and some are optional. For example, the hardwareProfile, osProfile, storageProfile, and networkProfile elements are required, but the diagnosticsProfile is optional. These profiles define settings such as:

- size
- name and credentials
- disk and operating system settings
- network interface
- boot diagnostics

Disks and images

In Azure, vhd files can represent disks or images. When the operating system in a vhd file is specialized to be a specific VM, it's referred to as a disk. When the operating

system in a vhd file is generalized to be used to create many VMs, it's referred to as an image.

Create new virtual machines and new disks from a platform image

When you create a VM, you must decide what operating system to use. The imageReference element is used to define the operating system of a new VM. The example shows a definition for a Windows Server operating system:

JSON

```
"imageReference": {
  "publisher": "MicrosoftWindowsServer",
  "offer": "WindowsServer",
  "sku": "2012-R2-Datacenter",
  "version": "latest"
},
```

If you want to create a Linux operating system, you might use this definition:

JSON

```
"imageReference": {
  "publisher": "Canonical",
  "offer": "UbuntuServer",
  "sku": "20.04.2-LTS",
  "version": "latest"
},
```

> ⓘ **Note**
>
> Modify `publisher`, `offer`, `sku` and `version` values accordingly.

Configuration settings for the operating system disk are assigned with the osDisk element. The example defines a new managed disk with the caching mode set to **ReadWrite** and that the disk is being created from a platform image:

JSON

```
"osDisk": {
  "name": "[concat('myOSDisk', copyindex())]",
  "caching": "ReadWrite",
```

```
  "createOption": "FromImage"
},
```

Create new virtual machines from existing managed disks

If you want to create virtual machines from existing disks, remove the imageReference and the osProfile elements and define these disk settings:

JSON

```
"osDisk": {
  "osType": "Windows",
  "managedDisk": {
    "id": "[resourceId('Microsoft.Compute/disks', [concat('myOSDisk',
copyindex())])]"
  },
  "caching": "ReadWrite",
  "createOption": "Attach"
},
```

Create new virtual machines from a managed image

If you want to create a virtual machine from a managed image, change the imageReference element and define these disk settings:

JSON

```
"storageProfile": {
  "imageReference": {
    "id": "[resourceId('Microsoft.Compute/images', 'myImage')]"
  },
  "osDisk": {
    "name": "[concat('myOSDisk', copyindex())]",
    "osType": "Windows",
    "caching": "ReadWrite",
    "createOption": "FromImage"
  }
},
```

Attach data disks

You can optionally add data disks to the VMs. The number of disks depends on the size of operating system disk that you use. With the size of the VMs set to Standard_DS1_v2, the maximum number of data disks that could be added to the them is two. In the example, one managed data disk is being added to each VM:

JSON

```
"dataDisks": [
  {
    "name": "[concat('myDataDisk', copyindex())]",
    "diskSizeGB": "100",
    "lun": 0,
    "caching": "ReadWrite",
    "createOption": "Empty"
  }
],
```

Extensions

Although extensions are a separate resource, they're closely tied to VMs. Extensions can be added as a child resource of the VM or as a separate resource. The example shows the Diagnostics Extension being added to the VMs:

JSON

```
{
  "name": "Microsoft.Insights.VMDiagnosticsSettings",
  "type": "extensions",
  "location": "[resourceGroup().location]",
  "apiVersion": "2016-03-30",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/myVM', copyindex())]"
  ],
  "properties": {
    "publisher": "Microsoft.Azure.Diagnostics",
    "type": "IaaSDiagnostics",
    "typeHandlerVersion": "1.5",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "xmlCfg": "[base64(concat(variables('wadcfgxstart'),
      variables('wadmetricsresourceid'),
      concat('myVM', copyindex()),
      variables('wadcfgxend')))]",
      "storageAccount": "[variables('storageName')]"
    },
    "protectedSettings": {
      "storageAccountName": "[variables('storageName')]",
      "storageAccountKey": "[listkeys(variables('accountid'),
        '2015-06-15').key1]",
      "storageAccountEndPoint": "https://core.windows.net"
    }
  }
},
```

This extension resource uses the storageName variable and the diagnostic variables to provide values. If you want to change the data that is collected by this extension, you can add more performance counters to the wadperfcounters variable. You could also choose to put the diagnostics data into a different storage account than where the VM disks are stored.

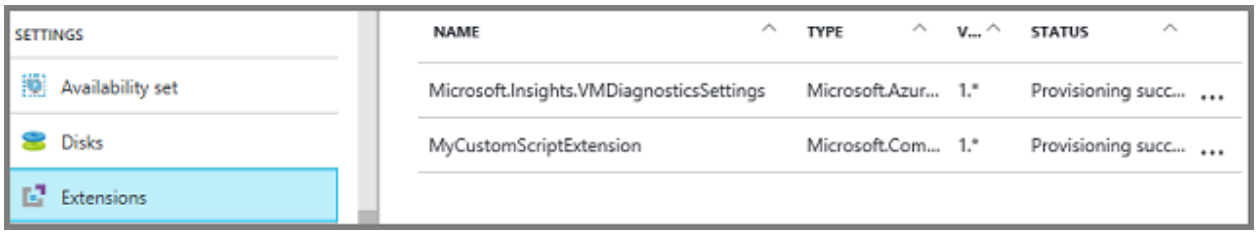
There are many extensions that you can install on a VM, but the most useful is probably the Custom Script Extension. In the example, a PowerShell script named start.ps1 runs on each VM when it first starts:

JSON

```
{
  "name": "MyCustomScriptExtension",
  "type": "extensions",
  "apiVersion": "2016-03-30",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/myVM', copyindex())]"
  ],
  "properties": {
    "publisher": "Microsoft.Compute",
    "type": "CustomScriptExtension",
    "typeHandlerVersion": "1.7",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "fileUris": [
        "[concat('https://', variables('storageName'),
          '.blob.core.windows.net/customscripts/start.ps1')]"
      ],
      "commandToExecute": "powershell.exe -ExecutionPolicy Unrestricted -
File start.ps1"
    }
  }
}
```

The start.ps1 script can accomplish many configuration tasks. For example, the data disks that are added to the VMs in the example aren't initialized; you can use a custom script to initialize them. If you have multiple startup tasks to do, you can use the start.ps1 file to call other PowerShell scripts in Azure storage. The example uses PowerShell, but you can use any scripting method that is available on the operating system that you're using.

You can see the status of the installed extensions from the Extensions settings in the portal:

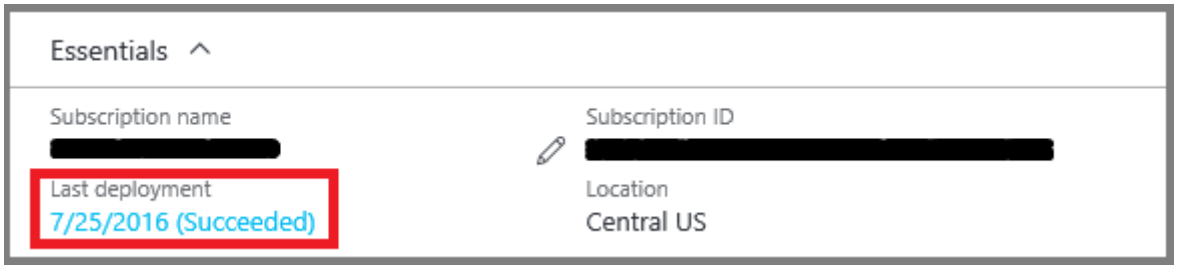


You can also get extension information by using the **Get-AzVMExtension** PowerShell command, the **vm extension get** Azure CLI command, or the **Get extension information** REST API.

Deployments

When you deploy a template, Azure tracks the resources that you deployed as a group and automatically assigns a name to this deployed group. The name of the deployment is the same as the name of the template.

If you're curious about the status of resources in the deployment, view the resource group in the Azure portal:



It's not a problem to use the same template to create resources or to update existing resources. When you use commands to deploy templates, you have the opportunity to say which mode you want to use. The mode can be set to either **Complete** or **Incremental**. The default is to do incremental updates. Be careful when using the **Complete** mode because you may accidentally delete resources. When you set the mode to **Complete**, Resource Manager deletes any resources in the resource group that aren't in the template.

Next Steps

- Create your own template using Authoring Azure Resource Manager templates.
- Deploy the template that you created using Create a Windows virtual machine with a Resource Manager template.
- Learn how to manage the VMs that you created by reviewing Create and manage Windows VMs with the Azure PowerShell module.

- For the JSON syntax and properties of resource types in templates, see Azure Resource Manager template reference.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Frequently asked question about Linux Virtual Machines

This article addresses some common questions about Linux virtual machines created in Azure using the Resource Manager deployment model. For the Windows version of this topic, see Frequently asked question about Windows Virtual Machines

What can I run on an Azure VM?

All subscribers can run server software on an Azure virtual machine. For more information, see Linux on Azure-Endorsed Distributions

How much storage can I use with a virtual machine?

Each data disk can be up to 32,767 GiB. The number of data disks you can use depends on the size of the virtual machine. For details, see Sizes for Virtual Machines.

Azure managed disks are the recommended disk storage offerings for use with Azure Virtual Machines for persistent storage of data. You can use multiple managed disks with each Virtual Machine. Managed disks offer two types of durable storage options: Premium and Standard managed disks. For pricing information, see managed disks Pricing.

Azure storage accounts can also provide storage for the operating system disk and any data disks. Each disk is a .vhd file stored as a page blob. For pricing details, see Storage Pricing Details.

How can I access my virtual machine?

Establish a remote connection to sign on to the virtual machine, using Secure Shell (SSH). See the instructions on how to connect from Windows or from Linux and Mac. By default, SSH allows a maximum of 10 concurrent connections. You can increase this number by editing the configuration file.

If you're having problems, check out Troubleshoot Secure Shell (SSH) connections.

Can I use the temporary disk, mounted under (/mnt) to store data?

Don't use the temporary disk, mounted under (/mnt) to store data. It's only there for temporary storage and if you want swap files. Data stored in there can't be recovered and it will be removed during certain virtual machine operations (stop/start, redeploy, and others).

Can I copy or clone an existing Azure VM?

Yes. For instructions, see How to create a copy of a Linux virtual machine in the Resource Manager deployment model.

Why am I not seeing Canada Central and Canada East regions through Azure Resource Manager?

The two new regions of Canada Central and Canada East aren't automatically registered for virtual machine creation for existing Azure subscriptions. This registration is done automatically when a virtual machine is deployed through the Azure portal to any other region using Azure Resource Manager. After a virtual machine is deployed to any other Azure region, the new regions should be available for subsequent virtual machines.

Can I add a NIC to my VM after it's created?

Yes, this is now possible. The VM first needs to be stopped deallocated. Then you can add or remove a NIC (unless it's the last NIC on the VM).

Are there any computer name requirements?

Yes. The computer name can be a maximum of 64 characters in length. See Naming conventions rules and restrictions for more information around naming your resources.

Are there any resource group name requirements?

Yes. The resource group name can be a maximum of 90 characters in length. See Naming conventions rules and restrictions for more information about resource groups.

What are the username requirements when creating a VM?

Usernames should be 1 - 32 characters in length.

The following usernames aren't allowed:

- `1`
- `123`
- `a`
- `actuser`
- `adm`
- `admin`
- `admin1`
- `admin2`
- `administrator`
- `aspnet`
- `backup`
- `console`
- `david`
- `guest`
- `john`
- `owner`
- `root`
- `server`
- `sql`
- `support_388945a0`
- `support`
- `sys`
- `test`
- `test1`
- `test2`
- `test3`
- `user`

- `user1`
- `user2`
- `user3`
- `user4`
- `user5`
- `video`

What are the password requirements when creating a VM?

There are varying password length requirements, depending on the tool you're using:

- Azure portal - between 12 - 72 characters
- Azure PowerShell - between 8 - 123 characters
- Azure CLI - between 12 - 123 characters
- Azure Resource Manager (ARM) templates - 12 - 72 characters and control characters aren't allowed

Passwords must also meet three out of the following four complexity requirements:

- Have lower characters
- Have upper characters
- Have a digit
- Have a special character (Regex match [\W_])

The following passwords aren't allowed:

- abc@123
- P@$$w0rd
- P@ssw0rd
- P@ssword123
- Pa$$word
- pass@word1
- Password!
- Password1
- Password22
- iloveyou!

Frequently asked question about Windows Virtual Machines

This article addresses some common questions about Windows virtual machines created in Azure using the Resource Manager deployment model. For the Linux version of this topic, see Frequently asked question about Linux Virtual Machines.

What can I run on an Azure VM?

All subscribers can run server software on an Azure virtual machine. For information about the support policy for running Microsoft server software in Azure, see Microsoft server software support for Azure Virtual Machines.

Certain versions of Windows 7, Windows 8.1, and Windows 10 are available to MSDN Azure benefit subscribers and MSDN Dev and Test Pay-As-You-Go subscribers, for development and test tasks. For details, including instructions and limitations, see Windows Client images for MSDN subscribers.

How much storage can I use with a virtual machine?

Each data disk can be up to 32,767 GiB. The number of data disks you can use depends on the size of the virtual machine. For details, see Sizes for Virtual Machines.

Azure managed disks are the recommended disk storage offerings for use with Azure Virtual Machines for persistent storage of data. You can use multiple managed disks with each Virtual Machine. Managed disks offer two types of durable storage options: Premium and Standard managed disks. For pricing information, see managed disks Pricing.

Azure storage accounts can also provide storage for the operating system disk and any data disks. Each disk is a .vhd file stored as a page blob. For pricing details, see Storage Pricing Details.

How can I access my virtual machine?

Establish a remote connection using Remote Desktop Connection (RDP) for a Windows VM. For instructions, see How to connect and sign on to an Azure virtual machine running Windows. A maximum of two concurrent connections are supported, unless the server is configured as a Remote Desktop Services session host.

If you're having problems with Remote Desktop, see Troubleshoot Remote Desktop connections to a Windows-based Azure Virtual Machine.

If you're familiar with Hyper-V, you might be looking for a tool similar to VMConnect. Azure doesn't offer a similar tool because console access to a virtual machine isn't supported.

Can I use the temporary disk (the D: drive by default) to store data?

Don't use the temporary disk to store data. It is only temporary storage, so you would risk losing data that can't be recovered. Data loss can occur when the virtual machine moves to a different host. Resizing a virtual machine, updating the host, or a hardware failure on the host are some of the reasons a virtual machine might move.

If you have an application that needs to use the D: drive letter, you can reassign drive letters so that the temporary disk uses something other than D:. For instructions, see Change the drive letter of the Windows temporary disk.

How can I change the drive letter of the temporary disk?

You can change the drive letter by moving the page file and reassigning drive letters, but you need to make sure you do the steps in a specific order. For instructions, see Change the drive letter of the Windows temporary disk.

Can I add an existing VM to an availability set?

No. If you want your VM to be part of an availability set, you need to create the VM within the set. There currently isn't a way to add a VM to an availability set after it has been created.

Can I upload a virtual machine to Azure?

Yes. For instructions, see Migrating on-premises VMs to Azure.

Can I resize the OS disk?

Yes. For instructions, see How to expand the OS drive of a Virtual Machine in an Azure Resource Group.

Can I copy or clone an existing Azure VM?

Yes. Using managed images, you can create an image of a virtual machine and then use the image to build multiple new VMs. For instructions, see Create a custom image of a VM.

Why am I not seeing Canada Central and Canada East regions through Azure Resource Manager?

The two new regions of Canada Central and Canada East are not automatically registered for virtual machine creation for existing Azure subscriptions. This registration is done automatically when a virtual machine is deployed through the Azure portal to any other region using Azure Resource Manager. After a virtual machine is deployed to any other Azure region, the new regions should be available for subsequent virtual machines.

Does Azure support Linux VMs?

Yes. To quickly create a Linux VM to try out, see Create a Linux VM on Azure using the Portal.

Can I add a NIC to my VM after it's created?

Yes, this is now possible. The VM first needs to be stopped deallocated. Then you can add or remove a NIC (unless it's the last NIC on the VM).

Are there any computer name requirements?

Yes. The computer name can be a maximum of 15 characters in length. See Naming conventions rules and restrictions for more information around naming your resources.

Are there any resource group name requirements?

Yes. The resource group name can be a maximum of 90 characters in length. See Naming conventions rules and restrictions for more information about resource groups.

What are the username requirements when creating a VM?

Usernames can be a maximum of 20 characters in length and cannot end in a period (".").

The following usernames are not allowed:

- `1`
- `123`
- `a`
- `actuser`
- `adm`
- `admin`
- `admin1`
- `admin2`
- `administrator`
- `aspnet`
- `backup`
- `console`
- `david`
- `guest`
- `john`
- `owner`
- `root`
- `server`
- `sql`
- `support_388945a0`
- `support`
- `sys`
- `test`
- `test1`

- `test2`
- `test3`
- `user`
- `user1`
- `user2`

What are the password requirements when creating a VM?

There are varying password length requirements, depending on the tool you are using:

- Azure portal - between 12 - 123 characters
- Azure PowerShell - between 8 - 123 characters
- Azure CLI - between 12 - 123

- Have lower characters
- Have upper characters
- Have a digit
- Have a special character (Regex match [\W_])

The following passwords are not allowed:

Expand table

abc@123	iloveyou!	P@$$w0rd	P@ssw0rd	P@ssword123
Pa$$word	pass@word1	Password!	Password1	Password22

For additional information and restrictions for password creation reference this password guidance documentation.

Support and troubleshooting for Azure VMs

Article • 08/22/2024

Here are suggestions for where you can get help when developing your Azure Virtual Machines solutions.

Self help troubleshooting



Various articles explain how to determine, diagnose, and fix issues that you might encounter when using Azure Virtual Machines. Use these articles to troubleshoot deployment failures, unexpected restarts, connection issues and more.

For a full list of self help troubleshooting content, see Azure Windows Virtual Machines troubleshooting documentation and Azure Linux Virtual Machines troubleshooting documentation.

Post a question on Microsoft Q&A



For quick and reliable answers on your technical product questions from Microsoft Engineers, Azure Most Valuable Professionals (MVPs), or our expert community, engage with us on Microsoft Q&A, Azure's preferred destination for community support.

If you can't find an answer to your problem using search, submit a new question to Microsoft Q&A. Use one of the following tags when asking your question:

Expand table

Area	Tag
Azure Virtual Machines	azure-virtual-machines
Azure SQL Virtual Machines	azure-sql-virtual-machines
Azure Virtual Machine Scale Sets	azure-virtual-machine-scale-set

Create an Azure support request

Explore the range of Azure support options and choose the plan that best fits, whether you're a developer just starting your cloud journey or a large organization deploying business-critical, strategic applications. Azure customers can create and manage support requests in the Azure portal.

- If you already have an Azure Support Plan, open a support request here.
- To sign up for a new Azure Support Plan, compare support plans and select the plan that works for you.

Create a GitHub issue

If you need help with the language and tools used to develop and manage Azure Virtual Machines, open an issue in its repository on GitHub.

Expand table

Library	GitHub issues URL
Azure PowerShell	https://github.com/Azure/azure-powershell/issues
Azure CLI	https://github.com/Azure/azure-cli/issues
Azure REST API	https://github.com/Azure/azure-rest-api-specs/issues
Azure SDK for Java	https://github.com/Azure/azure-sdk-for-java/issues
Azure SDK for Python	https://github.com/Azure/azure-sdk-for-python/issues
Azure SDK for .NET	https://github.com/Azure/azure-sdk-for-net/issues
Azure SDK for JavaScript	https://github.com/Azure/azure-sdk-for-js/issues
Jenkins	https://github.com/Azure/jenkins/issues
Terraform	https://github.com/Azure/terraform/issues
Ansible	https://github.com/Azure/Ansible/issues

Stay informed of updates and new releases

Learn about important product updates, roadmap, and announcements in Azure Updates.

News and information about Azure Virtual Machines is shared at the Azure blog.

Next steps

Learn more about Azure Virtual Machines

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A